

**Кузьменко О.В., Бойко А.О., Колдовський В.В., Яровенко Г.М., Сігал  
Н.В., Деркач В.І., Войтович В.І.**

**Автоматизований аналіз касових операцій банків з  
питань дотримання законодавства України, яке  
регулює відносини у сфері запобігання та протидії  
легалізації (відмиванню) доходів, одержаних  
злочинним шляхом**

проміжний звіт про НДР

## ЗМІСТ

ВСТУП.....	5
I. ПРОГРАМНІ ЗАСОБИ ТА БАЗИ ДАНИХ: НЕВІД'ЄМНА СКЛАДОВА СУЧАСНОГО БАНКУ .....	10
1.1. Методи штучного інтелекту.....	10
1.2. Методи створення і ведення баз даних комерційному банку.....	18
1.3. Основні засади автоматизації банківських процесів.....	29
II. БАЗОВІ ЗАСАДИ ПРОВЕДЕННЯ ПЕРЕВІРКИ БАНКУ З ПИТАНЬ ФІНАНСОВОГО МОНІТОРИНГУ .....	40
III. АВТОМАТИЗОВАНИЙ ІНСТРУМЕНТ ІНСПЕКТОРА.....	46
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57

## ВСТУП

З розвитком задач автоматизації банківської діяльності змінюються підходи до порядку здійснення банківських операцій, їх обліку, взаємодії головного банку чи балансових установ з іншими підрозділами в межах одного банку. На сьогодні інформаційні технології є невід'ємною частиною нашого життя, тим більше, вони є «кровоносною системою» і в банківській діяльності.

Будь-яка операція банку має своє відображення в програмних комплексах: починаючи зі здійснення платежів і закінчуючи обміном оперативними повідомленнями між відділами в банку, щоденним навчанням персоналу. До відома, ще 10 років тому існували банки, у яких прикладний модуль «Операційний день банку» (далі – ОДБ) дозволяв повернутися в будь-який день року та здійснити там внутрішню проводку.

Показником якісного розвитку задач автоматизації є переведення всіх без виключення філій в безбалансові відділення. Очевидно, що без належного програмного забезпечення, стійких комунікаційних систем такого не було б.

Зараз існують банки, які повністю відмовились від зберігання більшості документів на паперових носіях. Збереження ж документації в електронних архівах дає змогу банкам не тільки забезпечувати їх компактність, але й дозволяє більш якісно здійснювати їх обробку з метою контролю.

Саме вимоги до оперативного контролю з боку керівництва або власників банку чи керівників підрозділів будь-яких рівнів і спонукають до розвитку програмного забезпечення в банку.

Існують банки, у яких жодна операція не виконується без відповідного погодження (верифікації) у відповідних структурах головного банку. І це здійснюється оперативно, непомітно для клієнта. Що це дає:

оперативний контроль, повне виключення несанкціонованих операцій на місцях. Є, звичайно, нюанси, але вони пов'язані, в першу чергу, з людським фактором. Чи можливий би був такий механізм без належного програмного забезпечення?

З одного боку, зміна стратегії і тактики банку вимагає зміни у відповідних програмних комплексах, системах електронної пошти та передачі іншої інформації, і, навпаки, зі зміною архітектури взаємодії задач автоматизації змінюється архітектура системи банку загалом.

Рівень програмного забезпечення різний у різних банків, але можна визначити мінімум інформації, яка відображається в електронному вигляді у всіх банках, при цьому алгоритми обробки більш-менш однакові. Наприклад, внутрішні та міжбанківські платежі, інформація про клієнта банку, ведення рахунків банку, сюди ж: ведення реєстрів операцій, які підлягають фінансовому моніторингу, анкети клієнтів за результатами ідентифікації.

Слід зауважити, що ідентифікація клієнта – це не якась закінчена діяльність (операція) банку, це безперервний процес. Що краще: кожного разу при отриманні якоїсь уточнюючої інформації про клієнта заводити нову паперову анкету (дописувати на полях існуючої – що ще гірше), чи мати анкету в електронному вигляді і доповнювати її в будь-який момент, а коли треба – роздрукувати? Звичайно, друге. Тому, якщо де ще немає програми (модуля) для ведення та супроводження анкет або якщо нею не користуються – це тимчасове явище, рано чи пізно переважатиме електронна форма.

Отже, є всі підстави вважати, що задачі автоматизації розвиватимуться і в майбутньому. Будь-який банк зацікавлений в оптимізації своєї структури, та, головне, в оперативному та ефективному контролі за своєю діяльністю. Водночас вже сьогодні ми можемо з упевненістю говорити про певний мінімум інформації, яка має відображення на електронних носіях у всіх без виключення банківських установах.

Яка ж ситуація складається під час перевірки з питань фінансового моніторингу у інспекторів і у банків?

Банки вимушені нести витрати на доопрацювання програмного забезпечення, залучення персоналу на роботу в цьому напрямку.

Прибутку з цього банк не має, зате має серйозні проблеми у разі невиконання вимог Закону України «Про запобігання та протидію легалізації (відмиванню) доходів, одержаних злочинним шляхом, або фінансуванню тероризму» (далі – Закон) та положень Національного банку України.

Після введення Закону у 2003 році існувала думка, що це тимчасове явище і, відповідно, ставлення до цього було недбалим. З тих пір пройшло майже 8 років і вага фінансового моніторингу постійно зростає.

Слід зазначити, що використання автоматизованих інструментів при здійсненні регулювання і нагляду визнане міжнародною практикою.

Так, наприклад, у Банку Франції існує спеціальний підрозділ, який здійснює аналіз даних комерційних банків та, в разі потреби, формує і надсилає повідомлення до TRACFIN (рис. 1).

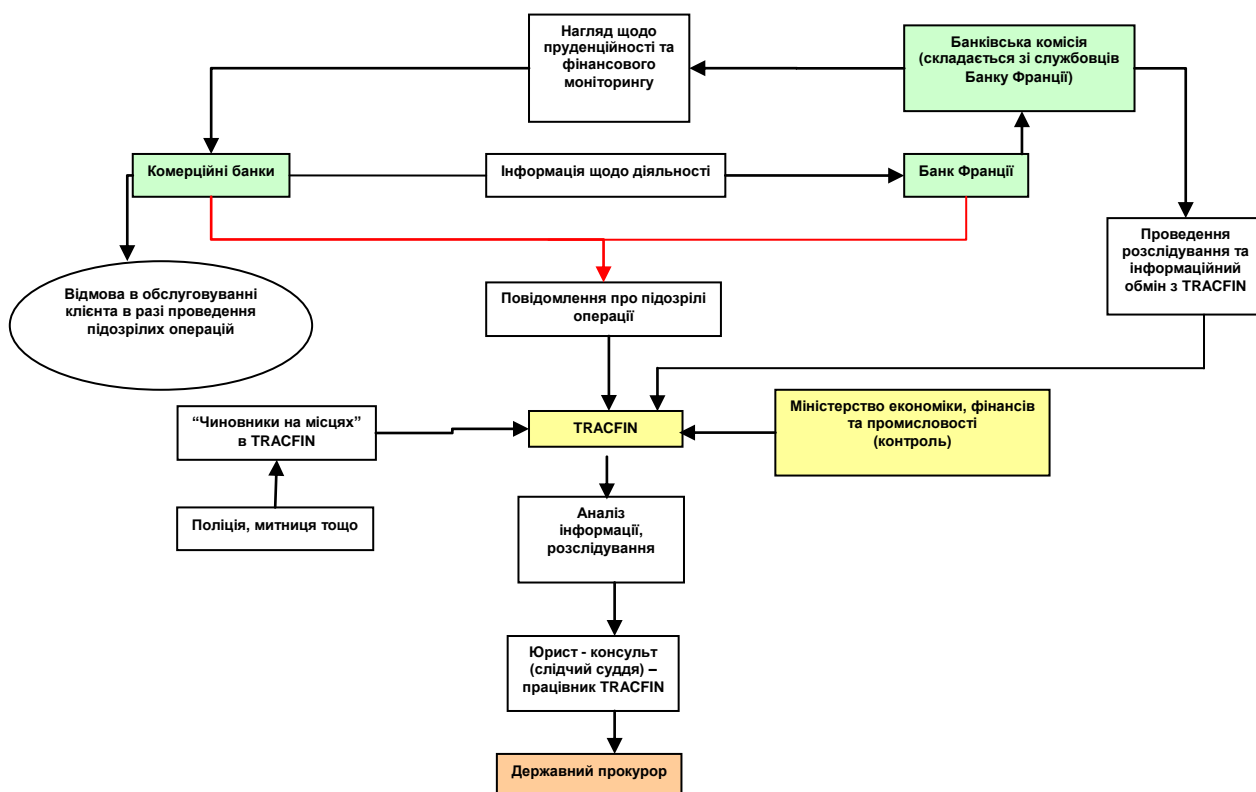


Рис. 1. Система фінансового моніторингу Франції (банківські установи)

Система паралельного аналізу Банком Франції даних щодо фінансових операцій, здійснених через банківську систему країни, забезпечує стимул для комерційних банків бути ретельними, унеможливаючи формалізований підхід до проведення фінансового моніторингу.

Визначальну роль у системі фінансового моніторингу у банківській сфері Франції відіграє саме Банківська комісія, яка, зокрема, здійснює виїзні та безвиїзні перевірки банків із вказаного питання.

Спеціальний підхід до інспектування базується на оцінці механізму (схеми) здійснення аудиту щодо внутрішнього моніторингу та щодо здійснення трансакцій.

При цьому визначено 3 основні моменти, які потребують уваги в ході інспектування:

- узгодженість статутних та регуляторних норм, та їх внутрішньої реалізації (процедур, правил тощо);
- адекватність організаційних правил та засобів аудиту (зокрема, інформаційних ресурсів);
- якість імплементації вказаних вимог на найнижчих рівнях.

Разом з тим Банківською комісією виділено 2 найбільш важливі інструменти інспектування:

- аналіз інституціональної схеми;
- аналіз файлів (клієнтських) та фінансових операцій.

При цьому аналіз інституціональної схеми включає три стадії:

- розуміння запроваджених схем;
- визначення ризику залучення організацій до легалізації коштів;
- аналіз на предмет адекватності схеми здійснення фінансового моніторингу встановленому рівню ризику.

Аналіз файлів (клієнтських) та фінансових операцій здійснюється за допомогою різноманітного інструментарію, включаючи ІТ-інструменти.

Отже, Банківська комісія Банку Франції володіє різноманітним сучасним інструментарієм, що дозволяє здійснювати нагляд за кредитними

установами, зокрема банками, на високому професійному рівні, при одночасній оптимізації використаних для цього ресурсів: матеріальних та людських.

Більше того, провідні експерти стверджують, що протягом наступних 3 років зростання витрат на запобігання та протидію легалізації коштів, отриманих злочинним шляхом, та фінансуванню тероризму, може сягнути 30 %. При цьому основними компонентами, що можуть їх зумовити, є:

- моніторинг фінансових операцій;
- навчання персоналу;
- санкції.

Тобто попередні інвестиції в систему фінансового моніторингу не знизили витрати, вони перенаправили потоки витрат на постійний процес впровадження та синхронізації даної системи до загальнокорпоративної системи банківського бізнесу та найм досвідчених спеціалістів для моніторингу фінансових операцій.

Крім того, основним викликом для подальшого розвитку систем фінансового моніторингу є запровадження підходу на основі оцінки ризиків щодо всіх напрямків діяльності, а не лише при класифікації клієнтів.

Не омине цей процес і нашу країну.

За таких умов головне завдання інспекторів і банківського нагляду у сфері фінансового моніторингу в Україні – забезпечити виконання вимог Закону та нормативних документів Національного банку України, а головне забезпечити діяльність усіх без виключення банків за одними правилами («принцип справедливості»).

# I. ПРОГРАМНІ ЗАСОБИ ТА БАЗИ ДАНИХ: НЕВІД'ЄМНА СКЛАДОВА СУЧАСНОГО БАНКУ

## 1.1. Методи штучного інтелекту

Зростання рівнів різноманітних видів інформаційних потоків дуже часто зумовлено необхідністю ухвалювати рішення за таких умов: неточність, двозначність, невизначеність, нечіткість і необґрунтованість інформації. Необхідність підвищення швидкості і адекватності даного процесу вимагає залучення програмних засобів, які здатні сприяти його здійсненню. Програмні засоби, в основу роботи яких покладена виключно класична логіка, тобто алгоритми рішення відомих задач, не в змозі надавати об'єктивні рекомендації для прийняття рішень в невизначених ситуаціях. Якісне рішення цих задач під силу лише програмним системам, які здатні функціонувати подібно до людського інтелекту, а саме: мати здатність здобувати, адаптувати, модифікувати і поповнювати знання з метою вирішення задач, формалізація яких ускладнена. Такі системи належать до категорії штучного інтелекту, що здатні в залежності від поточної ситуації самі розробляти алгоритми рішення новоутворених проблем.

З моменту визнання штучного інтелекту одним із напрямків наукових досліджень (50-ті роки ХХ століття) до наших часів у полі зору розробників інтелектуальних систем знаходились завдання різноманітного спрямування. Поміж них реальних технологічних успіхів було досягнуто в процесі вирішення таких класів задач: доведення теорем, розпізнавання образів, машинного перекладу, ігрових програм, машинної творчості, експертних систем тощо. Найпоширеніше практичне застосування технології штучного інтелекту знайшли у вигляді експертних систем, комерційне впровадження яких почалося з початку 80-х років минулого сторіччя. З того часу експертні системи значною мірою використовуються в бізнесі, науці, техніці, на виробництві.



Експертна система – це програмна система, яка оперує зі знаннями в певній предметній області з метою розробки рекомендацій для вирішення проблем. Отже, ключова відмінність методології розробки експертних систем від програмних систем, які мають алгоритмічну основу, полягає у відокремленні поняття даних від поняття знань. Цю відмінність наглядно ілюструє аналогія з класичним виразом, який оприлюднив Д. Кнут у своїй роботі, присвяченій програмуванню алгоритмічних задач [13]:

$$\text{Програма} = \text{Структура даних} + \text{Алгоритми.}$$

Стосовно експертних систем цей вираз виглядає таким чином:

$$\text{Експертна система} = \text{Знання} + \text{Логічний висновок.}$$

Експертна система, що побудована за таким принципом, може узяти на себе функції, виконання яких вимагає залучення досвіду фахівця, або виконання ролі асистента для особи, що приймає рішення. В будь-якому випадку людина – фахівець певної предметної області, має нагоду отримати результати вищої якості за умови співпраці з експертною системою.

Моделювання рівнів операційного ризику комерційного банку ґрунтується на використанні байєсівського підходу і зводиться до розрахунку імовірності відхилення реальних результатів бізнесу від очікуваних внаслідок низки порушень та інших чинників. Відхилення результатів відбувається внаслідок існування причинних факторів неявного характеру. Використання теореми Байєса спрямовує рішення зазначеної задачі до визначення ймовірності існування факторів операційного ризику за наявності подій негативного характеру – інцидентів операційного ризику, що відбулися.

Такий підхід до прогнозування повторення інцидентів операційного ризику має певний проблемний момент, який пов'язаний з відсутністю однозначної методики встановлення причинно-наслідкового зв'язку між факторами та інцидентами операційного ризику. Способи розрахунку показників оцінки операційного ризику ґрунтуються на суб'єктивних рішеннях експертів стосовно існування цих зв'язків. Оскільки своєчасне

впровадження новітніх продуктів у сферу банківського бізнесу в наші часи прийнято вважати запорукою успіху, комплекс показників ризикового аналізу повинен зазнавати постійних змін. Здійснення таких змін вимагає залучення дедалі більших обсягів знань різних за фахом експертів. Зважаючи на наведену формалізацію призначення експертних систем, неважко дійти висновку стосовно доречності їхнього використання в ролі акумулятора таких знань.

Експертні системи накопичують знання завдяки інженерії знань. Інженерія знань являє собою процес отримання знань від експертів з метою подальшого представлення знань в експертній системі. Класичний варіант інженерії знань ґрунтується на обширних інтерв'ю з експертами. На початку цього процесу інженер по знаннях встановлює діалог з експертом, щоб виявити його знання. Потім інженер по знаннях представляє знання в явному вигляді для внесення в базу знань. Після цього експерт проводить оцінку експертної системи і передає критичні зауваження інженеру по знаннях. Такий процес повторюється до тих пір, доки експерт не оцінить результати роботи системи як прийнятні.

Найбільш поширений метод накопичення знань в експертних системах ґрунтується на використанні системи продукційних правил. Продукційними називаються правила, що організовані у вигляді IF-THEN структур. Частина продукційного правила, що розміщена між ключовими словами IF і THEN носить назву антецедента або лівої частини (LHS – left-hand-side) правила. На практиці застосовуються також назви “умовний елемент” та “шаблон”. Після слова THEN заноситься список дій, які повинні бути виконані відповідно до правила. Ця частина продукційного правила носить назву “консеквент”, або “права частина (RHS – Right-Hand Side) правила”.

Продукційні правила сумісно з інтерпретатором, який управляє їх активізацією залежно від наявності фактів, складають продукційну модель представлення і використання знань в експертних систем. Такі системи носять назву продукційних. У продукційних системах знання, що

представлені у формі множини правил, визначають висновки, які повинні бути зроблені (або не зроблені) у певних ситуаціях.

Застосовуючи методику продукційних правил, з'являється нагода перекласти інтелектуальний тягар з накопичення знань для оцінки операційного ризику банку на спеціалізовану експертну систему. Як приклад розглянемо організацію знань стосовно ідентифікації існування факторів операційного ризику на основі даних з форм статистичної звітності Національного банку. Зокрема, за допомогою звіту про застосування до банків заходів впливу (форма № 682) можна сформуванати знання стосовно підвищення ймовірності існування факторів операційного ризику в залежності від виду порушень. В продукційній системі ці правила можуть виглядати наступним чином:

*IF порушення нормативів ліквідності THEN зловживання посадовим становищем для отримання власної вигоди;*

*IF порушення максимального розміру кредитів, гарантій та поручительств THEN невірна оцінка кредитоспроможності клієнта;*

*IF несвоєчасне подання, приховування або перекручення встановленої НБУ звітності про валютні операції THEN невиконання своїх безпосередніх обов'язків з метою спрощення роботи;*

*IF недотримання лімітів відкритої валютної позиції THEN помилки виконання, розрахунку і обслуговування операцій;*

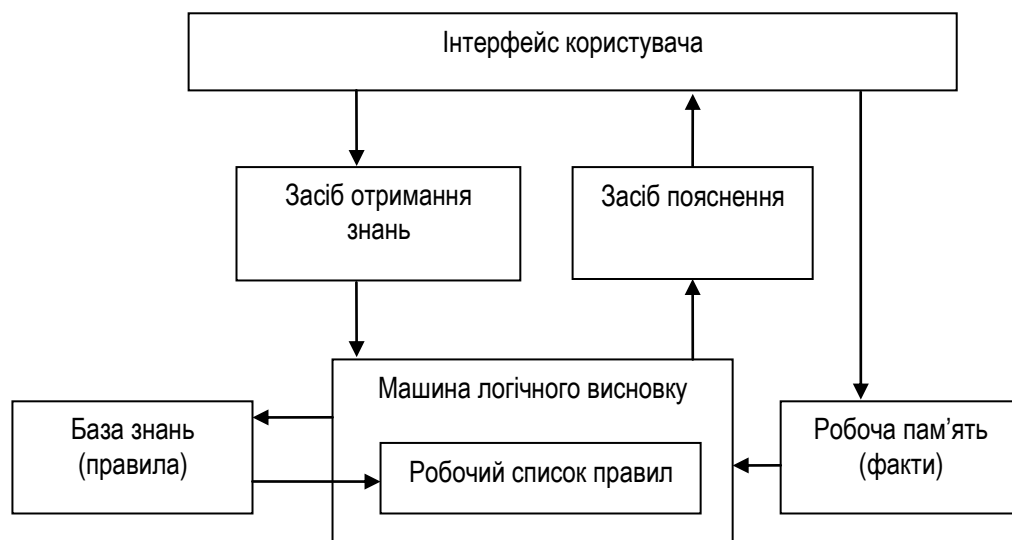
*IF Порушення банками касової дисципліни THEN несанкціоноване надання інформації та консультування клієнта співробітниками;*

*IF Порушення порядку, строків і технології виконання банківських операцій, що встановлені нормативно-правовими актами Національного банку THEN збої в роботі комп'ютерного обладнання.*

Метод представлення знань з використанням продукційних правил дозволяє фахівцям з управління операційним ризиком отримати цілком відчутні переваги. По-перше, завдяки такій організації спрощується як представлення знань, так і процес розширення експертної системи по методу

розробки крок за кроком. По-друге, такі експертні системи дозволяють легко створювати засоби пояснення за допомогою правил. Засіб пояснення дозволяє стежити за тим, активізація яких правил була застосована, що дає змогу відновити хід міркувань, які привели до певного висновку. По-третє, ця методика має суттєві аналогії з пізнавальним процесом людини. Згідно з результатами досліджень, одержаними Ньюеллом і Саймоном [29], правила є найбільш природним способом моделювання процесу вирішення задач людиною. Крім того, в процесі формалізації знань, якими володіють експерти, їм досить нескладно пояснити структуру представлення знань на основі достатньо простої схеми правил IF-THEN.

Загальну структуру експертної системи, заснованої на правилах, можна зобразити у вигляді рис. 2.



**Рис. 2. Загальна структура експертної системи**

Наведена структура наочно ілюструє основні аспекти реалізації експертних систем, тому розглянемо детально сутність та призначення її компонентів.

Інтерфейс користувача – це механізм, за допомогою якого відбувається спілкування користувача з експертною системою. Залежно від призначення системи інтерфейс користувача може використовувати простий текстовий дисплей або складний растровий дисплей з високою розподільною здатністю.

Останній зазвичай застосовується для задач моделювання, які вирішує експертна система.

Засіб отримання знань являє собою автоматизований спосіб, який дозволяє користувачу вводити знання в систему, не застосовуючи явного кодування знань за допомогою інженера по знаннях. Цей інструментальний засіб в деяких експертних системах здатний навчатися, здійснюючи автоматичне формування правил на підставі прикладів. Для формування правил в машинному навчанні застосовуються такі методи і алгоритми, як: ID3, C4.5, C5.1, штучні генетичні алгоритми і нейронні мережі.

База знань системи вміщує знання, необхідні для вирішення задач в певній проблемній області, в нашому випадку – управлінні операційним ризиком. Базу знань експертної системи, в якій знання закодовані у формі правил, називають продукційною пам'яттю. До складу дій консеквентів правил зазвичай входить додавання або видалення фактів з робочої пам'яті або досягнення результатів. Формат опису цих дій залежить від синтаксису мови програмування експертної системи.

Машина логічного висновку є програмним компонентом, який визначає, які антецеденти правил (якщо такі існують) виконуються згідно з фактами. Для цього машина логічного висновку виконує наступні дії:

- ухвалює рішення про те, яким правилам відповідають факти;
- розподіляє за пріоритетами правила, обрані до виконання;
- виконує правило з найвищим пріоритетом.

Як класичні стратегії рішення задач в експертних системах використовуються два загальні методи логічного висновку: прямий логічний висновок і зворотний логічний висновок. Загальною стратегією вирішення задач є розбиття їх на фрагменти, які можна легше доводити. При цьому системи з прямим логічним висновком управляються даними або фактами. Вони починають свою роботу з відомих початкових фактів і продовжують роботу, використовуючи правила для формування нових висновків або виконання певних дій.

Системи із зворотним логічним висновком управляються цілями або гіпотезами. Вони починають свою роботу з деякої гіпотези, або мети, яку користувач намагається довести, і продовжують роботу, відшукуючи правила, які дозволять довести істинність гіпотези. До інших методів, що використовуються для виконання конкретизованих завдань, можуть входити: аналіз цілей і засобів, спрощення задачі, перебір з поверненнями, метод «запланувати-виробити-перевірити», ієрархічне планування тощо.

Робоча пам'ять системи призначена для розміщення фактів, що стосуються поточного стану об'єкта досліджень. Факти, що знаходяться в робочій пам'яті, не взаємодіють один з одним, на відміну від правил, що зберігаються в базі знань. Якщо в робочій пам'яті є факт, який відповідає умовній частині правила, машина логічного висновку розміщує це правило в робочому списку правил. У випадку, якщо правило має декілька шаблонів, то для того, щоб правило можна було розмістити в робочому списку правил, всі ці шаблони повинні бути розпізнані як відповідні. Як умову відповідності деяких шаблонів можна вказати відсутність певних фактів в робочій пам'яті. Машина логічного висновку працює в режимі здійснення циклів «розпізнавання-дія».

Робочий список правил являє собою список правил, створений машиною логічного висновку та розташований за пріоритетами, шаблони яких відповідають фактам, що знаходяться в робочій пам'яті. Правило, всі шаблони якого розпізнані як відповідні, називається активізованим, або реалізованим. У робочому списку правил може бути одночасно присутнім декілька активізованих правил. В цьому випадку машина логічного висновку повинна вибрати залежно від пріоритету одне з правил для запуску дії.

Після завершення виконання всіх правил управління повертається до інтерпретатора команд верхнього рівня, щоб користувач міг видати командному інтерпретатору експертної системи додаткові інструкції. Роль верхнього рівня системи виконує інтерфейс користувача, який по суті є механізмом інтерпретації команд користувача.

Головною особливістю експертної системи є передбачений в ній засіб пояснення, який відображає інформацію про те, як система дійшла певного висновку. В системах, заснованих на правилах, нескладно організувати пояснення яким чином був отриманий певний висновок, оскільки хронологія активізації правил і вміст робочої пам'яті можна зберігати в стеку. Розвинені засоби пояснення можуть дати користувачу можливість ставити питання на кшталт «що», «якщо» і вивчати альтернативні шляхи формування висновків за принципом гіпотетичних міркувань.

Підсумовуючи викладені положення, можна дійти висновку, що в разі організації раціональної взаємодії між особою, яка приймає рішення з мінімізації операційних ризиків та експертною системою, можна досягти певних переваг. Маються на увазі переваги в порівнянні з необхідністю взаємодії з експертами для ухвалення прийнятних рішень. Ці переваги явно проявляються в наступних моментах:

- рівень знань експертної системи, що були скомбіновані шляхом об'єднання знань декількох експертів, ймовірно перевищуватиме рівень знань окремо взятого експерта-людини;
- знання експертної системи зберігаються протягом невизначено довгого часу на відміну від експертів-людей, які можуть піти на пенсію, звільнитися з роботи тощо;
- здатність експертної системи детально пояснити свої міркування, які привели до певного висновку, у будь-який час і з практично необмеженою кількістю повторень, сприяє підвищенню довіри до ухваленого рішення;
- експертна система гарантує об'єктивний результат в екстремальних ситуаціях, в яких людина-експерт може виявитися нездатною діяти з максимальною ефективністю через дію стресу, втоми, хвороби тощо;
- експертну систему можна використовувати як інтелектуальну навчальну програму, для розвинення у нових співробітників аналітичних навичок завдяки здійсненню поступового детального розгляду міркувань системи в процесі рішення задач.

Прийняття рішень в процесі управління операційним ризиком багато в чому залежить від об'єктивності експертних знань. Тому методика формалізації знань експертів та накопичення їх в експертній системі являє, на наш погляд, один з найбільш прийнятних на сьогодні підходів до створення програмного забезпечення підтримки дій в умовах невизначеності. Метод продукційних правил вважається класичним для формалізації інтелекту і може бути успішно застосований в інженерії знань експертів з оцінки операційного ризику комерційного банку. Отже, створення відповідної інтелектуальної системи є вкрай необхідним для сучасного банку, а її побудову доцільно починати саме з накопичення даних. Звичайно, наступні кроки в розвитку системи будуть пов'язані з новими проблемами та методами їх подолання, які в алгоритмічному програмуванні зазвичай не застосовуються. Проте успіх цього процесу здатен покласти початок такому необхідному сьогодні зниженню міри суб'єктивності рішень як в ризик-менеджменті, так і в загальній практиці управління банківською діяльністю.

## **1.2. Методи створення і ведення баз даних комерційному банку**

Один з ключових моментів створення експертної системи полягає в організації технологічної структури бази знань, системи її зберігання та доступу до її елементів. Класичним підходом вважається організація бази знань в системах, розроблених за допомогою спеціалізованої мови програмування експертних систем CLIPS. Знання в таких системах мають структуру продукційних правил і зберігаються списком у файлах текстового формату зі специфічним розширенням (.clp). Такий підхід може бути прийнятним для створення баз знань з невеликою кількістю правил. Проте для створення бази знань оцінки операційного ризику банку можливості нарощування інформаційного обсягу в такій системі істотно обмежені внаслідок високої схильності до збільшення надмірності даних.



Термін «надмірність даних» використовується для опису ситуації, коли одні і ті ж дані зберігаються в різних місцях зовнішньої пам'яті (файлах). Надмірність в експертних системах небажана з декількох причин. Перша причина – неоднозначність. Виникає ситуація, коли одне і те саме правило має різні шаблони в різних записах файлів бази знань. Ефективна робота системи вимагає відсутності подібних неоднозначностей. Друга причина – неузгодженість. У випадку, коли використовується паралельний аналіз, кожна машина логічного висновку по-різному сприймає список дій одного правила, існує висока ймовірність неузгодженості. Наприклад, якщо в системі, яка працює на одному комп'ютері, змінився консеквент правила, необхідно внести такі самі зміни в файли бази знань на інших комп'ютерах, де працюють машини логічного висновку. Такий процес називається розповсюдженням оновлень і, зазвичай, вимагає значних ресурсних витрат. Третя причина – марна праця. Дублювання існуючих антецедентів і консеквентів правил для урахування нових варіантів їх співвідношень, є марною витратою часу, сил і коштів.

В експертних системах, які використовують базу знань, сформовану як записи продукційних правил в текстових файлах, ймовірність неузгодженості, неоднозначності та марних витрат праці дуже висока. Для експертної системи оцінки операційного ризику, що вимагає використання великої кількості правил, такий підхід неприпустимий. Цієї проблеми можна уникнути, сформувавши базу знань у вигляді спеціалізованої програмної системи, що управляє процесом створення, зберігання і обробки правил – сервера баз даних. Реалізація функцій сервера баз даних покладається на особовий клас програмного забезпечення – систему управління базою даних (СУБД). По своїй суті, СУБД виконуватиме роль посередника між програмою, що виконує функції машини логічного висновку, і даними, які формують правила.

Проблему надмірності даних в структурі продукційних правил бази знань експертної системи оцінки операційного ризику СУБД вирішує завдяки

реалізації концепції інтегрованого використання даних. Під концепцією інтегрованого використання даних мається на увазі виконання наступних положень:

- по-перше, різні користувачі можуть використовувати одні й ті ж дані;
- по-друге, ці дані можуть використовуватися різними користувачами в один і той самий час.

Процес інтегрованого використання даних називається паралельним доступом або паралелізмом (concurrency). Паралелізмом необхідно керувати, інакше дані можна легко пошкодити (наприклад, якщо один користувач змінює елемент правила, який використовується в цей момент машиною логічного висновку). Тому, крім паралелізму СУБД має забезпечувати гарантії безпеки й цілісності бази даних. Користувачі системи повинні мати нагоду захистити базу знань від несанкціонованого доступу, а також відновити їх у разі будь-яких системних збоїв. Централізоване управління безпекою даних – одна з найбільш важливих особливостей СУБД.

В експертній системі, для організації бази знань якої використовується СУБД, дані продовжують фізично зберігатися у файлах. Але це не текстові файли, доступ до яких можна отримати використовуючи текстовий редактор. Ці файли утворюють загальну множину даних, доступ до вмісту якої можливо отримати виключно через СУБД. Роль СУБД в забезпеченні доступу до даних полягає в тому, щоб генерувати запити, що дозволяють використовувати функціональні можливості системи управління файлами комп'ютера, на зовнішній пам'яті якого вони розташовані. СУБД виконує роль додаткового рівня програмного забезпечення, надбудованого над системним програмним забезпеченням.

Та частина інформації з бази даних, яка потрібна певному користувачу, називається представленням (view). СУБД повинна підтримувати можливості різноманітного представлення частин загального обсягу даних. Кожне представлення бази даних – це окрема логічна структура, побудована з фізичних даних, що лежать в її основі. Щоб забезпечити інтерфейс між

фізичною пам'яттю бази даних та її різноманітними логічними версіями, СУБД, у свою чергу, повинна поділитись на декілька рівнів.

У будь-якій системі з базою даних є центральний, або так званий концептуальний рівень – логічне представлення даних системи. Концептуальний рівень повинен мати наступні характеристики:

- незалежність від того, як фізично зберігаються дані;
- повнота, тобто він повинен містити опис усіх даних, що зберігаються в системі.

Концептуальний рівень СУБД складається з усіх об'єктів бази даних, доступних користувачам. Об'єкт бази даних – це її певний логічний елемент. Залежно від типу бази даних її користувачам будуть доступні різні типи об'єктів. Концептуальний рівень СУБД є останнім рівнем представлення даних, що доступний користувачу. Користувачі навмисно усунені від розгляду питань про те, як насправді зберігаються дані на фізичному рівні.

Сукупність всіх представлень утворює так званий зовнішній рівень бази даних – інтерфейс між базою даних і її користувачами. Якщо концептуальна схема бази даних модифікується, то всі представлення, які причетні до цієї модифікації, необхідно буде змінити так, щоб вони залишались для своїх користувачів незмінними. Логічна незалежність даних полягає в усуненні користувачів від зміни логічного представлення бази даних.

Існує також інша форма незалежності даних, так звана фізична незалежність даних. Вона полягає в усуненні користувачів від змін, що відбуваються у фізичному сховищі бази даних. Фізичне сховище пересічної бази даних часто піддається оновленням і змінам з метою підвищення продуктивності та оперативного відображення змін, що відбуваються з реальними об'єктами (в нашому випадку – продукційними правилами), дані про які зберігаються в базі даних. На найнижчому рівні СУБД повинна встановити відповідність між представленням бази даних у вигляді концептуальної схеми та її фізичним представленням. Це відображення

називається внутрішнім рівнем системи з базою даних. Він виступає як інтерфейс між СУБД і операційною системою комп'ютера, на якому вона встановлена. Якщо фізичне сховище бази даних змінюється, то СУБД повинна на внутрішньому рівні знов встановити відповідність концептуальної схеми новому фізичному представленню. Сама концептуальна схема повинна залишитися незмінною.

Таким чином, СУБД складається з трьох рівнів: множини відображень концептуального рівня в представлення користувачів, самого концептуального рівня і відображення концептуального рівня у фізичне сховище. Ці три рівні називаються відповідно зовнішнім, концептуальним і внутрішнім рівнями.

Такий розподіл СУБД на рівні був запропонований як стандарт ANSI/SPARC (1978). Насправді, такий чіткий розподіл на рівні рідко використовується на практиці. Зокрема, з міркувань підвищення продуктивності, СУБД часто не звертається до засобів операційної системи комп'ютера, на якому вона встановлена, а сама виконує ті операції, які стосуються управління файлами. Слід зазначити, що упродовж еволюції СУБД було розроблено значна кількість моделей практичної реалізації приведених положень. Для визначення найбільш прийнятної з них для організації бази знань системи оцінки операційного ризику розглянемо чотири моделі, які найчастіше використовуються як базові підходи до проектування СУБД: ієрархічний, мережевий, реляційний і об'єктно-орієнтований.

У 1968 році компанія IBM запропонувала своїм клієнтам систему управління інформацією (Information Management System, IMS). Це була одна з перших спроб досягти інтегрованого управління файлами на основі баз даних і один з перших прикладів СУБД. Дані в ієрархічній базі були концептуально організовані в набори, які зв'язувалися один з одним відносинами володіння. Ієрархічні бази даних підходять для тих інформаційних систем, які природно ґрунтуються на ієрархічній моделі.

Існує ряд високопродуктивних систем, побудованих на основі IMS. Проте більшість систем неможливо реалізувати у вигляді IMS таким чином, щоб уникнути великої кількості повторень даних.

На початку 70-х років XX сторіччя Конференцією по мовах і системах даних (Conference On Data Systems Languages, CODASYL) була створена спеціальна робоча група – Database Task Group, метою якої була розробка методичних вказівок з реалізації інтегрованого підходу до управління файлами, заснованого на технології баз даних. Результатом роботи групи стала розробка мережевої моделі, яка є, по суті, модифікацією ієрархічної моделі. У мережевій моделі існують дві основні конструкції: записи і зв'язки. Зв'язок являє собою набір фізичних покажчиків, які задають відносини володіння між наборами записів. На відміну від ієрархічної моделі, у мережевій моделі немає обмеження, яке вимагає, щоб володіння задавалося тільки в одному напрямку, тому набір записів може брати участь у довільній кількості зв'язків володіння. Використовуючи мережевий підхід, можна шляхом ретельного аналізу даних усунути надмірність, і використання файлів системи стане інтегрованим. Але така інтеграція досягається за рахунок значної складності в організації доступу до вмісту файлів. Мережеві бази даних характеризуються великою кількістю наборів записів, кожен з яких містить порівняно невеликий обсяг інформації, і багато покажчиків на іншу множину записів. Навіть написання простих запитів до такої системи даних може вимагати складну навігацію від одного набору записів до інших.

Реляційна модель бази даних була вперше запропонована І. Коддом у 1970 році). Вона позитивно відрізнялася від моделей баз даних, що застосовувались до неї і, як наслідок, у 80-х роках XX сторіччя отримала загальне визнання як найбільш узгоджена й практична модель розробки СУБД. Згідно з реляційною моделлю дані на концептуальному рівні представляються у вигляді звичайних двовимірних таблиць, що складаються з рядків і стовпців. Рядки таблиць в реляційній теорії називаються кортежами, а стовпці – атрибутами. Кортежі практично відображають ті

розуміння, що у попередніх моделях відповідали записам файлів. Атрибути, у свою чергу, відображають характеристики кожного кортежу. У реляційній базі даних зв'язок між даними в різних таблицях здійснюється за допомогою значень визначених атрибутів. Реляційні системи забезпечують більш практичне середовище розробки, ніж попередні підходи. Реляційні структури даних легко розуміти і створювати. Крім того, процеси розробки прикладних програм, що їх використовують, також не належать до категорії високої складності. Тому останніми роками переважна більшість виробників сучасних СУБД у тій чи іншій мірі використовують реляційну модель. Проте у реляційної моделі є певні недоліки, які частково можуть бути усунуті завдяки використанню об'єктно-орієнтованої моделі.

Реляційний підхід до реалізації СУБД найчастіше піддається критиці за те, що він ґрунтується на ідеї пасивної множини даних. У них відсутні засоби, які дозволяють моделювати реальну поведінку даних. Крім того, його семантичні можливості також достатньо обмежені, тому буває важко представляти дійсне значення об'єктів даних. Об'єктно-орієнтована технологія баз даних намагається подолати ці обмеження. Схема об'єктно-орієнтованої бази даних складається з колекції класів. Клас є колекцією об'єктів, причому структура і поведінка об'єктів одного класу однакові. Видима структура об'єкта визначається властивостями його класу. Важливою властивістю об'єктно-орієнтованої бази даних є те, що користувачу не потрібно знати про взаємодію об'єктів. Використовуючи об'єкти і методи, можна зберігати і неодноразово використовувати не тільки структуру об'єкта бази даних, але і його поведінку. Однак практичних реалізацій об'єктно-орієнтованих СУБД на даний момент існує порівняно небагато. Незважаючи на спроби стандартизації, що відбуваються останнім часом, визначення поняття «об'єктно-орієнтована база даних» все ще недостатньо чітке в порівнянні із визначенням реляційної моделі. Це призвело до значних відмінностей у способах реалізації об'єктно-орієнтованих баз даних. Тому розробники ІТ-систем, яким добре знайома наочна реляційна модель

організації даних і пов'язані з нею відносно однорідні програмні продукти, не поспішають переходити до об'єктно-орієнтованих систем.

Підводячи підсумки аналізу моделей реалізації СУБД, можна дійти висновку, що перші два з цих підходів (ієрархічний та мережевий) представляють більше історичний, ніж практичний інтерес. Останнім часом більшість інформаційних систем включають ті або інші аспекти реляційного підходу, тому реляційні бази даних домінують на ринку інформаційних систем. Тому реляційна модель реалізації СУБД видається найбільш прийнятною для реалізації бази знань системи оцінки операційного ризику. Для підтвердження розглянемо технологічну структуру реляційної бази даних.

Як було зазначено раніше, у реляційній базі даних всі дані зберігаються у двовимірних таблицях. Для кожного об'єкта інформацію про який планується зберігати в базі даних, створюється окрема таблиця. Усі об'єкти з однаковими характеристиками зберігаються в одній таблиці. Характеристики об'єкта представляються заголовками стовпців, що описуються в заголовку таблиці. Екземпляри конкретного об'єкта представляються рядками таблиці.

Усі реляційні таблиці повинні мати заголовок. Заголовок складається з імені таблиці та імен атрибутів (стовпців), які складають дану таблицю. Кількість атрибутів визначає ступінь таблиці. У реляційній теорії обмеження на кількість атрибутів, що становлять таблицю, не накладаються, але на практиці в СУБД зазвичай вказується верхня межа для їхньої кількості. Кортєжі реляційної таблиці (рядки) утворюють її тіло. Кортєж являє собою впорядкований список атрибутів певного об'єкта. Кожен атрибут характеризується його положенням у кортєжі. Кількість кортєжів таблиці визначає її кардинальність. Кортєжі таблиці можуть зберігатися й відображатися в довільній послідовності. У більшості реляційних систем кортєжі зберігаються і відображаються в тому порядку, в якому вони заносились до таблиці.

На дані, які можна присвоювати атрибутам, накладаються певні обмеження. Це можуть бути цілі числа або послідовності символів. Область значень, які може приймати атрибут, називається його доменом. На деякому рівні поняття «домен» дуже близьке поняттю типів даних у програмуванні. Як і тип даних, домен не тільки визначає множину значень, що може приймати атрибут, але й задає діапазон допустимих операцій для кожного типу значень (наприклад, складання й віднімання чисел, розбиття і конкатенація символічних рядків тощо). При визначенні атрибуту необхідно задати для нього ім'я і домен. З цього моменту кожне значення, що привласнюється атрибуту, повинно відповідати його домену. Домени можуть бути достатньо узагальненими (наприклад, «позитивні цілі числа в діапазоні від 0 до 99999», «рядки завдовжки не більше 20 символів») або достатньо конкретними (наприклад, «депозитний рахунок»). Більшість реляційних програмних продуктів забезпечує підтримку загальних доменів у формі базових типів даних. Найрозвиненіші з них пропонують користувачам бази даних засоби, що дозволяють створювати свої власні домени.

Важливим моментом реляційної технології є те, що реляційні домени повинні бути простими, тобто складатися тільки з окремих величин. У кожному кортежі кожному атрибуту можна надати тільки одне значення. Багатозначні домени не допускаються. Якщо атрибут таблиці повинен бути багатозначний, то для його реалізації необхідно створити додаткову таблицю. Багатозначний домен не є атомарним, тобто він може містити значення, які не є простими «атомами». Під «атомами» розуміють об'єкти, які не допускають їх розклад на складові. У реляційній базі даних всі домени повинні бути атомарними.

Базова таблиця являє собою найнижчий рівень представлення даних, який доступний користувачам бази даних. Усі дані реляційної бази даних зберігаються у вигляді набору базових таблиць. Але одержувати дані і маніпулювати ними можна також за допомогою представлень. Представлення можна трактувати як логічну таблицю, яка прямо або



опосередковано одержує дані з базових таблиць. Представлення може бути просто підмножиною деякої базової таблиці. У представленні можуть знаходитися дані з декількох базових таблиць. Представлення до певної міри можуть трактуватися як базові таблиці: їх можна використовувати як для отримання даних, так і для додавання, видалення і змін значень даних.

При створенні представлення домени атрибутів визначаються базовими таблицями, звідки беруться дані для цього представлення. Створення представлення ґрунтується на визначенні списку базових таблиць, дані яких будуть покладені в його основу. Коли базова таблиця видаляється з бази даних, усі представлення, що походять від неї (повністю або частково, безпосередньо або опосередковано), логічно припиняють своє існування.

Організація фізичного рівня реляційних баз даних ґрунтується на взаємодії базових таблиць СУБД і файлів, в яких дані фізично зберігаються. Запис даних у зовнішню пам'ять і отримання їх звідти утворюють одну із задач операційної системи комп'ютера. Операційна система є набором напівперманентних програм, які забезпечують інтерфейс між прикладними програмами й апаратним забезпеченням комп'ютера. Вона надає програмам такі послуги, як управління пам'яттю й організацію вводу-виводу. Операційна система сприймає СУБД як звичайну прикладну програму, що потребує обслуговування. Основним необхідним для СУБД сервісом, який може запропонувати операційна система, є обробка фізичних файлів.

На практиці не завжди існує чітке відділення файлової системи від СУБД. Багато операційних систем не надають необхідні для СУБД можливості по управлінню файлами. З іншого боку, деякі СУБД самі ігнорують файлову систему і взаємодіють безпосередньо з програмою управління дисковою пам'яттю. Організація внутрішнього рівня реляційної системи може бути досить різноманітною. Наприклад, в одних системах під кожну створювану таблицю виділяється окремий файл системи, в інших файл системи створюється для кожного набору таблиць, що належать окремому користувачу. Важливо те, що всі ці системи виглядають як реляційна база

даних для своїх користувачів. Теоретично, вони можуть надавати користувачам однакові можливості. Проте внутрішня організація реляційних систем впливає на продуктивність інформаційної системи, що їх використовують.

У класичній схемі реляційна СУБД має два рівні взаємодії: один – з користувачем бази даних, а другий – з операційною системою комп'ютера, на зовнішній пам'яті якого зберігаються файли з даними. Користувач може розглядати і використовувати базу даних тільки як набір таблиць. Все інше від користувача приховане. Управління таблицями й блоками даних у табличному просторі відносять виключно до задач СУБД.

Підсумовуючи викладені положення, можна зазначити, що при використанні реляційної бази даних для організації бази знань системи оцінки операційного ризику можна на концептуальному рівні створити структуру таблиць для збереження антецедентів та консеквентів продукційних правил, яка дозволить позбутися надмірності даних. Структури базових таблиць забезпечать контроль за шаблонуванням частин правил. Використовуючи реляційний механізм, можна забезпечити оптимальні структури правил у вигляді представлень для передачі в машину логічного висновку. З'являється можливість покласти на СУБД рішення завдань з оптимізації розміщень елементів правил в файлах зовнішньої пам'яті, управління доступу до них та відновлення на випадок аварій в системі. Для цього на ринку реляційних СУБД існує низка продуктів, що позитивно зарекомендували себе у багатьох автоматизованих інформаційних системах, а саме: ORACLE, MS SQL SERVER, DB2, INFORMIX та інші. Отже, використання реляційної СУБД в експертній системі оцінки операційного ризику являє собою основу для створення надійної, інтегрованої та гнучкої бази знань.

### **1.3. Основні засади автоматизації банківських процесів**

Інструментальні засоби розробки програмних систем утворюються поєднанням мови програмування з набором допоміжних програм. Практично всі інструментальні засоби, що використовуються в процесі розробки експертних систем, використовують методологію автоматизації проектування на основі прототипів. По відношенню до програмного забезпечення термін “прототип” означає працюючу модель програми, яка функціонально еквівалентна підмножині кінцевого продукту. Ідея використання прототипів полягає в розробці на ранній стадії роботи проекту спрощеної версії кінцевої програми, яка була б в змозі послужити доказом продуктивності основних ідей, покладених в основу проекту. Тобто прототип повинен бути здатний вирішувати одну з характерних задач для заданої області застосування. На основі аналізу досвіду роботи з прототипом розробники можуть уточнити вимоги до основних функціональних характеристик експертної системи. Працездатність прототипу може послужити наочним доказом можливості рішення проблем за допомогою створюваної системи ще до того, як на її розробку будуть витрачені значні засоби.

Процес розробки експертної системи, як правило, складається з послідовності окремих етапів, упродовж яких нарощуються можливості системи, причому кожний з етапів поділяється на фази: проектування, реалізація, компоновка та тестування. В результаті, після завершення чергового етапу, утворюється система, яка здатна справлятися з більшими по складності варіантами проблеми. На відміну від експертних систем, при створенні більшості програмних продуктів інших видів, використовується інша модель процесу: спочатку розробляється специфікація продукту, потім виконується планування, проектування компонентів, їх реалізація, компоновка комплексу та тестування кінцевого варіанта. Той факт, що при розробці експертних систем існує можливість спочатку побудувати та всебічно випробувати прототип, дозволяє уникнути безлічі переробок в

процесі створення робочої версії системи. Однак слід зазначити, що технологія послідовного нарощування функціональних можливостей містить в собі проблему інтеграції нових функцій системи з функціями, що були реалізовані в попередніх варіантах. Тому інструментальні засоби розробки експертних систем від початку створювалися на основі модульного представлення знань, з урахуванням необхідності подолання виникаючих при цьому ускладнень.

Слід зазначити, що розробка та впровадження експертної системи оцінки операційного ризику неодмінно зіткнеться з проблемами, характерними для процесу створення експертних систем. Систематизація цих проблем дозволить виробити практичні рекомендації для їх успішного подолання. Один з варіантів систематизації цих проблем та способів їх подолання представлений в табл. 1.

Успіх у розробці експертної системи з оцінки операційного ризику багато в чому залежить від вибору інструментальних засобів. Практика вибору інструментальних засобів для побудови експертної системи ґрунтується на зіставленні характеристик проблем, що має вирішувати експертна система, та необхідних функціональних можливостей інструментального комплексу.

При цьому рекомендується дотримуватись наступних загальних правил:

- слід вибирати інструмент зі ступенем сервісної насиченості, який не перевищує необхідного рівня для вирішення даної задачі;
- вибір інструментарію повинен визначатися в першу чергу характеристиками задачі, яку вирішуватиме експертна система, а не іншими сторонніми обставинами (наприклад, тим, що якийсь інструмент вже є в наявності або знайомий розробнику краще за інших);
- якщо успіх проекту залежить від терміну розробки, то слід вибирати інструментальне середовище з вбудованими засобами формування

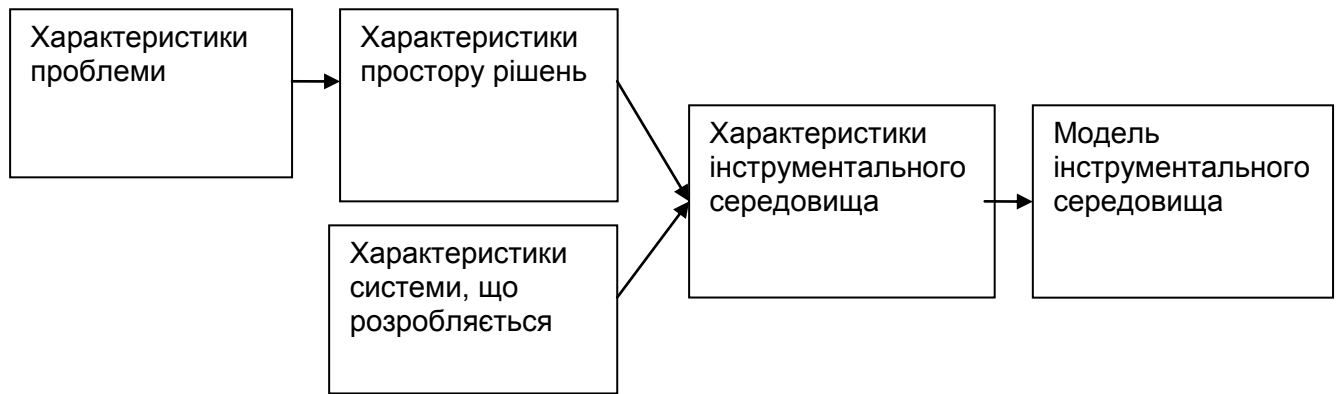
- пояснень та елементів користувальницького інтерфейсу, оскільки їх розробка найбільш трудомістка;
- необхідно намагатися якнайшвидше провести випробування нового інструментального середовища на реальних даних.

Таблиця 1

### Характерні проблеми, що виникають при розробці експертних систем

Сутність проблеми	Причина проблеми	Подолання проблеми
Знання, що стосуються предметної області, дуже тісно переплетені з іншими частинами системи	Складність відокремлення знань предметної області від знань загального застосування	Покласти в основу організації бази знань сукупність продукційних правил, збережених в реляційній базі даних
База знань, яка сформувалася в процесі опиту експертів, виявляється неповною настільки, що не дозволяє вирішувати потрібні задачі	Відсутність в базі знань фундаментальних концепцій предметної області, або ці концепції представлені з помилками	Послідовно нарощувати об'єм бази знань, починаючи з фундаментальних понять – це дозволить ще на ранніх стадіях розробки виявити вказану проблему
Середовище розробки позбавлене вбудованих засобів формування функцій пояснення експертної системи	Додавання таких функцій у вже спроектовану систему являє собою надскладну задачу	Слід піклуватися про прозорість експертної системи з перших кроків її розробки
Система може містити надмірну кількість дуже специфічних правил, що призводить до уповільнення роботи системи та ускладнює управління нею	Складність трансформації знань предметної області в набір продукційних правил	Слід уникати надто специфічних і надто загальних правил, необхідно прагнути знайти компроміс між ефективністю правил та їх зрозумілістю

Процес рішення питання вибору можна представити схематично, як показано на рис. 3.



**Рис. 3. Типова схема вибору інструментальних засобів для розробки експертної системи**

Найважливішим питанням в процесі вибору інструментального середовища є питання стосовно способу визначення характеристик проблеми, для вирішення якої призначається експертна система. Ці характеристики можна звести до 4 основних категорій:

1) *малий простір рішень, надійні дані та знання*. Передбачається, що кількість альтернатив, які слід брати до уваги при пошуку рішення, є невеликою, всі дані є достовірними та істинність правил не викликає сумнівів. Для вирішення проблем цієї категорії можна скористатися готовими рішеннями, тобто раніше створеною оболонкою на базі експертної системи, що вирішувала аналогічну проблему в іншій предметній області;

2) *ненадійні дані або знання*. Якщо дані або знання ненадійні, то існує небезпека, що дані, які вводяться в систему, не є достовірними, а правила в базі знань не мають однозначності. В цьому випадку в експертній системі потрібно комбінувати інформацію від декількох джерел та використовувати логіку нечітких міркувань;

3) *великий факторизований простір рішень*. Простір пошуку можна назвати факторизованим, якщо існує можливість розділити його на декілька незалежних підпросторів, які можна обробляти окремо. Причому для різних підпросторів можуть бути використані різні множини правил або окремі підмножини однієї і тієї ж множини правил. Зазвичай таке розбиття

виконується на рівні проблеми, тобто велика загальна проблема розбивається на декілька дрібніших. Успіх в досягненні головної мети, таким чином, оцінюється за сукупністю успіхів в досягненні незалежних цілей;

4) великий нефакторизований простір рішень. Простір рішень може виявитися нефакторизованим, якщо задача допускає вироблення рішення будь-якого компонента тільки в контексті всього проекту. Загальний підхід до роботи у великому просторі пошуку полягає в тому, щоб послідовно розглядати його на різних рівнях абстракції. Тобто потрібно використовувати варіанти описання простору з різним рівнем урахування деталей. Рішення проблеми таким методом часто називають низхідним уточненням.

З'ясувавши характеристики проблем, для вирішення яких розробляється експертна система, можна визначитися з властивостями простору рішень. Потім вони розглядаються спільно з передбачуваними характеристиками системи: моделлю подання знань, напрямком логічного висновку, способом формування пояснень. В результаті виявляються бажані характеристики інструментального середовища, які дозволяють підібрати потрібну модель інструментального середовища. Як показує практика, більшість розробників явно або неявно використовують саме такий підхід при створенні експертних систем.

В процесі вибору інструментального середовища важливе значення мають також наступні аспекти:

- наскільки просте середовище у використанні;
- як швидко розробники експертної системи зможуть оволодіти методикою роботи в цьому середовищі;
- яку підтримку готова надавати фірма-розробник середовища;
- яка буде загальна вартість середовища з урахуванням прямих і непрямих витрат.

В матеріалах, що описують програмні засоби з розробки експертних систем, можна зустріти твердження, що даним інструментом «може успішно користуватися програміст, малознайомий з технологіями штучного

інтелекту» або навіть непрограміст. Проте практика показує, що це не так. Практичне оволодіння типовими інструментальними засобами проектування експертних систем не поступається за складністю оволодінню новою мовою програмування.

Як правило, типове середовище розробки експертних систем підтримує чотири режими роботи:

- підготовка та редагування бази знань;
- використання бази знань для виконання консультацій, тобто прогін програми;
- виявлення та усунення помилок на стадії компіляції;
- виявлення та усунення помилок на стадії виконання.

Як показав досвід, навіть досвідчені програмісти важко засвоюють методику сумісного використання цих режимів в процесі проектування експертної системи. Це пов'язано, перш за все, з тим, що стандартна стратегія розробки бази знань передбачає постійне нарощування її об'єму. Тому інженеру по знаннях доводиться виконувати інтерактивні процедури поповнення бази знань значно частіше, ніж звичайному програмісту виконувати розширення функцій програми.

За своїм призначенням та функціональними характеристиками інструментальні засоби, що використовуються в програмуванні експертних систем, можна розділити на наступні категорії:

1) *оболонки експертних систем*. Системи типу оболонки експертних систем створюються, як правило, на основі експертних систем, які достатньо добре зарекомендували себе на практиці. В процесі створення оболонки з системи-прототипу видаляються компоненти, які є специфічними для області її безпосереднього застосування, та залишаються ті, що не мають вузької спеціалізації. В загальному сенсі, оболонки експертних систем створюються з метою дозволити непрограмістам скористатися результатами роботи програмістів, що вирішували аналогічні проблеми. Клас цих інструментальних засобів орієнтований на достатньо вузький клас задач,



проте, незважаючи на обмеження, цей тип експертних систем прогресує. Наприклад, оболонка M.4 може функціонувати під управлінням будь-якої з операційних систем персональних комп'ютерів, підключатися до баз даних, включати фрагменти програмного коду мовами Visual BASIC та Visual C++;

2) *мови описання продукційних правил.* Являють собою ефективний засіб швидкого створення прототипів експертних систем. Вони дозволяють забезпечити гнучкість процесу розробки, мінімізацію матеріальних витрат та термінів виконання проекту. Одним з найвідоміших представників таких мов є OPS5. Для цієї мови характерний порівняно простий синтаксис та механізм активізації правил. Однак для мови OPS5 характерною ознакою є труднощі при реалізації деяких типів структур управління ходом виконання. Наприклад, до них можна віднести рекурсивні та ітераційні цикли, оскільки вони вимагають серйозного ускладнення описання процесу обробки правил. Розробники мов, подібних OPS, завжди вимушені шукати компроміс між наочністю засобів мови програмування та ефективністю виконання програмного коду;

3) *об'єктно-орієнтовані мови.* Мовами об'єктно-орієнтованого програмування створюється програмне середовище для організації знань в термінах декларативного представлення об'єктів предметної області. Усі дії, пов'язані з процедурною стороною рішення проблем, розподіляються між цими об'єктами, які мають в своєму розпорядженні власні процедури та можуть спілкуватися один з одним за допомогою інтерфейсів передачі повідомлень. Корисним аспектом об'єктно-орієнтованого програмування є можливість інтеграції символічних обчислень в операційне середовище, яке базується на засобах графічного інтерфейсу. Оснащення експертної системи цими засобами дозволяє краще представити користувачу процеси, що відбуваються в системі. До недоліків використання об'єктно-орієнтованого стилю в програмуванні експертних систем можна віднести складність в організації співвідношення програмних об'єктів з абстрактними поняттями та категоріями предметної області;

4) *мови логічного програмування.* Типовою мовою логічного програмування експертних систем є PROLOG. Для цього PROLOG володіє достатньо корисними можливостями, а саме: вбудований в PROLOG режим управління приблизно відповідає стратегії зворотного логічного висновку; індексовану базу даних фраз мови PROLOG можна використовувати для представлення правил; рекурсивні структури даних (графи та дерева) можна організувати за допомогою фраз мови PROLOG; універсальний механізм зіставлення мови PROLOG дозволяє виконувати зіставлення даних та шаблонів, що включають змінні; мовні засоби PROLOG дозволяють програмісту розробити власний механізм обробки невизначеності. Проте практика застосування ідей логічного програмування в експертних системах не позбавлена недоліків. Зокрема, синтаксичні та семантичні обмеження, що присутні в стандартних версіях PROLOG, не були подолані ані в системах MECNO та PLANNER, ані в інших системах, що базуються на аналогічній ідеології;

5) *середовище програмування з підтримкою декількох парадигм.* Засоби цієї категорії включають декілька програмних модулів, що дозволяє комбінувати в процесі розробки експертної системи різні стилі програмування, вибираючи відповідні поєднання різних методів. Причиною їх створення стали результати роботи експертних систем з різними схемами представлення знань та логічного висновку. Виявилось, що кожна з них має свої слабкі сторони. В результаті їх аналізу логічним чином була сформована ідея об'єднання методик в єдине середовище, в якому переваги одних компенсують недоліки інших. Одним з перших багатофункціональних середовищ штучного інтелекту став відповідний продукт, що має назву LOOPS (Bobrow and Stefik, 1983). В ньому в рамках єдиної архітектури обміну повідомленнями були об'єднані чотири парадигми програмування: процедурно-орієнтоване програмування; програмування, орієнтоване на правила; об'єктно-орієнтоване програмування; програмування, орієнтоване на дані. Основу системи складає об'єктно-орієнтована парадигма. В рамках її

модулів можна комбінувати модулі середовища, що підтримують різні стилі програмування. Такий стиль об'єднання парадигм реалізований в мові CLIPS;

б) *додаткові модулі програмування експертних систем.* Засоби цієї категорії є автономними програмними модулями, які призначені для виконання специфічних задач в рамках вибраної архітектури експертної системи. Під додатковими модулями розуміються корисні програмні розробки, які можна виконувати разом з основним програмним додатком. Як правило, такі програми реалізують деякі спеціальні функції, підключаючи їх, начебто, з-за меж системи. Причому звернення до таких функцій не потребує додаткового програмування в основному додатку.

В міру збільшення складності проекрованої системи відбувається збільшення об'єму бази знань, додавання до розгляду різного роду невизначеностей, включення в роботу системи додаткових режимів. Тому стратегія проектування вимагає від розробників дедалі більш ретельної попередньої підготовки. Крім того, можна виділити інші характерні причини складності вибору інструментального середовища розробки експертних систем:

- більшість розвинених середовищ розробки надто дорогі для того, щоб купувати їх для проведення порівняльного аналізу;
- час, необхідний для освоєння навиків роботи з системою та виявлення її сильних та слабких сторін, надто великий, тому складно проводити порівняння конкуруючих моделей на практиці;
- термінологія, яку застосовують в документації розробники різних систем, істотно відрізняється, тому проводити їх порівняння на основі технічної документації достатньо важко.

Останнє зауваження справедливе відносно більшості програмних продуктів, що пропонуються на ринку. Коли ж йдеться про програмні засоби, що пов'язані з областю експертних систем, то новизна та незвичність термінології ще більш посилює проблему. Вже давно в середовищі фахівців

існує думка, що порівняння конкуруючих систем одного класу можна виконувати тільки після ретельного вивчення їх на практиці.

Слід зазначити, що будь-які інструментальні засоби потребують адекватної методології користування ними. В літературних джерелах, присвячених програмуванню, для виразу рівня адекватності часто застосовується поняття «стиль програмування». Дотримуючись загальноприйнятого стилю програмування, можна уникнути небезпеки зробити програмну систему нежиттєздатною ще до закінчення етапу розробки проекту. Тому доречно буде розглянути загальні рекомендації, які визначають стиль програмування експертної системи оцінки операційного ризику:

1. Задача, яку передбачається вирішувати за допомогою експертної системи, повинна бути повністю під силу експерту-людині.
2. Задача повинна бути чітко сформульована. Краще створити систему, яка зможе надійно вирішувати обмежену задачу, ніж систему, що претендує на вирішення широкого класу задач, проте дає вірне рішення лише час від часу.
3. Починаючи з першої стадії роботи над системою, необхідно визначити, як вона буде вдосконалюватися та окреслити межі, яких вона повинна досягти в процесі еволюції.
4. Слід ретельно відпрацювати поведінку системи на наборі окремих випадків та організувати бібліотеку таких випадків. Тобто приклади, які застосовувались на етапі проектування, повинні бути репрезентовані.
5. Потрібно відділити ті знання, які є специфічними для предметної області, від знань, які стосуються загальної методики рішення проблем. Бажано, наскільки це можливо, спростити машину логічного висновку в системі.

6. Необхідно на самих перших стадіях проектування системи розробити однозначні угоди про оформлення програм. Це надасть їй одноманітний вигляд.
7. Бажано поступатися продуктивністю програми, якщо це зробить її зрозумілішою та спростить її супровід. Це необхідно, оскільки в роботі інтерактивної експертної системи велика частина часу йде на діалог з користувачем та звернення до баз знань.
8. Як тільки постане питання про розробку нового прототипу системи, від попереднього необхідно відмовитися. Багато проектів зазнали невдачі лише тому, що їх автори не змогли позбавитися прихильності до першого варіанта реалізації власних ідей. Звичайно, в процесі розробки нового прототипу потрібно враховувати досвід створення попереднього, але тільки досвід, а не програмний код.
9. Розробка експертної системи, яка буде здатна успішно працювати, вимагає наполегливості і терпіння професійного програміста, залучення до роботи досвідченого експерта у відповідній області та певного рівня примусу з боку керівництва.

Наведений перелік рекомендацій, звичайно, не можна вважати вичерпним, зважаючи хоча б на той факт, що інструментальні засоби постійно знаходяться в процесі розвитку. Проте вони можуть претендувати на звання фундаментальних в процесі створення програмних продуктів класу експертних систем. Програміст, який буде дотримуватись цих рекомендацій, відповідно, буде дотримуватись такого стилю програмування, що дозволить максимізувати ймовірність успіху в розробці експертної системи оцінки операційного ризику комерційного банку.

## **II. БАЗОВІ ЗАСАДИ ПРОВЕДЕННЯ ПЕРЕВІРКИ БАНКУ З ПИТАНЬ ФІНАНСОВОГО МОНІТОРИНГУ**

Основні аспекти перевірки:

- відповідність внутрішньобанківської документації з питань фінансового моніторингу;
- відповідність кадрового забезпечення;
- відповідність програмного забезпечення;
- якість виявлення операцій;
- якість проведення ідентифікації;
- якість навчання;
- якість аудиту.

Якщо з перевіркою відповідності внутрішніх положень, кадрового забезпечення при дещо формалізованому підході все зрозуміло, то питання якості навчання, аудиту, програмного забезпечення втрачають сенс без оцінки якості виявлення операцій. Якщо якість виявлення низька – значить, якість навчання низька і якість програмного забезпечення низька.

Теж саме стосується й ідентифікації – характер операцій клієнтів, їх відповідність можна достеменно встановити за результатами аналізу операцій.

Тобто наріжним каменем є операції клієнтів, а для перевірки – оцінка якості виявлення операцій, з якої випливає решта оцінок.

### **Оцінка якості виявлення операцій**

Інспектор здійснює вибір клієнтів, аналізуючи обсяги операцій, форму власності, форму господарювання, вид економічної діяльності. Після цього здійснює запит на надання виписок за рахунками обраних клієнтів, як правило на паперових носіях, або в електронному вигляді, переглядаючи операції на екрані. Форма ж інформації на паперових носіях та на екрані – одна і та ж сама.

Далі досліджуються первинні документи – документи дня і документи каси. Виходячи з того, що за умови опрацювання кожної сторінки паперового документу фахівець може витратити цілий рік, то об'єктивно перевірка є вибірковою. Частка таким чином опрацьованих документів є мінімальною, тому і похибка в оцінці – максимальна. Не попався «важкий» період, не попався «ризиковий з точки зору фінансового моніторингу» клієнт.

Тому виникає необхідність розробки відповідного ІТ-інструмента, який дозволить підвищити якість перевірки. Запровадження відповідних заходів здійснюється у декілька етапів:

**I етап.** Формування запиту на надання електронних масивів інформації:

- книга відкритих/закритих рахунків;
- паспорти (реєстраційні картки) клієнтів;
- оборотно-сальдові відомості (щомісячні);
- касові операції.

**II етап.** Вибір рахунків за книгою відкритих та закритих рахунків:

- вибір рахунків, відкритих у період інспектування.

**III етап.** Вибір рахунків за оборотно-сальдовими відомостями:

- розрахунок коефіцієнта відношення максимальних та мінімальних оборотів;
- вибір рахунків клієнтів, за якими були значні відхилення за сумами оборотів;
- розрахунок коефіцієнта відношення середньоденних оборотів до залишків на рахунках клієнтів;
- вибір рахунків клієнтів, за якими були значні відхилення середньоденних оборотів до залишків.

**IV етап.** Вибір рахунків за паспортами (реєстраційними картками) клієнтів:

- вибір рахунків клієнтів з найбільш ризиковими видами діяльності з врахуванням оборотів та відхилень за ними;

- вибір рахунків клієнтів за формами власності, за організаційно-правовими формами господарювання з врахуванням оборотів та відхилень за ними;
- вибір рахунків нерезидентів (за кодом «характеристика клієнта»);
- вибір рахунків клієнтів-суб'єктів господарювання, зареєстрованих в державних органах у період інспектування та період, що передує йому на 3 місяці.

**V етап.** Формування запиту на надання електронних масивів інформації:

- за операціями за рахунками, відібраними на II, III та IV етапах;
- витяг з реєстру операцій, які підлягали фінансовому моніторингу.

**VI етап.** Аналіз касових операцій:

- вибір та аналіз операцій за переліком касових символів, які передбачають додаткове вивчення;
- вибір інших касових операцій, яким можуть бути властиві ознаки внутрішнього та обов'язкового фінансового моніторингу (валютні обмінні операції, грошові перекази тощо).

**VII етап.** Аналіз операцій за рахунками клієнтів з метою виявлення операцій, яким притаманні ознаки обов'язкового фінансового моніторингу:

- аналіз операцій за рахунками, відкритими у період інспектування, з метою виявлення операцій, яким притаманні ознаки обов'язкового фінансового моніторингу – 4020, 4040 (*«зарахування/списання грошових коштів на рахунок/з рахунку юридичної особи, у випадку, якщо операції на зазначеному рахунку не проводилися з моменту його відкриття»*);
- аналіз операцій за рахунками клієнтів, зареєстрованих у період інспектування та період, що передує йому на 3 місяці, з метою виявлення операцій, яким притаманні ознаки обов'язкового фінансового моніторингу – 4010, 4030 (*«зарахування/списання грошових коштів на рахунок/з рахунку юридичної особи, період діяльності якої не перевищує трьох місяців з дня її реєстрації»*);
- аналіз операцій за рахунками клієнтів, з метою виявлення операцій, яким притаманні ознаки обов'язкового фінансового моніторингу – 1031, 1032



- («зарахування на рахунок коштів у готівковій формі з їх подальшим переказом того самого або наступного операційного дня іншій особі»);*
- аналіз операцій за рахунками клієнтів, з метою виявлення операцій, яким притаманна ознака обов'язкового фінансового моніторингу – 5010, *(«проведення фінансових операцій з цінними паперами на пред'явника, не розміщеними в депозитаріях, придбання цінних паперів за готівку»);*
  - аналіз операцій за рахунками клієнтів, з метою виявлення операцій, яким притаманні ознаки обов'язкового фінансового моніторингу – 5020-5060 *(«виплата/отримання страхового відшкодування/страхової премії; виплата виграшу»);*
  - аналіз операцій за рахунками клієнтів, з метою виявлення операцій, яким притаманні ознаки обов'язкового фінансового моніторингу – 3010-3051 *(«операції з грошовими коштами, цінними паперами, коли хоча б одна із сторін має відповідну реєстрацію, місце проживання чи місцезнаходження в країні (на території), яка не бере участі у міжнародному співробітництві у сфері запобігання та протидії легалізації (відмиванню) доходів, одержаних злочинним шляхом, та фінансуванню тероризму»);*
  - аналіз операцій за рахунками клієнтів з метою виявлення операцій, яким притаманні ознаки обов'язкового фінансового моніторингу – 1030 *(«переказ коштів у готівковій формі за кордон з вимогою видати одержувачу кошти готівкою»);*
  - аналіз операцій за рахунками клієнтів, з метою виявлення операцій, яким притаманні ознаки обов'язкового фінансового моніторингу – 2030 *(«переказ коштів на рахунок, відкритий у фінансовій установі в країні, що віднесена Кабінетом Міністрів України до переліку офшорних зон»)*

**VIII етап.** Аналіз операцій за рахунками клієнтів з метою виявлення операцій, яким притаманні ознаки внутрішнього фінансового моніторингу:

- аналіз операцій клієнтів за рахунками, відкритими у період інспектування, з метою виявлення операцій, яким притаманна ознака внутрішнього

- фінансового моніторингу – 900 (*«інші фінансові операції, для яких у суб'єкта первинного фінансового моніторингу виникають підстави вважати, що фінансова операція проводиться з метою легалізації (відмивання) доходів або фінансування тероризму»*), таких як виплата фінансової допомоги, значні суми отримання готівки на господарські потреби тощо;
- аналіз операцій клієнтів за рахунками, відкритими у період інспектування, з метою виявлення операцій, яким притаманні ознаки внутрішнього фінансового моніторингу – 301-325 (*«виявлення неодноразового здійснення фінансових операцій, характер яких дає підстави вважати, що метою їх здійснення є уникнення процедур обов'язкового фінансового моніторингу»*);
  - аналіз операцій клієнтів за рахунками, відкритими у період інспектування, з метою виявлення операцій, яким притаманні ознаки внутрішнього фінансового моніторингу – 201-212 (*«невідповідність фінансової операції діяльності юридичної особи, що встановлена статутними документами цієї особи»*);
  - аналіз операцій клієнтів за рахунками, відкритими у період інспектування, з метою виявлення операцій, яким притаманні ознаки внутрішнього фінансового моніторингу – 101-127 (*«заплутаний або незвичний характер фінансової операції, яка не має очевидного економічного сенсу або очевидної законної мети»*).

**ІХ етап.** Перевірка внесення виявлених операцій до реєстру:

- формування переліку операцій, яким притаманні ознаки внутрішнього та обов'язкового фінансового моніторингу, за результатами аналізу, проведеного на VI, VII, VIII етапах;
- перевірка наявності інформації у реєстрі операцій, які підлягали фінансовому моніторингу, за сформованим переліком;
- визначення причин невнесення інформації до реєстру щодо їх обґрунтованості.

**X етап.** Формування вихідних форм за результатами проведення вибору, аналізу операцій згідно з планом.

**XI етап.** Формування звіту (дповідної записки) за результатами проведеної роботи.

**XII етап.** Обговорення результатів проведених заходів щодо підвищення ефективності перевірок за допомогою обробки даних, що надаються установами банків в електронному вигляді, визначення шляхів подальшого впровадження та розвитку.

### III. АВТОМАТИЗОВАНИЙ ІНСТРУМЕНТ ІНСПЕКТОРА

Запропоновані заходи щодо підвищення ефективності перевірок з питань дотримання вимог законодавства України, яке регулює відносини у сфері запобігання легалізації доходів, отриманих злочинним шляхом, за допомогою обробки даних, що надаються установами банків в електронному вигляді (далі – заходи автоматизації перевірок) спрямовані на виконання таких завдань:

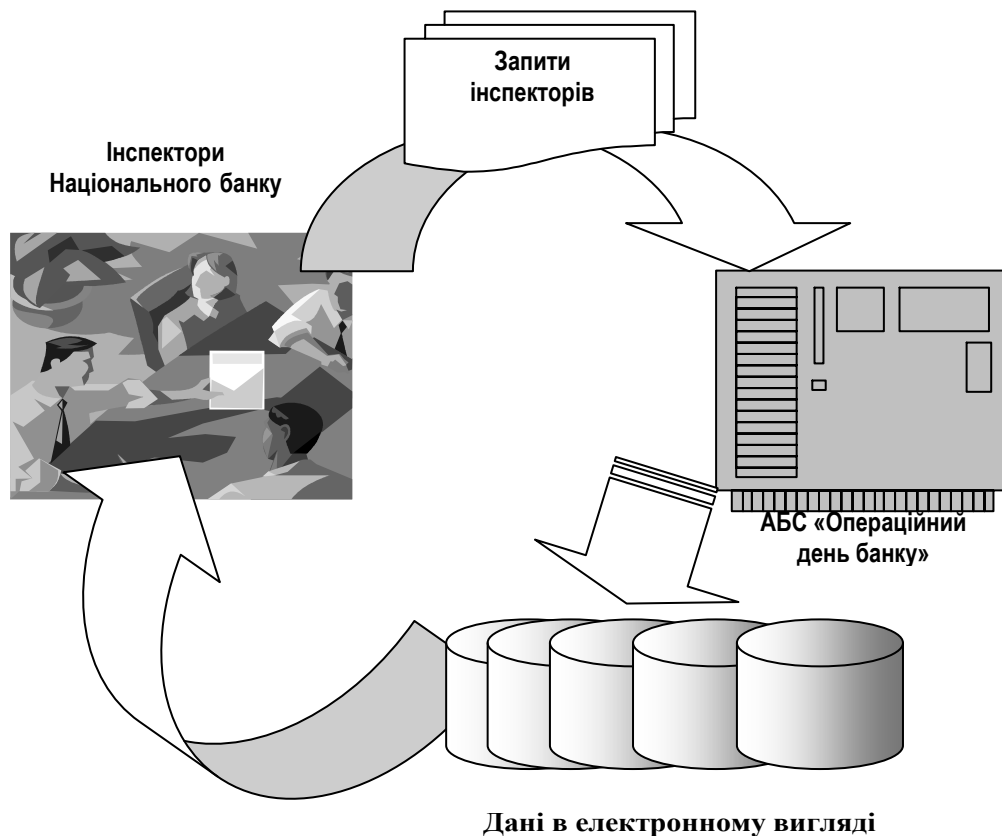
- підвищення ефективності та адекватності оцінок діяльності установи, спрямованої на виконання вимог законодавства України, яке регулює відносини у сфері запобігання легалізації доходів, отриманих злочинним шляхом;
- підвищення продуктивності виконання своїх обов'язків інспекторами Національного банку під час проведення перевірок з питань дотримання вимог законодавства України, яке регулює відносини у сфері запобігання легалізації доходів, одержаних злочинним шляхом (далі – перевірок), за допомогою ефективного використання даних, що надаються банківськими установами в електронному вигляді, автоматизації процесів їх обробки та аналізу, автоматизації процесу генерації таблиць, схем, робочих документів та інших вихідних форм, в яких відображено результати перевірки.

Наслідком запровадження запропонованих тут заходів є вивільнення інспекторських ресурсів та спрямування їх на підвищення якості аналізу та оцінки діяльності банківської установи, складання адекватних проміжних та узагальнених висновків і рекомендацій за результатами перевірки.

Очевидно, що перехід від інформації, яка надається установами банків під час проведення перевірок, на паперових носіях, на користь даних в електронному вигляді, надасть широкі можливості щодо способів виявлення та попередження порушень законодавства, яке регулює відносини у сфері

запобігання легалізації злочинних доходів. Крім того, обсяги інформації, що вимагається інспекторами під час перевірки у разі надання її в електронному вигляді, можуть бути значно більшими.

У разі дотримання запропонованих тут заходів закономірним буде запровадження алгоритмів обміну інформацією визначеного формату між інспекторами та установою банку, яка перевіряється (рис. 4).



**Рис. 4. Порядок обміну інформацією під час проведення**

Формування інспекторами електронних запитів визначеного формату, дозволить банкам здійснювати автоматизовану їх обробку та виконання – надання необхідної електронної інформації, необхідної для проведення перевірки.

Таким чином, запропоновані заходи сприятимуть запобіганню зайвого відволікання працівників установи банку від виконання своїх обов'язків щодо пошуку та підготовки інформації для інспекторів.

Слід зауважити, що запровадження зазначених заходів вимагатиме вибіркового звіряння наданих в електронному вигляді даних з інформацією на паперових носіях, завіреною підписами посадових осіб установи банку (виконавців) та/або відповідними печатками (штампами).

Для посилення відповідальності на інформацію, яка підлягає обміну за схемою «запит-дані» має накладатися електронний підпис інспектора, що склав запит, з одного боку, і посадових осіб установи банку та/або виконавців, які підготували електронні дані, з іншого.

Основною передумовою, яка дозволяє вважати реалістичним швидке та маловитратне запровадження даних заходів, є те, що за останні роки усі банки намагаються якомога більше інформації про операції зберігати в електронному вигляді та обробляти її за допомогою відповідних програмних комплексів. У деяких банків чи не всі процеси знаходять своє відображення в електронному вигляді. Не є виключенням відображення діяльності установ банків щодо фінансового моніторингу та ідентифікації.

Тому вимоги інспекторів надавати інформацію в електронному вигляді у відповідному, нескладному для виконання, форматі, для банків не будуть складною задачею.

Можливості запровадження даних заходів є майже необмеженими, але на даному етапі пропонуємо зміни до способів проведення перевірок за такими напрямками:

- оцінка якості виявлення операцій, які підлягають фінансовому моніторингу;
- оцінка повноти формування реєстру операцій, які підлягають фінансовому моніторингу, та своєчасності надсилання інформації до уповноваженого органу;
- оцінка якості проведення ідентифікації та повноти ведення анкет;
- оцінка якості підготовки персоналу.

Причини, які можуть негативно впливати на запровадження заходів автоматизації перевірок:

1. Відсутність у інспекторів елементарних відомостей щодо загальних принципів автоматизації операцій банку, збереження даних в електронному вигляді, порядку їх обробки та ін.

2. Відсутність необхідних знань щодо використання великих електронних масивів стандартними засобами обробки інформації (напр., Microsoft Exel, Access тощо).

3. Відсутність спеціалізованого програмного забезпечення, яке може здійснювати обробку електронних даних більш складними алгоритмами.

Для подолання негативних впливів пропонуємо запроваджувати заходи з автоматизації перевірок у декілька етапів:

*I етап.* Ознайомлення з концепцією запропонованих змін зацікавлених осіб з метою її обговорення та вдосконалення на початковому етапі та подальшого просування, в тому числі шляхом розробки та затвердження відповідної нормативно-правової бази, створення робочої групи.

*II етап.* Розробка відповідних алгоритмів проведення перевірки.

*III етап.* Розробка базового програмного забезпечення.

*IV етап.* Підготовка персоналу щодо роботи з даними в електронному вигляді, та використання програмного забезпечення.

*V етап.* Визначення групи осіб, відповідальних за постійне поновлення алгоритмів роботи, програмного забезпечення тощо.

*VI етап.* Розробка та використання інформаційного ресурсу (внутрішнього web-сайту Національного банку) з обмеженим доступом з метою оперативного обговорення визначених алгоритмів роботи, роботи програмного забезпечення, доступу до оновленого програмного забезпечення, порядку його використання.

### **Автоматизація перевірок (приклади, конкретні випадки)**

Конкретизуємо зміни до способів проведення перевірок якості виявлення операцій, які підлягають фінансовому моніторингу, на конкретних прикладах.

Слід зауважити, що запропоновані нижче заходи з автоматизації перевірок підлягають неодноразовому обговоренню та доопрацюванню фахівцями Національного банку (робочою групою), в тому числі з точки зору дотримання правових норм, тому питання дотримання таких норм у кожному конкретному випадку зараз не підніматимуться.

#### *Підготовка до проведення перевірки*

Під час планування та підготовки до перевірки, керівнику інспекційної групи необхідно, окрім виконання вимог відповідних нормативно-правових документів, у повідомленнях до банку навести перелік інформації та формат подання такої інформації в електронному вигляді для завантаження (імпорту) з метою опрацювання за допомогою або стандартних засобів обробки масивів інформації (наприклад, Microsoft Excel, Access), або відповідного прикладного програмного забезпечення.

На цьому етапі пропонується зобов'язати установу банку здійснити підготовку до надання таких даних (масивів даних) в електронному вигляді (рис. 5):

- книга відкритих рахунків;
- книга закритих рахунків;
- дані за операціями клієнтів за період, що перевірятиметься;
- дані за касовими операціями установи банку;
- дані карток клієнтів;
- дані анкет клієнтів;
- дані реєстру операцій, які підлягають фінансовому моніторингу за період, що перевірятиметься;
- оборотно-сальдові відомості в розрізі аналітичних рахунків за період, що перевірятиметься.





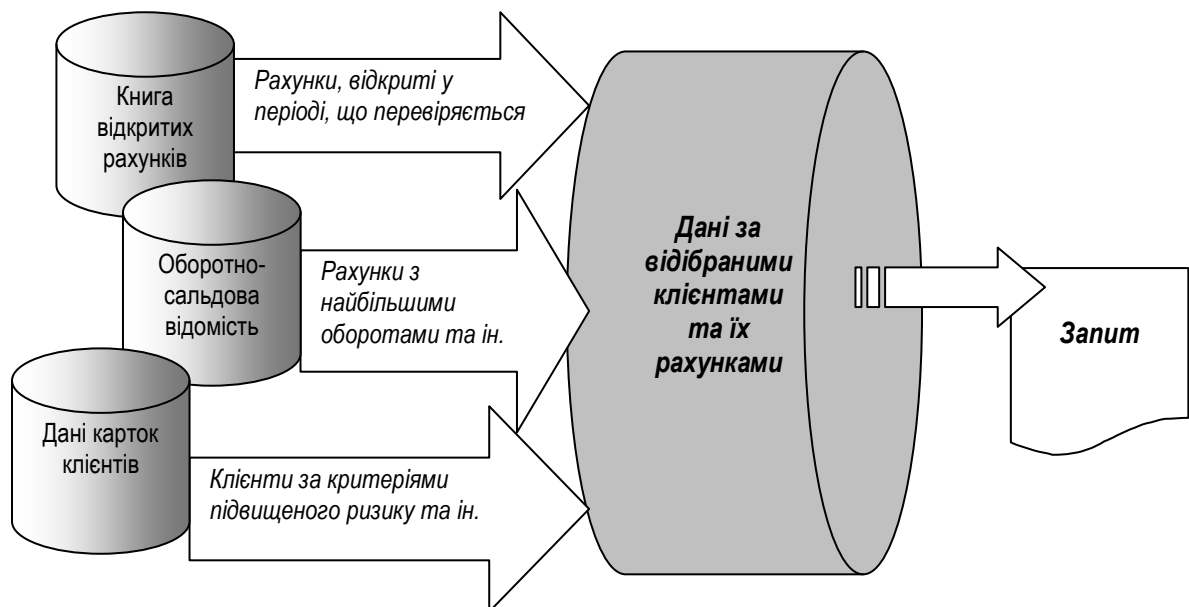
**Рис. 5. Електронні дані (масиви даних) для проведення перевірки**

### *Початок перевірки*

На початку перевірки інспектору (керівнику інспекційної групи) необхідно скласти запит на отримання таких даних в електронному вигляді:

- книга відкритих рахунків;
- дані карток клієнтів;
- оборотно-сальдові відомості.

На основі отриманих даних здійснюється відбір клієнтів, які перевірятимуться щодо якості проведення ідентифікації та виявлення операцій, які підлягають фінансовому моніторингу (рис. 6).



**Рис. 6. Відбір клієнтів та операцій для перевірки**

Слід зазначити, що у разі незначних обсягів операцій, які здійснюються в установі банку, що перевіряється, відбір клієнтів та рахунків можна не здійснювати – опрацювання буде здійснюватися за даними усіх клієнтів установи та усіх операцій.

У результаті відбору операцій інспектор складає запит на отримання такої інформації:

- дані за операціями та рахунками відібраних клієнтів (або усіх клієнтів), відкритими у банківській установі, яка перевіряється, за період, що підлягає перевірці (операції клієнтів банку);
- дані анкет відібраних клієнтів (або усіх клієнтів).

Для подальшого аналізу роботи установи банку необхідно буде скласти запити на отримання:

- електронних даних за касовими операціями установи банку;
- книги закритих рахунків в електронному вигляді;
- електронних даних реєстру операцій.

*Перевірка якості виявлення операцій, які підлягають фінансовому моніторингу*

Перевірка виконання установою вимог щодо виявлення операцій, які підлягають фінансовому моніторингу, є чи не найбільш трудомісткою за обсягами даних, що перевіряються.

Враховуючи те, що банки мають вносити до реєстру всі без виключення операції, які мають ознаки обов'язкового та внутрішнього фінансового моніторингу, надання оцінки якості виявлення таких операцій має базуватися на опрацюванні якомога більшого масиву клієнтських операцій.

Для прикладу пропонуємо розглянути декілька ефективних способів виявлення найбільш типових операцій, що мають ознаки обов'язкового фінансового моніторингу, за умови надання банком даних в електронному вигляді визначеного формату.

Виявлення операцій, що підлягають обов'язковому фінансовому моніторингу за ознакою «зарахування на рахунок коштів у готівковій формі з

їх подальшим переказом того самого або наступного операційного дня іншій особі» (код ознаки: 1031, 1032)

Алгоритм виявлення таких операцій є чи не найпростішим та найефективнішим з точки зору автоматичного виявлення.

Автоматичній обробці підлягають дані за операціями за рахунками клієнтів (операціями клієнтів банку) та дані за операціями, які були внесені до реєстру (реєстр операцій) (рис. 7.). Як видно зі схеми, виявлення операцій за даною ознакою здійснюється в три етапи.

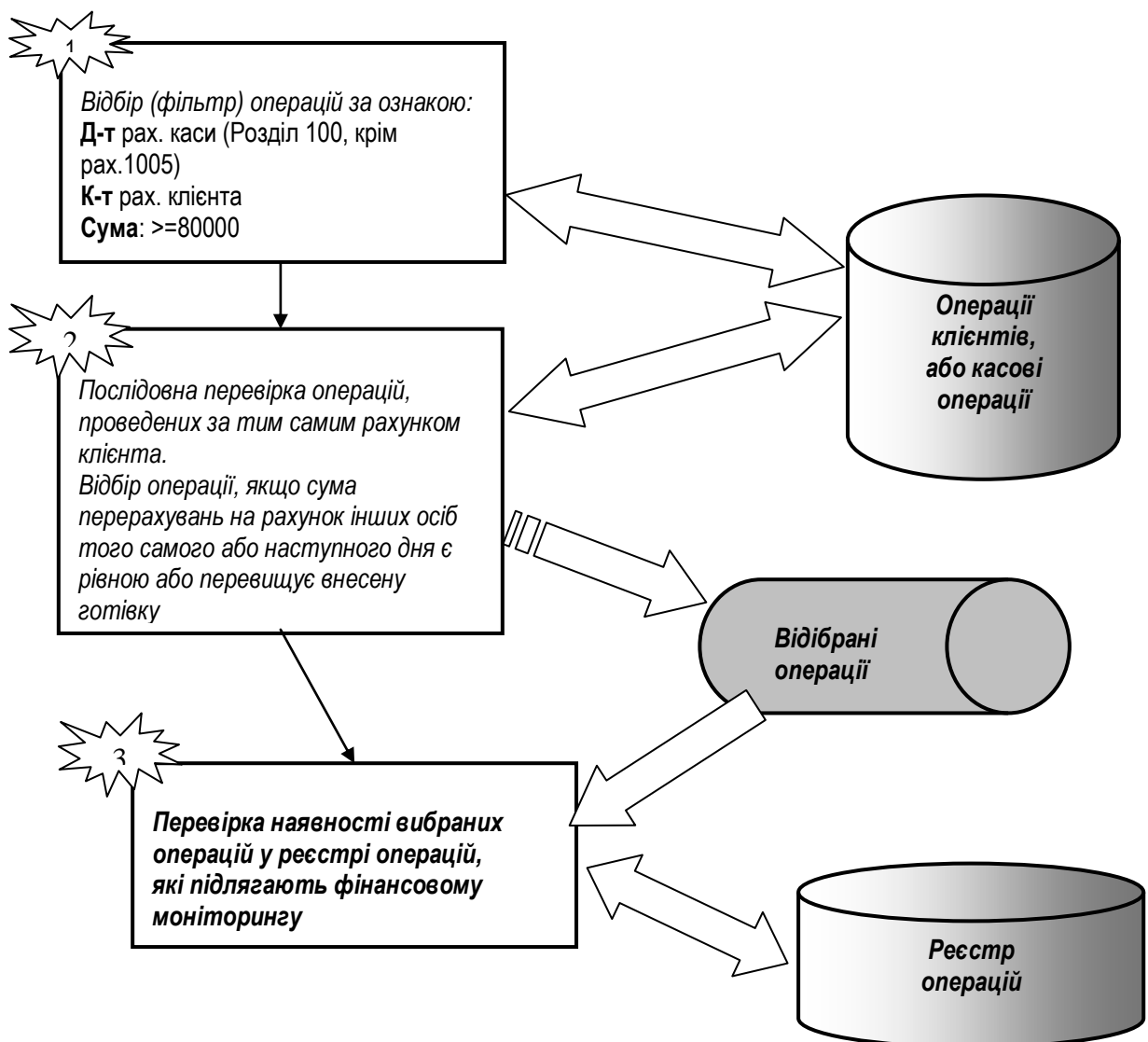


Рис. 7. Виявлення операцій за ознакою 1031, 1032

На **першому етапі** здійснюється відбір (фільтр) операцій внесення готівки на рахунок клієнтів шляхом перегляду даних за операціями за

рахунками клієнтів, визначених попередньо (вибіркова перевірка), або перегляду даних за операціями за рахунками каси (суцільна перевірка), суми за якими є рівними або перевищують 80,0 тис. грн.

На **другому етапі** здійснюється відбір (фільтр) операцій з подальшого перерахування того самого або наступного робочого дня іншим особам, причому загальна сума таких перерахувань має бути рівною або перевищувати суму попередньо внесеної готівки. Даний етап можливий за наявності відповідно складного програмного коду (програми обробки даних або відповідного макросу). При відсутності програмного коду подальша обробка відібраних на першому етапі операцій здійснюється в «ручному» режимі.

На **третьому етапі** здійснюється безпосередньо виявлення порушень, викликаних невнесенням інформації про відібрані операції шляхом перевірки її наявності у реєстрі операцій, які підлягають фінансовому моніторингу.

## ВИСНОВКИ

Автоматизоване виконання зазначених етапів дозволить досить швидко і чітко визначити повноту надання інформації за операціями, яким притаманна зазначена ознака обов'язкового фінансового моніторингу.

Виявлення операцій, що підлягають обов'язковому фінансовому моніторингу за ознакою «зарахування грошових коштів на рахунок чи списання грошових коштів з рахунку, період діяльності якої не перевищує трьох місяців з дня її реєстрації» (код ознаки: 4010, 4030)

Алгоритм виявлення таких операцій є нескладним та ефективним з точки зору автоматичного виявлення операцій за даною ознакою.

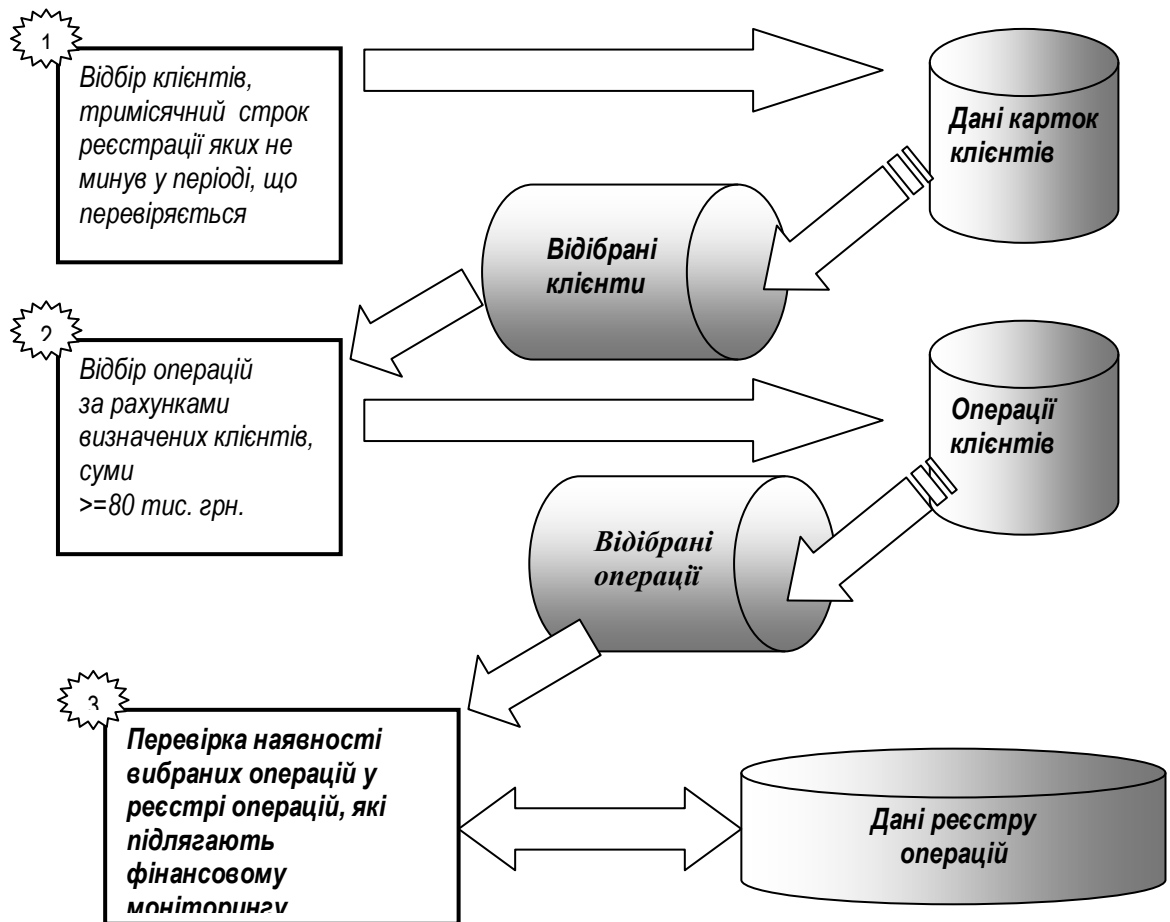
Автоматичній обробці підлягають дані карток клієнтів, дані за операціями за рахунками клієнтів (*операцій клієнтів банку*) та дані за операціями, які були внесені до реєстру (*реєстр операцій*) (рис. 8).

Виявлення операцій за даною ознакою здійснюється у три етапи.

На **першому етапі** здійснюється відбір (фільтр) клієнтів, тримісячний строк реєстрації яких (згідно з даними свідоцтва про реєстрацію) не минув протягом періоду, який перевіряється. Іншими словами, клієнт має бути відібраний у проміжний масив даних, якщо він зареєстрований не раніше ніж за два місяці до початку перевірки.

На наступному, **другому етапі**, здійснюється відбір (фільтр) операцій за рахунками відібраних на першому етапі клієнтів, суми яких є рівними або перевищують 80,0 тис. грн.

На останньому, **третьому етапі**, здійснюється безпосередньо виявлення порушень, викликаних невнесенням інформації про відібрані операції, шляхом перевірки її наявності у реєстрі операцій, які підлягають фінансовому моніторингу.



**Рис. 8. Виявлення операцій за ознакою 4010, 4030**

Отже, автоматизоване виконання зазначених етапів дозволить досить швидко і чітко визначити повноту надання інформації за операціями, яким притаманна зазначена ознака обов'язкового фінансового моніторингу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про запобігання та протидію легалізації (відмивання) доходів, одержаних злочинним шляхом [Електронний ресурс] : закон України за станом на 1 січня 2006 р. – Режим доступу : <http://zakon.rada.gov.ua>.
2. Методичні вказівки з інспектування банків «Система оцінки ризиків», затверджені постановою Правління Національного банку України від 15.03.2004 № 104 за станом на 15 березня 2004. – Режим доступу : [http://www.bank.gov.ua/Bank\\_supervision/Risks/104.pdf](http://www.bank.gov.ua/Bank_supervision/Risks/104.pdf).
3. Моделювання оцінки ризиків використання послуг банків або їх структурних одиниць з метою легалізації кримінальних доходів або фінансування тероризму : монографія / [С. О. Дмитров, О. В. Меренкова, Л. Г. Левченко, Т. А. Медвідь] ; під заг. редакцією О. М. Бережного. – Суми : ДВНЗ “УАБС НБУ”, 2008. – 75 с.
4. Про затвердження деяких форм обліку та подання інформації, пов'язаної із здійсненням фінансового моніторингу, та Інструкції щодо їх заповнення [Електронний ресурс] : наказ Державного департаменту фінансового моніторингу Міністерства фінансів України від 13 травня 2003 року № 48 / Держкомітет фінансового моніторингу України. — Режим доступу : [http://sdfm.gov.ua/content/file/Site\\_docs/2009/030809/Nakaz48.html](http://sdfm.gov.ua/content/file/Site_docs/2009/030809/Nakaz48.html).
5. Про затвердження Положення про здійснення банками фінансового моніторингу [Електронний ресурс] : постанова Правління Національного банку України від 14.05.2003 №189 за станом на 20 листопада 2007 р. – Режим доступу : <http://zakon.rada.gov.ua>.
6. Про окремі питання організації та проведення в банках фінансового моніторингу [Електронний ресурс] : лист Національного банку України Територіальним управлінням Національного банку України, банкам України, Асоціації українських банків, Українському кредитно-банківському союзу від 04 вересня 2009 р. № 48-012/1939-16688. – Режим доступу : <http://www.bank.gov.ua>.

7. Типології легалізації (відмивання) доходів, одержаних в результаті здійснення фінансових операцій з неліквідними цінними паперами: затверджені наказом Держфінмоніторингу України від 20.12.2007 № 230 [Електронний ресурс] / Держкомітет фінансового моніторингу України. – Режим доступу : [ttp://www.sdfm.gov.ua/content/File/Site\\_docs/25.12.2007/Typologies.htm](http://www.sdfm.gov.ua/content/File/Site_docs/25.12.2007/Typologies.htm).
8. Типології легалізації злочинних коштів в Україні в 2004–2005 роках: затверджені наказом Держфінмоніторингу України від 07.12.2005 № 249 [Електронний ресурс] / Держкомітет фінансового моніторингу України. — Режим доступу: [http://www.sdfm.gov.ua/content/File/Site\\_docs/24.05.06/18-05-2006.pdf](http://www.sdfm.gov.ua/content/File/Site_docs/24.05.06/18-05-2006.pdf).
9. Моделювання оцінки операційного ризику комерційного банку / під загальною редакцією С.О. Дмитрова // Суми : ДВНЗ «УАБС НБУ». – 2010. – 310 с.
10. Моделювання оцінки ризиків використання послуг банків або їх структурних одиниць з метою легалізації кримінальних доходів або фінансування тероризму : монографія / [С. О. Дмитров, О. В. Меренкова, Л. Г. Левченко, Т. А. Медвідь] ; під заг. редакцією О. М. Бережного. – Суми : ДВНЗ “УАБС НБУ”, 2008. – 75 с.
11. Guidance on a Risk Based Approach for Managing Money Laundering Risks [Electronic resource] : Wolfsberg Group – Режим доступу : <http://www.wolfsberg-principles.com>.
12. Guidance on the Risk-Based Approach to combating money laundering and terrorist financing [Electronic resource] : Financial Action Task Force (FATF) – Режим доступу : <http://www.fatf-gafi.org>.
13. Money Laundering Red Flags [Electronic resource]: Federal Financial Institutions Examination Council's (FFIEC) – Режим доступу : <http://www.ffiec.gov>.
14. Risk-Based Approach: Guidance for Money Service Businesses [Electronic resource] : Financial Action Task Force (FATF) – Режим доступу : <http://www.fatf-gafi.org>.