

*В дослідженні розвинута формалізація нафтогазової предметної області, на основі введення ідеї агента-орієнтованого програмування. Запропоновано використання агента-орієнтованого програмування для прогнозування неорганічних речовин по стовбуру свердловини при видобутку вуглеводнів*

*Ключові слова: неорганічні речовини, відкладення, інтелектуальні агенти, вуглеводні*

*В исследовании развита формализация нефтегазовой предметной области, на основе введения идеи агента-ориентированного программирования. Предложено использование агента-ориентированного программирования для прогнозирования неорганических веществ по стволу скважины при добыче углеводородов*

*Ключевые слова: неорганические вещества, отложения, интеллектуальные агенты, углеводороды*

*In this research formalization of oil and gas subject, that is basis on introduction idea agent-oriented programming is given. The use of the agent-oriented programming is offered for prognostication inorganic matters on the barrel of mining hole at the booty of hydrocarbons*

*Keywords: inorganic matter, sediments, intelligent agents, hydrocarbons*

## ЗАСТОСУВАННЯ ІНТЕЛЕКТУАЛЬНИХ АГЕНТІВ ДО ФОРМАЛІЗАЦІЇ ПРОЦЕСУ УТВОРЕННЯ СОЛЕЙ В СТОВБУРІ СВЕРДЛОВИНИ

**М. М. Яцишин**

Кандидат технічних наук, доцент\*

Контактний тел.: 066-743-99-60

E-mail: yatsyshyn@gmail.com

**Т. В. Дитко**

Асистент\*

Контактний тел.: 050-677-77-73

E-mail: t.dytko@mail.ru

**І. В. Бронівський**

Асистент\*

Контактний тел.: 067-962-23-74

E-mail: admin@brun.if.ua

\*Кафедра програмного забезпечення автоматизованих систем

Івано-Франківський національний технічний університет нафти і газу  
вул. Карпатська 15, м. Івано-Франківськ, Україна, 76000

### Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

У сучасних умовах процеси пошуку, розвідки, розробки нафтогазового родовища та експлуатації свердловин відзначаються слабкою структуризацією даних.

При експлуатації газових і газоконденсатних свердловин виникає проблема утворення нашарувань по стовбуру свердловини. У склад цих відкладень входять як органічні так і неорганічні речовини (НР). У відкладеннях утворюються агресивні сполуки неорганічного походження, які негативно впливають на стан колон експлуатаційної свердловини. Важливою є проблема формалізації процесу поведінки хімічного складу пластових вод, які транспортуються по стовбуру свердловини при різних термодинамічних умовах.

За умови, якщо б послідовне (процедурне) програмування дозволяло знаходити рішення в будь-яких ситуаціях, то не було би необхідності для розвитку технологій паралельного і розподіленого програмування і відповідно агента-орієнтованого. У процесі пошуку розробниками нових підходів у нафтогазовій

промисловості і створюються альтернативні моделі програмного забезпечення. У багатьох відносинах агенти й агента-орієнтоване програмування можна розглядати як чергову ступінь розвитку програмування [1, 2].

Однією з областей, які відповідають вище наведеним характеристикам є можливість прогнозування відкладень НР по стовбуру свердловини при видобутку газу і газоконденсату. Для опису цієї частини нафтогазової предметної області (ПО) застосуємо агента-орієнтоване моделювання, а програмним середовищем – C++. Де під агентом вважають це головний виконавець в домені який володіє однією або декількома можливостями, які утворюють єдину і комплексну модель виконання, яка може включати доступ до зовнішнього програмного забезпечення (ПЗ), користувачам (людям) і засобам зв'язку. Яка ж різниця між агентом і об'єктом? Причина чому об'єкт можливо замінити на агент – агенти по суті засновані на об'єктах. Вимогою до визначення агента буде те, що він повинен задовольняти визначенню об'єкта, тобто агент - це об'єкт визначеного виду. Виходячи з того, що C++ підтримує інтерфейсні, контейнерні і каркасні класи, ми можемо з таким же успіхом ввести й агентні класи.

**Аналіз останніх досліджень і публікацій**

Існує кілька категорій агентів: інтерфейсні агенти (представляють наступне покоління взаємодії між людиною і комп'ютером і забезпечують новий користувацький інтерфейс з комп'ютером); агенти пошуку (виконують різні види пошуку); агенти моніторингу/керування (патрулюють, спостерігають, відслідковують виконувані дії, керують і контролюють пристрої й умови, дані і процеси); агенти підтримки прийняття рішень (забезпечують аналіз і синтез інформації, інтерпретацію умов і даних, планування дій і оцінку результату). Існують і агенти змішаного типу, які можна віднести до кількох категорій одночасно, оскільки для розподілу агентів по категоріях немає ніяких жорстких правил.

Ці категорії представлені для зручності і використовуються як відправну точку в спробі класифікувати агенти.

Вище визначені лише види діяльності, що характерні для агентів тієї або іншої категорії. Ці категорії не є винятковою сферою агентів. Подібним чином по категоріях можна розділити й інші класи ПЗ (наприклад, експертні й об'єктно-орієнтовані системи). У деяких випадках єдиною відмінністю може виявитися сам факт того, що розглядається агенти, а не об'єкти чи експертні системи [3].

Для формалізації процесу утворення солей, прийдеться працювати з кількома типами агентів, які будуть цілісно співпрацювати в системі.

Виділення невирішених раніше частин загальної проблеми, яким присвячується дана стаття

**Формулювання цілей статті (постановка завдання)**

При розв'язанні проблем і питань, які ставить сучасний стан розвитку досліджень у нафтогазовій ПО, дослідники зіштовхуються з проблемою невизначеності, слабой структурованості і малою інформативністю даних, які характеризують той чи інший етап розробки, чи окремі процеси у ньому. На даному етапі у ПО існують процеси, які розв'язуються з недостатньою точністю і впевненістю.

Спробуємо запропонувати вирішення проблеми прогнозування неорганічних речовин, за допомогою інструментів агенто-орієнтованого програмування.

**Виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів**

Клас агента (рис. 1) визначає типові методи ініціалізації, читання і запису, що повинні мати практично будь-який об'єкт. Атрибути цього класу включають змінні стани, що визначають об'єкт. Якщо ж обмежитися переліком цих атрибутів і методів, ми отримаємо тільки традиційний об'єкт. Рациональний компоненти створюють когнітивні структури даних і методи міркувань (логічного висновку).

Саме рациональний компонент трансформує "звичайний" об'єкт в агент [3].



Рис. 1. Логічна структура класу агента

**Когнітивні структури даних**

Метод організації агента визначає, як дані повинні бути концептуально структуровані і які операції доступу можуть бути застосовані до цієї структури.

Якщо для типів даних взагалі й абстрактних типів даних важливо, що зберігати, то для структур даних важливо, як зберігати. Наприклад, цілочисельний тип даних визначає деяку "сутність", що характеризується наявністю компонента даних і деякої кількості арифметичних операцій. Цей компонент даних не має дробової частини і складається з негативних і позитивних чисел. Специфікація типу даних нічого не "вказує на" про те, як цілі числа потрібно використовувати або як до них отримати доступ. Однак специфікація структури даних (наприклад, стека) визначає список елементів, що зберігаються за принципом "останнім прибув - першим обслужений" (last-in-first-out- LIFO). Іншими словами, елемент, поміщений у стек останнім, повинний бути витягнутий з нього раніш інших елементів. Когнітивні структури даних обмежують правила організації даних і доступу до них такими, котрі відносяться до області логіки і епістемології.

Особливості когнітивних структур даних визначаються правилами логічного висновку, методами міркувань (тобто дедукцією, індукцією й абдукцією), поняттями епістемологічних даних, знаннями, обґрунтуваннями, переконаннями, посилками, висловленнями, помилковими доведеннями і висновками.

Абстрактні типи звичайно реалізуються як типи даних, оголошені за допомогою ключових слів struct або class.

Наприклад, так:

```
struct question {
    //...
    string RequiredInformation;
    target_object QuestionDomain;
    string Tense;
    string Mood;
    //...
};

class justification {
    //...
    time EventTime;
    bool Observed;
    bool Present;
    //...
};
```

Шаблонні і контейнерні C++-класи можна використовувати для організації таких когнітивних структур даних, як знання, наприклад, так:

```
class preliminary_knowledge {
//...
data_task <question,belief> Opinion;
data_task <conclusion,justification> SimpleKnowledge;
set<propositions> Argument;
//...
};
```

### Методи міркувань

Під методами міркувань (рис. 1) розуміють дедукцію, індукцію й абдукцію. Незважаючи на те що в агентно-орієнтованій архітектурі потрібне їхнє використання, не існує конкретних посилань на те, як вони реалізуються. Дедукція, індукція й абдукція відносяться до процесів високого рівня. Подробиці реалізації цих процесів - особиста справа розроблювача ПЗ. Не існує єдино правильного способу реалізації процесу міркувань, який іноді називають машиною, (або механізмом) логічного висновку. При цьому на практиці застосовується кілька розповсюджених способів реалізації цього процесу. Знайшли тут застосування методи аналізу цілей і засобів, а також такі алгоритми графів, як "пошук в глибину" (Depth First Search - DFS) і "пошук у ширину" (Breadth First Search - BFS). Існує також ціла сукупність методів доказу теорем, який можна використовувати для реалізації методів міркувань і механізмів логічного висновку. Тут важливо відзначити, що клас агента може мати один або кілька методів міркувань. Серед основних способів їх реалізації можна виділити: зворотна побудова ланцюжка (керований цілями метод, у якому процес починається з припущення, твердження або гіпотези і прагне знайти підтверджуюче доведення); пряма побудова ланцюжка (керований даними метод, що починається з аналізу наявних даних або фактів і приходить до визначених висновків); аналіз цілей і засобів (використовує безліч операторів для послідовного рішення під задач доти, поки не буде вирішена вся задача в цілому).

Ці методи досить зрозумілі і широко доступні в багатьох комп'ютерних бібліотеках, оболонках і мовах програмування. Вони є "будівельними блоками" для базових методів міркувань. Щоб зрозуміти, як відбувається процес міркування, використовуємо одне з правил генерування висновку, а саме модуспоненс (правило відділення), і побудуємо простий метод міркування. Візьмемо наступне твердження. Якщо при аналізі пластової води концентрація певних НР перевищує максимальну допустиму норму при якій можливе формування НР, то утворення солевих відкладів можливе. За умови якщо ми з'ясуємо, що концентрація НР дійсно існує, то можливе утворення НР. Правило модуспоненс має наступний формат.

$P \rightarrow Q$

$P$

—

$Q$

Тут:

$P$  – концентрація НР перевищує максимальну допустиму норму при якій можливе формування солевих відкладів,

$Q$  – утворення неорганічних (солевих) відкладів.

Спроекуємо простий агент забезпечення рішення, що дозволить довідатися, будуть відклади чи ні. Цьому агенту потрібно довідатися все можливе про компонентний склад і властивості присутніх у пластовій воді НР. Нехай є НР [4]:

Галіт (NaCl)  
 Сульфат натрію (Na<sub>2</sub>SO<sub>4</sub>)  
 Хлористий калій (KCl)  
 Сірчанокислий калій (K<sub>2</sub>SO<sub>4</sub>)  
 Хлористий магній (MgCl<sub>2</sub>)  
 Сірчанокислий магній (MgSO<sub>4</sub>)  
 Хлористий калій (CaCl<sub>2</sub>)  
 Кальцит (CaSO<sub>4</sub>)

Кожна з цих НР має свої властивості(характеристики), які визначенні дослідним шляхом. Якщо наш агент одержить доступ до довідника чи спеціально створеної для цього бази даних цих властивостей, то приведений вище список НР можна використовувати для представлення деякої частини переконань нашого агента.

Виникає питання: як перейти від списку НР до переконань? Для початку спробуємо розробити просту структуру тверджень.

```
struct existing_deposit {
//...
string Consist;
salt Concentration;
string Prognostation;
salt Value;
//...
};
```

Потім використаємо контейнерний клас для представлення переконань нашого агента що до властивостей НР.

```
set<existing_Deposit > NeoSubKnowledge;
```

Якщо визначена сіль міститься в множині *NeoSubKnowledge*, то наш агент переконаний у тім, що в зазначена сіль саме за такої концентрації буде відкладатись.

Отже, ми можемо зафіксувати будь-яку сіль відповідно до заданої структури.

```
//...
existing_deposit Deposit ;
Deposit.Consist.append ("NaCl ");
Deposit.Prognostation.append("Na2SO4");
Deposit.Concentration ("30"); (дані про концентрацію)
Deposit.Value ("активні"); (властивість солей присутніх у пластовій воді)
NeoSubKnowledge.insert (Deposit);
//...
```

Якщо помістити кожен НР(дані про її властивості) у множину *NeoSubKnowledge*, то переконання нашого агента про властивості НР будуть цілком описані. Зверніть увагу на те, що не всі компоненти(НР)

будуть виявленні при аналізі пластової води і вони відрізняються за активністю і розчинністю (АіР):

Галіт (NaCl): активність - (0,688 - 0,986),  
молекулярність - (2,5 - 6,0),  
розчинність - (42,4 - 46,0).

Сульфат натрію (Na<sub>2</sub>SO<sub>4</sub>):  
активність - (0,1376 - 0,452),  
молекулярність - (0,1 - 4,0),  
розчинність - (42,1 - 44,4).

Хлористий калій (KCl):  
активність - (0,770 - 0,583),  
молекулярність - (0,1 - 4,5),  
розчинність - (68 - 74,8).

Сірчаноокислий калій (K<sub>2</sub>SO<sub>4</sub>):  
активність - (0,436 - 0,229),  
молекулярність - (0,1 - 0,7),  
розчинність - (29,7 - 34,2).

Хлористий магній (MgCl<sub>2</sub>):  
активність - (0,474 - 13,92),  
молекулярність - (0,1 - 5,0),  
розчинність - (95,3 - 125,7).

Сірчаноокислий магній (MgSO<sub>4</sub>):  
активність - (0,13 - 0,0473),  
молекулярність - (0,1 - 3,0),  
розчинність - (24,1 - 6,7).

Хлористий калій (CaCl<sub>2</sub>):  
активність - (0,448 - 11,11),  
молекулярність - (0,1 - 5,0),  
розчинність - (129 - 137).

Кальцит (CaSO<sub>4</sub>):  
активність - (0,150 - 0,0357),  
молекулярність - (0,1 - 3,0),  
розчинність - (0,130 - 0,053).

Тому, незважаючи на те що не у всіх свердловинах і не завжди однаковий компонентний склад НР у пластовій воді, база даних про властивості НР дозволяє зробити висновок про утворення чи ні неорганічних відкладень. Завдання полягає от чому: як про це може довідатися наш агент? Агент на основі своїх знань про властивості НР повинен мати деякий алгоритм генерування висновку про те, чи існують неорганічні відкладення. Ми використовуємо простий ланцюговий метод. Переглядаємо елементи множини *NeoSubKnowledge* і знаходимо дані про властивості НР присутніх у пластовій воді. Опитуємо атрибут **What (які)** цього елемента. Якщо він буде рівним значенню "АіР цієї солі" відповідного компонента, процес пошуку зупиниться, оскільки вже можна припустити, що є умови для відкладень. У протилежному випадку зберігається знайдений (проміжні) характеристики компонента у стеці. Потім шукаємо НР з атрибутом **Whom**, рівним "граничне значення концентрації". При цьому може виявитися, що характеристики цієї НР не достатні для утворення неорганічних відкладень. Оскільки далі зберігати елемент множини, що відповідає характеристикам солі "Галіт", немає ніякого сенсу, він видаляється із стека, зробивши позначку, що властивості цієї солі були розглянуті. Потім повторюється пошук НР, які присутні у пластовій воді. Знаходимо НР: "Сульфат натрію". Перевіряється, чи не дорівнює його атрибут **What(які)** значенню "АіР цієї солі", якщо результат перевірки "true", зберігаємо цей елемент у стеку.

Потім шукається НР з атрибутом **Whom**, рівним " граничне значення концентрації ". Знаходиться НР " Природна сода " і також поміщається у стек. Після цього переглядається НР для знаходження елемента, у якого атрибут **Whom** буде рівним значенню "граничне значення концентрації". Для кожної НР перевіряється значення атрибута **Whom**. Якщо він дорівнює значенню " граничне значення концентрації ", то НР, поміщені в стек, представляють у цілому сукупність(множину) НР, які можуть відкладатись при даних значеннях концентрації, перша з яких знаходиться на "дні" стека, а остання перевірена і яка задовольнила алгоритм генерування висновку - у вершині. Якщо ми пройдемо за всім списком наявних НР у пластовій воді відповідної свердловини і не знайдемо жодного з атрибутом **What(які)**, рівним значенню "АіР цієї солі" (аналіз пластової води взагалі показав значення присутності НР, якими можна знехтувати), або вичерпаються можливі варіанти перевірки атрибута **What(які)** для верхнього елемента стека, то, витягаючи верхній елемент зі стека, будемо шукати наступний елемент, значення атрибута **Whom** якого збігається зі значенням атрибута **What(які)** елемента, розташованого у вершині стека.

Цей процес повторюється доти, поки стек не спорожніє або все-таки не знайдемо сіль яка може відкладатись.

Для визначення, чи існує НР у пластовій воді відповідної свердловини, використовується в даному випадку спрощений метод DFS (Depth First Search - "пошук в глибину").

Запропонований простий агент буде використовувати DFS-метод для з'ясування, чи існує можливість існування відкладів НР. З'ясувавши цей факт, агент оновить свої переконання щодо відповідної свердловини.

Тепер агент переконаний, що у певній свердловині будуть відклади. Вводимо додаткову умову щодо відкладів у відповідній свердловині.

Якщо термобаричні умови свердловини будуть: температура – 412 К, дебіт – 0,5 м<sup>3</sup>/год і існують такі НР, які можуть відкластись, то у відповідній свердловині будуть відклади.

Тепер агент повинен з'ясувати, чи перевищують термобаричні умови свердловини граничні значення і чи існує можливість утворення відкладів НР. Щоб з'ясувати термобаричні умови відповідної свердловини, агент повинен спочатку дізнатись температуру на гирлі свердловини і розрахувати ці значення по її стовбуру. Нехай запевнимо програмного агента в тому, що термобаричні умови задовільність можливість утворення відкладів НР. Потім агент повинен переконатися в тому, що дебіт менший за 0,5 м<sup>3</sup>/с. На основі вмісту множини *NeoSubKnowledge* агент зможе прийти до висновку про існування можливості відкладів.

На підставі переконань про можливість відкладів НР і розрахованих термобаричних умов агент використовує процес прямої побудови ланцюжка (тобто міркувань від вихідних посилок до цільової гіпотези) і приходять до висновку, що відклади НР мають місце. Формат міркувань цього процесу має такий вид.

A → B

(B i C) → D

A

C

—

D

Тут:

A = термобаричні умови – температура 412 К,

B = Дебіт  $\leq 0,5$ ,

C = Існує можливість неорганічних відкладів,

D = НР відклади на відповідній свердловині будуть.

У цьому прикладі агент переконується, що елементи A і C істинні. З використанням правил ведення міркувань агент отримує, що елементи B і D дорівнюють значенню ІСТИНА. Отже, агент робить висновок про те, що НР відклади на відповідній свердловині будуть. Подібний вид обробки наявних даних можна застосувати до агента в ситуації, коли родовищ не одне, а кілька, і користувач хоче щоб агент регулярно складав прогноз відкладів НР по кожній свердловині зокрема. Користувач має намір потім одержувати від агента довідку про те, яка свердловина працювала, яка перебувала на плановому ремонті по попередженні утворення пробок у видобувній свердловині, а яка планується ставитись на ремонт і т.д. Агент повинний мати знання і повноваженнями встановлювати графіки планових ремонтів. Що півроку агент повинний представляти ряд прийнятних графіків планових ремонтів. Агент у цьому випадку для одержання результату використовує простий метод прямої побудови ланцюжка і метод DFS (Depth First Search). Щоб реалізувати цей вид міркувань, ми використовували такі типи даних, як struct та класи стеків і множин. Ці класи використовуються для

збереження знань, припущень і методів міркувань. Вони дозволяють реалізувати когнітивні структури даних (Cognitive Data Structures- CDS). Для підтримки процесу міркувань, а саме при опитуванні наших структур даних (стека і множини) використовуємо DFS-методи.

При сполученні методу прямої побудови ланцюжка і методу DFS створюється процес, відповідно до якого одне припущення може бути підтвержене на основі вже прийнятих попередніх рішень. Той факт, що агент раціональний і діє відповідно до правил побудови міркувань, дозволяє розробнику зосередитися на коректному моделюванні задачі, що виконується агентом, а не на прагненні явно керувати паралелізмом у програмі. Це обґрунтування повинно спиратися на добре визначені правила ведення, вид міркувань. Взаємодію агентів простіше представити, чим взаємодію анонімних модулів, оскільки границі між агентами більш чіткі й очевидні. Кожен агент має мету, що лежить на поверхні. Знання, або інформація, необхідні кожному агентові для досягнення його мети, у цьому випадку легко визначаються.

---

#### Висновки з даного дослідження і перспективи подальших розвідок у даному напрямку

---

В даному дослідженні запропоновано підхід до опису прогнозування процесу утворення неорганічних відкладень по стовбуру свердловини на основі агенто-орієнтованого підходу, що дає можливість розділення параметрів, які характеризують цей процес по класах, по зв'язках між ними і створення програмного забезпечення для наочного представлення отриманих результатів.

Подальші дослідження будуть спрямовані на вдосконалення запропонованого агента та розробку його прототипу. Для узгодження роботи інтелектуальних агентів буде використано методи "Класної дошки", семантичних мереж або фреймів.

---

#### Література

1. Никольский Б. П. Справочник химика / Б. П. Никольский, О.Н. Григоров./ 2-е издание.- издательство «Химия»- Ленинград.- 1968.- т.1-7
2. Юрчишин В.М. Використання теорії категорій для діагностики свердловин при їх консервації та ліквідації / В.М. Юрчишин /Науковий вісник національного технічного університету. - 2003.- №1.- С.121-124.
3. Яцишин М.М. Математична модель процесу кристалізації однокомпонентних солей у стовбуру експлуатаційної свердловини/ М.М. Яцишин, Т.В. Дитко, І.В. Бронівський /Розвідка та розробка нафтових родовищ 4(17) 2005.- Івано-Франківськ.- С.42-45.
4. Piessens F. Categorical data-specification / F. Piessens, E. Steegmans / Theory and Application of Categories.- vol.- 1.- No. 8.- 1995.- pp.156-173.
5. Довідник з нафтогазової справи. За заг. ред. докторів технічних наук В.С. Бойка, Р.М.Кондрата, Р.С. Яремійчука. - К.: Львів, 1996.- с.620.
6. Юрчишин В.М. Використання теорії категорій для діагностики свердловин при їх консервації та ліквідації / В.М. Юрчишин /Науковий вісник національного технічного університету. - 2003.- №1.- С.121-124.