

<https://doi.org/10.1109/ICPECA56706.2023.10075760>

14. Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, 404, 132306. <https://doi.org/10.1016/j.physd.2019.132306>
15. He, K., Zhang, X., Ren, S., & Sun, J. (2015, December). *Deep Residual Learning for Image Recognition*. arXiv. <http://arxiv.org/abs/1512.03385>
16. Miki, T. (1999). *Analog Implementation of Neo-Fuzzy Neuron and Its On-board Learning*.
17. Uchino, E., & Yamakawa, T. (1997). Soft Computing Based Signal Prediction, Restoration, and Filtering. In D. Ruan (Ed.), *Intelligent Hybrid Systems* (pp. 331–351). Springer US. https://doi.org/10.1007/978-1-4615-6191-0_14
18. Yamakawa, J., Uchino, E., Miki, J., & Kusanagi, H. (1992, July). A neo-fuzzy neuron and its application to system identification and prediction of the system behavior. *Proceedings of the 2nd International Conference on Fuzzy Logic & Neural Networks*.
19. Zhang, Y., Wang, G., Zhou, T., Huang, X., Lam, S., Sheng, J., Choi, K. S., Cai, J., & Ding, W. (2024). Takagi-Sugeno-Kang fuzzy system fusion: A survey at hierarchical, wide and stacked levels. *Information Fusion*, 101, 101977. <https://doi.org/10.1016/j.inffus.2023.101977>
20. Bodyanskiy, Ye. V., & Kulishova, N. E. (2014). Extended neo-fuzzy neuron in the task of images filtering. *Radio Electronics, Computer Science, Control*, 0(1). <https://doi.org/10.15588/1607-3274-2014-1-16>
21. Bodyanskiy, Y., Popov, S., & Titov, M. (2010). Robust Learning Algorithm for Networks of Neuro-Fuzzy Units. In T. Sobh (Ed.), *Innovations and Advances in Computer Sciences and Engineering* (pp. 343–346). Springer Netherlands. https://doi.org/10.1007/978-90-481-3658-2_59
22. Haykin, S. S. (1999). *Neural networks: A comprehensive foundation* (2nd ed). Prentice Hall.
23. Goodwin, G. C., Ramadge, P. J., & Caines, P. E. (1981). Discrete Time Stochastic Adaptive Control. *SIAM J. Control Optim.*, 19(6), 829–853. <https://doi.org/10.1137/0319052>
24. Widrow, B., & Hoff, M. E. (1960). *Adaptive Switching Circuits*. IRE WESCON Convention Record.

Надійшла до редколегії 07.08.2024 р.

Бодяньський Євген Володимирович, доктор технічних наук, професор, професор кафедри ІІІ ХНУРЕ, м. Харків, Україна, e-mail: yevgeniy.bodyanskiy@nure.ua; ORCID: 0000-0001-5418-2143
Чала Ольга Сергіївна, старший викладач кафедри ІІІ ХНУРЕ, м. Харків, Україна, e-mail: olha.chala@nure.ua; ORCID: 0000-0002-7603-1247

УДК 004.891.3

DOI: 10.30837/0135-1710.2024.181.050

I.A. МАЛЬКОВА, В.С. МАКЕШКО

МЕТОД АВТОМАТИЗОВАНОЇ ПОБУДОВИ БАЗИ ЗНАНЬ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОЦЕСНОГО УПРАВЛІННЯ

Запропоновано метод автоматизованої побудови та поповнення бази знань системи процесного управління. Для вдосконалення методу використано модифіковані предикатні моделі. Модифікацію предикатних моделей здійснено за рахунок використання традиційного повного циклу розробки моделі гнучкого багатоваріантного процесу. Наведено опис особливостей системи автоматизованої побудови бази знань при вирішенні задач прийняття рішень на ІТ-підприємстві. Проведено апробацію запропонованих рішень у ході автоматизованої побудови бази знань у ІТ-проєкті.

1. Вступ

У сучасному світі будь-які знання є стратегічним ресурсом, тому розробка методів і технологій управління знаннями, зокрема, базами знань (БЗ), залишається актуальною.

Зазвичай під управлінням знаннями, у першу чергу, розуміють систематичний збір та використання корпоративних знань з метою максимальної ефективності їх застосування.

Найважливішими роботами, які виконують у процесі управління знаннями, є їх видобуток, структурування та формалізація.

Складність і трудомісткість виконання цих робіт обумовлена складністю процесу створення систем, заснованих на знаннях [1]-[3].

Увагу розробників програмного забезпечення (ПЗ) вже давно привертає питання, як віднайти ефективний спосіб доставки до користувача інформації, якою традиційно володіють експерти-спеціалісти. Однією зі складових вирішення цього питання є створення БЗ для вирішення прикладних задач.

У наш час БЗ поступово стають невід'ємною частиною різних систем і технологій. Проте більшість розробок у даному напрямку присвячена створенню електронних архівів і каталогів. Зазвичай при такому підході визначення критеріїв пошуку необхідної інформації лягає на користувача, який може не мати жодного уявлення про предметну область, а отже, і не в змозі самостійно здійснювати цілеспрямований пошук.

Розробка нових методів і підходів до створення інформаційних систем (ІС) і програмних компонентів для них залишається перспективною галуззю наукових досліджень. Основним елементом ІС є БЗ, яка забезпечує введення, зберігання та обробку знань, закономірностей і взаємозв'язків будь-якої предметної області за запитами користувачів.

Процес формування БЗ під час розробки ІС традиційно вважають достатньо складним і пов'язують з задачами моделювання предметної області, ідентифікації (отримання), концептуалізації (структурування) і формалізації (представлення) знань з подальшою програмною реалізацією [4]. Ще більшої складності ці задачі набувають за необхідністю забезпечення віддаленого і розподіленого доступу, узгодження висновків експертів (фахівців), а також використання інформації, представлення її в різних формах, у тому числі у формі концептуальних моделей.

Зазвичай концептуальні моделі предметних областей формують у процесі розробки ПЗ, у процесі цілеспрямованого моделювання предметної області, у процесі побудови онтології або бізнес-моделі тощо. Під час створення моделей використовують різні нотації, мови і програмні засоби (системи концептуального, когнітивного, онтологічного моделювання, CASE-засоби). Більшість з них забезпечують створення концептуальних моделей і генерацію документації, проте передбачають обмежені можливості з перетворення побудованих моделей у структурі мови програмування (часткове/скелетне перетворення). А це ускладнює практичне використання побудованих моделей для формування БЗ при розробці ІС.

У зв'язку з цим проблема розробки ПЗ сервіс-орієнтованої ІС з можливостями розширення є актуальною. При цьому під можливістю розширення розуміють можливість створення користувачем нових програмних компонентів, які реалізують функції аналізу концептуальних моделей і генерації коду БЗ для інших форматів.

2. Аналіз літературних даних і постановка проблеми дослідження

Традиційні підходи до побудови БЗ не є придатними для систем, які функціонують у реальному часі, через те, що формалізація експертного досвіду у вигляді причинно-наслідкових залежностей вимагає значних витрат часу та участі кваліфікованих фахівців. Останнім часом набули поширення методи, підходи і технології автоматизованої побудови БЗ [5]-[7]. Загальний метод автоматизованої побудови БЗ на основі аналізу журналів подій запропонований в [7]. Згадувані у [5]-[7] методи призначені для виявлення знань у великих базах даних (БД), доступних у мережі Інтернет. Проте підходи, на яких базуються зазначені методи, мають істотний недолік, що звужує сферу їх використання: вони орієнтуються на статичний опис залежностей у предметній області.

У той самий час, при використанні БЗ для підтримки прийняття рішень у системі процесного управління необхідно постійно, синхронно з ходом виконання відповідних бізнес-процесів (БП), оновлювати бази фактів про предметну область. Необхідно також представляти багатоваріантність таких фактів і зв'язків між ними. Для вирішення цієї задачі використовують представлення знань на основі знаходження ймовірностей за допомогою марковських логічних мереж. В [8] запропонований метод автоматизованої побудови БЗ, що

використовує представлення знань на основі марковських логічних мереж. Методи імовірнісного виведення на базі марковських логічних мереж надано в [9]. Основна ідея таких мереж полягає у використанні шаблонів логічних залежностей для побудови набору зважених предикатів, що відображають знання про предметну область [10].

Проте під час побудови такого представлення для систем процесного управління необхідно враховувати особливості логів подій як вихідних даних для виявлення знань [11]. Урахування цих особливостей дозволяє спростити побудову та розширити БЗ.

Таким чином, розробка методу автоматизованої побудови і поповнення БЗ ІС, який забезпечував би безперервну актуалізацію знань на основі аналізу логів, є актуальною.

3. Мета і задачі дослідження

Основною метою дослідження є вдосконалення методу автоматизованої побудови та поповнення БЗ ІС управління ІТ-проєктами (зокрема, проєктами розробки ІС), що дозволить ефективно використовувати всі можливості БЗ під час вирішення основних задач управління проєктами.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- вдосконалення методу автоматизованої побудови і поповнення БЗ ІС на основі модифікованих предикатних моделей;
- дослідження особливостей визначення вихідних транзакцій під час планування та управління ІТ-проєктом;
- апробація запропонованих рішень у ході автоматизованої побудови БЗ у ІС управління ІТ-проєктом.

4. Дослідження моделей та методів автоматизованої побудови бази знань інформаційної системи процесного управління

На даний час для створення БЗ існує велика кількість підходів, які, у свою чергу, можна поділити на автоматизовані та неавтоматизовані. Слід зазначити, що під час застосування обох підходів людина (експерт) тією чи іншою мірою буде брати участь у формуванні та супроводі БЗ.

Автоматизоване формування та супровід БЗ ІС передбачає такі етапи.

Етап 1. Підготовка вхідного набору даних. На цьому етапі відбувається перенесення даних з БД (за умови наявності), створення набору даних з різних джерел інформації (фахівці, література, нормативні документи, Internet тощо). Автоматизація відбувається за рахунок застосування пошуку та сканування текстів, діалогу експерта та спеціальної програми, обробки протоколів інтерв'ю тощо.

Етап 2. Попередня обробка даних. Дані можуть містити помилки, аномальні значення або бути непридатними для автоматизованої обробки. На даному етапі також відбувається створення словників з предметної області та правил. Автоматизація цього етапу передбачає використання засобів оперативного аналізу OLAP, інтелектуального аналізу Data Mining і статистичного аналізу.

Етап 3. Нормалізація даних. Цей етап необхідний для методів, які працюють з тільки з даними визначеного виду (наприклад, нейронні мережі працюють тільки з числовими даними). На цьому етапі можуть бути використані як програмні, так і аналітичні методи перетворення типів даних.

Етап 4. Застосування методів формування знань. Ці методи обираються залежно від складності та архітектури ІС. Автоматизація формування знань відбувається за допомогою програм навчання, побудованих на нейронних мережах, генетичних алгоритмах тощо, за наявності репрезентативної вибірки прикладів прийняття рішень у визначеній предметній області.

Етап 5. Подальша обробка даних. Цей етап передбачає безпосередню обробку даних відповідно до запитів, інтерпретацію результатів і застосування отриманих знань засобами самої ІС.

Етап 6. Підтримка даних у актуальному стані. Існує багато предметних областей, у яких дані швидко застарівають, тому їх необхідно періодично оновлювати, підтримуючи у актуальному стані. Автоматизація на цьому етапі передбачає застосування таких самих засобів, як і на етапах 1 і 2.

Метод автоматичної побудови і використання БЗ використовує логіко-імовірнісне представлення знань на основі марковських логічних мереж. Зазначене представлення враховує такі особливості логів подій ІС:

- лог відображає властивості артефактів БП, тобто об'єктів, з якими цей процес взаємодіє;
- з кожною подією логів пов'язана часова мітка (timestamp), а також набір значень атрибутів;
- атрибути подій логів відповідають атрибутам артефактів БП;
- сукупність окремих атрибутів та властивостей артефактів описує контекст виконання дій БП.

Представлення знань має вигляд:

$$KB_{L,P} = (Af, \{ (f_i(Af), w_i) \}, \{ (r_j(F), w_j) \}, P(A = \{ \alpha_k^\tau \}) | C), \quad (1)$$

де Af – множина артефактів; F – множина логічних фактів f_i ; w_i – вага логічного факту f_i ; r_j – логічне правило, яке оперує логічними фактами; w_j – вага логічного правила r_j ; α_k^τ – значення властивості артефакту в момент τ запису події до логу БП; $P(A = \{ \alpha_k^\tau \})$ – ймовірність виконання правил за умови відомих на поточний момент значень властивостей артефактів; C – апріорно відомі каузальні залежності, які виступають як обмеження предметної області.

Зазначене представлення знань враховує послідовність розгортання БП у часі. Статичний аспект задається фактами і правилами з аргументами з логу подій, а динамічний – у вигляді поточного розподілу ймовірностей виконання правил відповідно внесених до логу подій.

Це дозволяє вирішувати задачу прогнозування найімовірнішої поведінки БП в тому випадку, якщо поточний стан раптово змінився внаслідок того, що виконавці змінили задану у моделі послідовність дій.

Імовірнісний розподіл можливих реалізацій БП враховує зважену суму правил і має традиційний для марковських мереж вигляд:

$$P(A = \alpha) = \frac{1}{Z} \exp(\sum_j w_j r_j(\{f_i\})). \quad (2)$$

Метод складається з таких етапів.

Етап 1. Побудова шаблону представлення даних і знань шляхом визначення класів артефактів (властивостей артефактів), типових фактів f_i і правил r_j відповідно до структури логу. Шаблони логічних фактів відображають атрибути подій. Шаблони правил відповідають послідовності подій і відображають зв'язок між контекстом і діями (станом дій) БП. Результат цього етапу: підмножина предикатів, які встановлюють логічні факти; підмножина предикатів, які встановлюють логічні правила. Кількість таких предикатів обмежена через те, що всі події логів мають однакову структуру. Підхід передбачає використання мінімального набору з трьох предикатів: предикат, який встановлює значення властивості артефакту; предикат, що задає набір властивостей для події; предикат, який визначає правила переходу між подіями.

Етап 2. Побудова/доповнення опису контексту предметної області. На даному етапі визначається приналежність атрибутів подій логів до класів артефактів та їхніх властивостей. Відзначимо, що під час побудови БЗ засобами реляційної системи управління базою даних (СУБД) перелік властивостей артефактів і унікальних значень цих властивостей формується з логу процесу у табличній формі шляхом SQL-запитів.

Етап 3. Побудова логічних фактів шляхом підстановки до предикатів-шаблонів аргументів у вигляді значень атрибутів подій. Такі аргументи характеризують поточний стан БП. Результатом етапу є БЗ у вигляді набору логічних фактів, які відображають нормальну поведінку БП.

Етап 4. Розрахунок ваг логічних фактів. Ваги розраховуються таким чином, щоб їхні значення відповідали ймовірності виконання правил на множині подій логів БП. Традиційно такий розрахунок у марковських логічних мережах виконують методами на основі градієнтного спуску. У результаті виконання даного етапу правила адаптуються з урахуванням додаткової інформації про нові події логів.

Етап 5. Побудова правил, аргументами яких є отримані на етапі 3 логічні факти. Правила формуються на основі предиката-шаблону підстановкою значень аргументів.

Етап 6. Розрахунок ймовірностей для правил на основі логічних фактів, аргументами яких є атрибути подій логів БП. У результаті виконання даного етапу може бути використана інформація про найімовірніші дії БП з прив'язкою до контексту. Додатковим результатом етапу може бути інформація про аномальну поведінку БП.

Етапи 2-6 повторюються протягом виконання БП і появи нових подій в логах.

У ході дослідження було запропоновано використовувати предикатні моделі, що описують артефакти ІТ-проекту.

Для вирішення задачі побудови моделі гнучкого процесу відомі традиційні методи інтелектуального аналізу процесів вимагають доопрацювання. Тому вдосконалений метод повинен забезпечувати врахування додаткового аспекту процесу – ієрархії його дій, а також можливість ефективного відсікання непотрібних дій під час конфігурування моделі процесу. Як математичний апарат цього вдосконалення доцільно використати алгебру кінцевих предикатів [10]. Отримана з використанням цього апарату вдосконалена предикатна моделі процесу дозволяє:

- забезпечити додаткові можливості за рахунок появи моделей гнучких процесів;
- полегшити загальне розуміння моделі за рахунок ієрархічного представлення моделі гнучкого процесу
- використати як елементи моделі існуючі формалізовані підпроцеси;
- побудувати моделі реально виконаних процесів обробки ресурсів на основі логів методами інтелектуального аналізу процесів.

Аналіз досліджень у галузі побудови моделей гнучких процесів показав, що повна модель такого процесу може бути отримана за допомогою традиційного повного циклу розробки моделі гнучкого багатоваріантного процесу, зокрема злиття моделей декількох «жорстких» процесів, які реалізують ідентичну функціональність різними способами.

5. Вирішення задачі автоматичної побудови бази знань інформаційної системи процесного управління

5.1. Вдосконалення методу автоматизованої побудови та поповнення бази знань інформаційної системи процесного управління на основі модифікованих предикатних моделей

В умовах поточної роботи підприємства неявні знання становлять найбільший інтерес. Проте саме ці знання найважче отримати.

У кожній організації, незалежно від сфери її діяльності, здійснюється безперервний процес руху знань, тобто здійснюється певний процес виробництва, узагальнення та

поширення знань. Моделювання процесів можна виконувати із застосуванням різних підходів і інструментальних засобів, залежно від вимог до моделі у кожному конкретному випадку. Зазвичай таке моделювання починається з ідентифікації процесу. Метою цієї ідентифікації стосовно предметної області є виявлення в діяльності організації процесів управління знаннями, їх описи та використання цих описів для управління процесами і їх поліпшення, а також встановлення єдиних вимог до правил і способу опису процесів організації.

Під час ідентифікації процесу на верхньому рівні обов'язково повинні бути визначені: назва процесу; входи процесу; виходи процесу; виконавець; структурні підрозділи; окремі працівники, зовнішні виконавці; керуючі входи процесу; нормативні, організаційно розпорядчі та методичні документи, що визначають вимоги до процесу.

Після ідентифікації відбувається декомпозиція процесу, яка передбачає представлення основного процесу у вигляді підпроцесів, тобто основних робіт, що слідують один за одним у певній послідовності протягом реалізації основного процесу. Вихід кожного підпроцесу одночасно є входом для наступного.

Усі знання ІТ-підприємства можна представити як розподілену БЗ, що проявляє себе у середовищі БП у формі об'єктів знань. До таких об'єктів належать агенти (працівники і групи працівників) та артефакти (документи, книги, комп'ютерні системи тощо).

Знання, які містять агенти, є суб'єктивними, тобто прихованими, у той час як знання, які містяться в артефактах, є об'єктивними, представленими у формі явних, закодованих лінгвістичних виразів [12].

Об'єкти знань можуть розглядатися як інтерфейс між результатами обробки знань і середовищем виконання БП, в якому рішення і дії здійснюються співробітниками організації під час виконання виробничих завдань.

Таким чином, метод, описаний у розділі 4, доцільно удосконалити. Це удосконалення ґрунтується на припущенні, за яким таблиці реляційної або об'єктно-реляційної БД ІС управління ІТ-проектами, які по суті є предикатами, можуть містити такі об'єкти знань:

- знання, які описують властивості артефактів;
- знання, які описують властивості подій;
- знання, які описують правила переходів між подіями.

Шаблон представлення даних на етапі 1 розглянутого методу доцільно розширити за рахунок додавання предикату, який задає множину припустимих для подальшого аналізу артефактів. Тобто для розглянутої предметної області необхідні артефакти, які описують методологію роботи над ІТ-проектами (наприклад, Scrum/Agile або планове виконання проєктів). У випадку зі Scrum/Agile необхідно описувати такі артефакти як план спринту, беклог, щоденні скрами, ретроспектива тощо. Необхідно також зробити висновки щодо цих артефактів на предмет доцільності їх використання під час аналізу. Наприклад, щоденні скрами можна вилучити зі списку артефактів для аналізу через те, що вони ніяк не впливають на успішність виконання проєкту.

На етапі 2 необхідно узгодити логи подій, взяті з системи управління ІТ-проєктом, з класами артефактів та їх властивостей. Тобто одного тільки опису недостатньо, потрібна відповідність вмісту логів визначеним артефактам.

Таким чином, для кожного виконаного проєкту вміст логів запропоновано записувати до БД, а під час побудови плану поточного проєкту виконувати порівняння накопичених таким чином знань з описами відповідного проєкту. Наприклад, можна сформулювати опис нульового спринту на основі логів нульових спринтів попередніх ідентичних проєктів. Застосування цих пропозицій дозволить оцінити успішність початку та просування поточного ІТ-проєкту.

Ґрунтуючись на використанні БЗ, доцільно виділити предикати, додані до запропонованих у розглянутому методі:

- предикати, які описують усі можливі артефакти;
- предикати, які описують клас артефактів проекту;
- предикати, які описують артефакти, відібрані для формування БЗ.

Таким чином, модернізований метод автоматизованої побудови та поповнення БЗ системи процесного управління містить такі етапи:

Етап 1. Побудова шаблону представлення даних і знань шляхом визначення класів артефактів (властивостей артефактів), типових фактів f_i і правил r_j відповідно до структури логу.

Етап 2. Доповнення опису контексту предметної області.

Етап 3. Побудова логічних фактів шляхом підстановки до предикатів-шаблонів аргументів у вигляді значень атрибутів подій.

Етап 4. Побудова правил, аргументами яких є отримані на етапі 3 логічні факти.

Етап 5. Розрахунок ймовірностей для правил на основі логічних фактів, аргументами яких є атрибути подій логів БП.

Розрахунок ймовірностей для правил визначається у такий спосіб.

Правило r_j визначається парою подій

$$r_j = (e_{ij}, e_{ij+1}), \quad (3)$$

де r_j – правило пар подій.

З урахуванням особливостей вхідних даних (журналів подій) запропоновано розраховувати ваги правил таким чином:

$$k = |\{trace_i: (e_{ij}, e_{ij+1}) \in trace_i\}|, \quad (4)$$

де k – кількість трас; $trace_i$ – траса; e_{ij} – подія.

Ймовірність для правила $j(r_j)$ розраховується за формулою:

$$j(r_j) = \frac{k}{K}, \quad (5)$$

де K – загальна кількість трас.

Таким чином, модернізація методу автоматизованої побудови та поповнення БЗ ІС процесного управління відбувається шляхом додавання нових предикатів та узгодження логів подій з класами артефактів та їх властивостей.

Подальшим розвитком модернізованого методу може бути аналіз успішності виконання прогнозів за допомогою БЗ і корегування вихідного набору предикатів на основі результатів цього аналізу.

5.2. Експериментальна перевірка виявлення вторгнень у комп'ютерну мережу

Основною метою експерименту була перевірка можливості застосування запропонованого методу під час вирішення задач підтримки прийняття рішень за умови виявлення вторгнень для процесів у комп'ютерних системах.

Під час проведення експерименту був використаний лог комп'ютерної системи CIDDS-002. У CIDDS-002 містяться логи з тижневими послідовностями подій. Кожна подія логів характеризується множиною атрибутів (Source IP Address, Source Port, Destination IP Address, Destination Port, Transport Protocol, Duration of the flow, Duration of the flow, Number of transmitted bytes тощо).

Логічні факти представлені у вигляді кон'юнкції значень атрибутів, наприклад, для логу з двох рядків факт f_1 має вигляд:

$$f_1(\text{SourceIPAddress} = \langle 192.168.220.51 \rangle \wedge \text{SourcePort} = \langle 51144 \rangle \wedge \text{DestinationIPAddress} = \langle 192.168.100.20 \rangle \wedge \dots) = \text{true}. \quad (6)$$

Логічні факти можуть бути узагальнені. Узагальнення логічних фактів полягає у виборі такої кількості атрибутів, яка б дозволила досягти заданої спільності факту без його спрощення. Іншими словами, під час побудови шаблону логічних фактів слід прибирати несуттєві атрибути. Зокрема, у наведеному прикладі не використано timestamp.

Логічні правила є супутніми для переходу між подіями, наприклад для фактів f_1 і f_2 з наведеного фрагмента логу, загальне правило переходу має вигляд $r_1(f_1 \wedge f_2)$.

З зазначеного логу відфільтровано послідовності подій, відповідні процесам нормального функціонування, а також відповідні процесам зовнішніх вторгнень. Ці послідовності представлено у вигляді окремих трас. Траса процесу нормального функціонування містить 1011427 подій, а траса процесу зовнішніх вторгнень – 37148 подій.

Ваги логічних фактів розраховуються окремо для обох трас. Це дозволяє реалізувати порівняння ймовірностей виконання правил для нормального функціонування і для зовнішніх вторгнень.

Спочатку виявляємо аномальну поведінку процесу. Проте аномальна поведінка не завжди свідчить про помилки або зовнішні вторгнення. Можливий варіант, що опис такої поведінки процесу просто не міститься у БЗ. Тому потім виконується порівняння з шаблонною поведінкою процесу під час вторгнення. У результаті відсікаються явні процеси вторгнення. Решта описів поведінки процесу, яка залишається після такого відсікання, вимагає подальшого розгляду.

Правило r_j визначається парою подій за формулою (3). Ваги правил розраховуються з використанням формули (4). Для розрахунку ймовірності правила використовується формула (5).

Під час експериментальної перевірки було запропоновано спростити визначення ваг логічних фактів порівняно з традиційними методами внаслідок необхідності роботи в реальному часі. Традиційні методи є вкрай ресурсоемними. Тому ваги фактів і правил визначаються на основі частоти появи значень атрибутів.

Було розраховано ваги переходів між атрибутами подій для кожної пари атрибутів. Ці ваги розраховано для нормальної роботи і для вторгнення. Ваги атрибутів підсумовуються при визначенні ваг логічного факту і правила.

Порівняння значень ваг переходів для випадку нормально процесу та для випадку зовнішнього вторгнення можна здійснити за даними, представленими у табл. 1.

Під час прийняття рішень щодо виявлення зовнішніх вторгнень виконується розрахунок ймовірності для правил переходу між подіями з урахуванням ймовірності логічного факту, який описує стан. Потім виконуються порівняння отриманих ймовірностей для процесу нормальної роботи і процесу вторгнень. За результатами порівняння приймається рішення про виявлені зовнішні вторгнення у ІС.

Таблиця 1

Ваги значень атрибутів для переходів між подіями

Ваги значень атрибутів події	
Нормальна обробка	Вторгнення
1,58192E-05	1,47835E-01
1,68079E-05	1,22064E-03
0,001384183	0,012473248
3,46046E-05	1,257432E-05
0,000375707	0,0
0,059390347	1,056037 E-05

5.3. Опис основних особливостей системи автоматизованої побудови бази знань

Основним ресурсом знань ІТ-підприємств є розроблене ПЗ та документація на нього. Ці знання використовують у всіх основних технологічних процесах ІТ-підприємства:

розробці проєктної документації, розробці програмних продуктів, документуванні, тестуванні, впровадженні і супроводі, а також у процесі побудови плану робіт ІТ-проєкту.

Для виявлення цих знань запропоновано застосовувати систему автоматизованої побудови (САП) БЗ. Головна концепція такої системи полягає у можливості управління знаннями про створені програмні продукти, документацію, процес розробки планів та самого ПЗ, тестування, впровадження, організаційні питання та знаннями з інших ІТ-проєктів та процесів, які надходять у БЗ під час діяльності підприємства.

Складові САП БЗ побудовані з використанням різних компонентів та рішень. Одним з ключових компонентів, який здійснює пошук, організацію знань у вигляді БЗ, забезпечує оперативний відбір і видачу релевантної інформації за запитом та виконує багато інших дій, є система Jira [13].

Таким чином, пошук інформації проводиться не у сховищі документів, а у БЗ, яка дозволяє врахувати семантику інформації, що знаходиться у документації на ПЗ, описах спринтів та беклогів, планах роботи тощо, і тим самим підвищити якість пошуку. Для здійснення такого пошуку запропоновано використовувати комбінацію синтаксичного і семантичного пошуку, тобто пошук повинен виконуватися у примірниках БЗ з урахуванням їх семантичних властивостей і зв'язків.

Основними компонентами САП БЗ є:

- БЗ ІС (використовується для опису інформаційних ресурсів з урахуванням семантики оброблюваної інформації);
- підсистема пошуку (виконує пошук інформації за запитом, навігацію, розташування інформації за рубриками і візуалізацію примірників БЗ ІС, забезпечує обмін знаннями між користувачами системи за допомогою механізмів коментування та тегів);
- підсистема компонування документів (виконує формування необхідних цілих документів у визначених форматах);
- підсистема розмежування доступу (забезпечує розмежування доступу до інформаційних ресурсів);
- підсистема забезпечення цілісності БЗ.

Методика створення САП БЗ та її перевірки складається з таких етапів.

Передпроєктний етап. Відбувається дослідження предметної області та ідентичних розробок, обстеження документації на ПЗ та інших ресурсів знань підприємства, обґрунтування необхідності створення САП БЗ, розробка технічного завдання.

Етап проєктування та розробки. Відбувається:

- проєктування продукційних БЗ ІС;
- розробка автоматизованих продукційних БЗ ІС (визначення понять і відносин між ними);
- наповнення БЗ (модульне структурування документації на ПЗ, створення примірників, заповнення властивостей і формування відносин між екземплярами);
- програмна реалізація (організація автоматизованої обробки і пошуку інформації за запитом, розробка шаблонів документів);
- розробка документації на САП БЗ;
- розробка стандарту, який регламентує правила роботи з САП БЗ, і пропозицій щодо зміни існуючих стандартів ІТ-підприємства у зв'язку з впровадженням системи.

Етап впровадження. Відбувається:

- проведення дослідних випробувань;
- навчання персоналу;
- введення в дію розроблених та модифікованих стандартів;
- введення САП БЗ у промислову експлуатацію.

Етап супроводу. Виконується аналіз функціонування, виявлення проблем, внесення змін до САП БЗ та стандартів організації.

Запропонована методика пройшла перевірку на одному з українських ІТ-підприємств. Під час перевірки було використано документацію на ПЗ, яка містить близько 90 % від усієї інформації підприємства. Обсяг документації становив понад 40000 сторінок тексту.

Технологічною платформою САП БЗ є сполучення продуктів Confluence та Jira. Систему реалізовано у вигляді Інтернет-порталу. БЗ документації на ПЗ реалізовано на основі бібліотеки документів Jira.

Для кожного концепту БЗ документації на ПЗ створений тип контенту (Content Type) зі своїм набором стовпців (атрибутів). Під час перевірки у БЗ документації на ПЗ було виділено близько 100 концептів і 7 типів відносин, створено близько 52000 примірників. Обсяг реструктурованої документації на ПЗ становить приблизно 60 % від похідного (понад 23000 сторінок текстової інформації).

В основі БЗ функціональних можливостей лежить карта БП моделі eТОМ (enhanced Telecom Operations Map) [14]. БЗ функціональних можливостей реалізовано на основі набору термінів керованих метаданих Jira. У БЗ функціональних можливостей створено близько 600 екземплярів.

БЗ продукції реалізовано на основі списків Jira, вона містить близько 5000 примірників.

5.4. Опис основних рішень за видами забезпечення інформаційної системи для ІТ-підприємства

При створенні ІС для ІТ-підприємства має місце проблема частоті зміни напрямку роботи, обумовлена вдосконаленням побудови моделей предметної області та алгоритмів вирішення задач. Зазвичай уточненню піддаються саме моделі предметної області, а алгоритми вирішення задач залишаються відносно постійними. Таким чином, виникає проблема наділення компонентів САП БЗ функціональністю, що забезпечує сприйняття змін у структурі оброблених об'єктів без зміни вихідних даних.

Така функціональність компонентів забезпечує гнучкість системи, що, з одного боку, знімає частину рутинної роботи з програмістів, з іншого – дозволяє самостійно налаштувати систему. Тому при розробці компонентів необхідно відділити методи створення та зберігання інформації про структуру та зміст предметної області від методів її обробки. Відповідно до цієї рекомендації запропоновано забезпечити зберігання БЗ у БД, а методи створення БЗ, управління БД і метод міркування за правилами реалізувати окремими модулями.

Отримана БЗ разом з інтегрованою машиною виведення утворюють продукційну експертну систему, результатом роботи якої є нові факти про розглянуті проблеми. Компонент дозволяє переглядати і зберігати результати виведення, а також відстежувати послідовність того, як спрацювали правила (елемент пояснення).

На рис. 1 наведено фрагмент архітектури САП БЗ ІТ- підприємства на основі БЗ, для якого розглянемо основні застосовані рішення з окремих частин забезпечення системи.

В даний час існують як комерційні, так і вільно поширювані сервіси, за допомогою яких можна реалізувати механізм міркування на основі продукцій. У даному дослідженні з цією метою було обрано можливості, які виникають в результаті взаємодії ІТ-продуктів Confluence та JIRA. Вибір обумовлений можливістю створення корпоративного Інтернет-порталу в організації, використання вікі-технології для створення БЗ, відстеження складових частин плану ІТ-проекту.

Підсистема цілісності реалізує інтерфейс користувача для доступу до БЗ і призначена для роботи (виконання операцій створення, модифікації і видалення) з фактами і правилами, наведеними в узагальненому вигляді в одній з БЗ. При цьому правила і факти можуть бути як абстрактними (зразки правил і фактів), так і конкретними (екземпляри

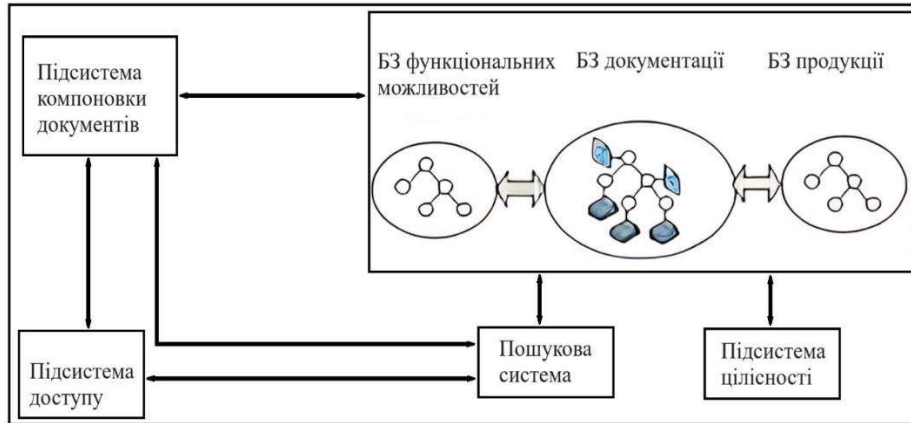


Рис. 1. Фрагмент архітектури системи автоматизованої побудови бази знань ІТ- підприємства на основі бази знань

правил і фактів). Факти і правила групуються відповідно до їх предметної класифікації, утворюючи БЗ. Кожна БЗ має унікальне ім'я і сприймається підсистемою цілісності як надана цій підсистемі функція, доступ до якої здійснюється через підсистему доступу. Підсистема цілісності розроблена з використанням HTML, CSS, JS і PHP.

Підсистема доступу реалізує призначений для користувача інтерфейс доступу до САП БЗ, передає машині виведення ідентифікатор БЗ або список правил і фактів у форматі машини виведення, виконує розмежування прав доступу.

Графічний редактор правил є частиною підсистеми цілісності. Редактор дозволяє використовувати окремі графічні примітиви для відображення всіх елементів правил; привласнювати окремим фактам суб'єктивні ймовірності у вигляді коефіцієнтів впевненості; наочніше відображати тип виконаних дій; відображати логічні оператори в умовах правил (або/ні). Редактор дозволяє також у простій і наочній формі спроектувати продукційне правило, задати умову і дію.

Головним елементом інформаційного забезпечення є БД для зберігання БЗ і початкових умов. Фрагмент моделі цієї БД наведено на рис. 2.

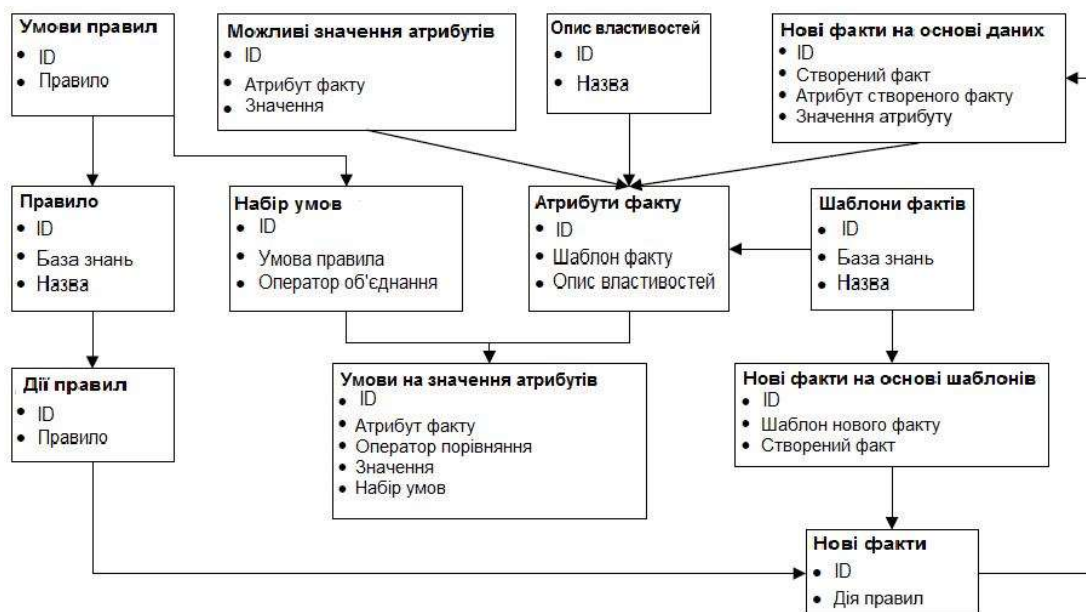


Рис. 2. Фрагмент моделі бази даних для зберігання бази знань

БД забезпечує зберігання БЗ, кожна з яких містить: факти; початкові умови; продукційні правила, які керують процесом формування фактів; заздалегідь заданий список обчислювальних процедур, що дозволяють перетворювати чисельні значення змінних.

Взаємодію з БД здійснюють підсистема компонування документів і пошукова підсистема. Підсистема доступу реалізує функції з'єднання з БД, відображення всіх таблиць, доступу до їх вмісту, а також виконання стандартних операцій і запитів (додавання, модифікації, видалення, пошуку). Модуль реалізований мовою SQL.

Інтернет-портал реалізований засобами Confluence [15]. Його основне завдання полягає у підготовці інформації (фактів і правил) з БЗ для її представлення користувачам у прийнятному для них візуальному вигляді.

Програмний інтерфейс доступу до БД реалізує функції з'єднання з БД, відображення всіх таблиць БД, доступу до їх вмісту, а також виконання стандартних операцій і запитів (додавання, модифікації, видалення, пошуку). Модуль реалізовано засобами мови SQL.

Таким чином, САП БЗ забезпечує завантаження візуальних моделей, їх перетворення у внутрішній універсальний формат (з метою побудови моделі предметної області), моделювання продукцій, їх генерацію, а також можливість спільної і розподіленої роботи всі учасників проекту.

6. Обговорення результатів дослідження

Для оцінки результативності та ефективності застосування САП БЗ було обрано дані з системи обліку трудовитрат і системи обліку та обробки дефектів. Ці системи є складовими загальної ІС управління проектами ІТ-підприємства. Аналіз даних виконувався за періоди часу однакової довжини, які було обрано з періодів часу до і після впровадження САП БЗ.

Впровадження САП БЗ у базові процеси ІТ-підприємства за виконаними розрахунками дозволило підвищити якість базових процесів (процесу проектування та розробки ПЗ на 15 %, процесу супроводу ПЗ на 17%), досягти покращення документації за рахунок появи нових споживчих властивостей (додавання гіпертексту, метаінформація, автоматичне формування документів), підвищити якість процесу пошуку інформації, зменшити трудовитрати на розробку документації на ПЗ, зменшити трудовитрати на супровід ПЗ. Так середній час на аналіз і прийняття рішення за повідомленням про помилки був зменшений на 26 %, середній час на відповідь за повідомленням з питаннями про експлуатацію ПЗ – на 33 %.

7. Висновки

У даному дослідженні було розглянуто особливості існуючого методу автоматизованої побудови та поповнення БЗ системи процесного управління. Запропоновано удосконалення предикатних моделей, які описують артефакти ІТ-проекту, та вдосконалення автоматизованої побудови БЗ шляхом розрахунку ймовірності правил на основі частоти їх появи у вхідних трасах подій.

Базуючись на результатах розгляду, було здійснено модернізацію етапів 1 та 2 існуючого методу. Виконано перевірку можливості застосування запропонованого модернізованого методу під час вирішення задач підтримки прийняття рішень за умови виявлення вторгнень для процесів у комп'ютерних системах.

Наведено опис основних характеристик САП БЗ, що враховують особливості модернізованого методу при вирішенні задач прийняття рішень на ІТ-підприємстві.

Проведено апробацію запропонованих рішень в процесі автоматизованого управління проектами ІТ-підприємства. У ході автоматизованої побудови продукційної БЗ виявлено переваги та недоліки методу.

Подальшим розвитком модернізованого методу може бути аналіз успішності виконання прогнозів за допомогою БЗ і корегування на основі результатів цього аналізу вихідного набору предикатів.

Перелік посилань:

1. Tuzovsky A.F., Yampolsky V.Z. The system approach to knowledge management systems designing and development. *Proceedings KORUS'2003*. Ulsan, 2003. P. 213–216.
2. Servin G. ABC of Knowledge Management. Caroline De Brian Publication, 2005. URL: https://www.fao.org/fileadmin/user_upload/knowledge/docs/ABC_of_KM.pdf (дата звернення: 11.07.2024).
3. Alavi M, Leidner D. E. Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *Management Information Systems Quarterly*. 2001. Vol. 25, № 1. P. 11-115.
4. Canadas J., Palma J., Tunez S. InSCo-Gen: A MDD tool for Web rule-based applications. *Lecture Notes in Computer Science*. 2009. Vol. 5648. P. 523–526.
5. Чала О. В. Принцип та метод еволюційної побудови бази знань на основі аналізу логів ІС процесного управління. *Бионика интеллекта*. 2017. № 1 (88). С. 80–84.
6. Левикін В.М. Концепція автоматизованої побудови бази знань у системі процесного управління. *Бионика интеллекта*. 2017. № 2 (89). С. 77–83.
7. Levykin V., Chala O. Method of automated construction and expansion of the knowledge base of the business process management system. *EUREKA: Physics and Engineering*. 2018. № 4. P. 29-35.
8. Richardson M., Domingos P. Markov logic networks. *Machine Learning*. 2006. №62 (1-2). P. 107–136. <https://doi.org/10.1007/s10994-006-5833-1>.
9. Singla, P., Domingos, P. Discriminative Training of Markov Logic Networks. *Proceedings of the 20th AAAI conference on Artificial intelligence*. 2005. P. 868-873.
10. Рудометкина М.Н., Спицын В.Г., Болотова Ю.А. Метод построения иерархической предикатной модели процесса. *Фундаментальные исследования*. 2014. № 9 (часть 12). С. 2666-2671
11. Yang H., van Dongen B.F., ter Hofstede A. H.M., Winn M.T., Wang J. Estimating Completeness of Event Logs. URL: <http://bpmcenter.org/wp-content/uploads/reports/2012/BPM-12-04.pdf> (дата звернення 14.07.2024).
12. Богачева Н. М. О разработке системы управления знаниями предприятий технического сервиса. *Научный журнал КубГАУ*. 2012. №80 (06).
13. Создание Agile-проекта в Jira. *ITFB*. URL: <https://itfb.com.ua/sozдание-agile-proekta-v-jira/> (дата звернення: 27.07.2024).
14. Blokdyk G. ETOM A Complete Guide - 2021 Edition. The Art of Service - ETOM Publishing, 2020. 313 p.
15. Sobia Publication. Learn JIRA with real-world examples + Confluence bonus: Learn to work on, manage & administer agile projects with this comprehensive Book on JIRA Software & Confluence. 52 p.

Надійшла до редколегії 01.08.2024 р.

Малькова Ірина Анатоліївна, асистент кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: iryna.malkova@nure.ua, ORCID: <http://orcid.org/0000-0001-6109-8193>

Макєнко Владислав Сергійович, розробник Frontend-елементів, фрілансер, м. Харків, Україна, e-mail: mvmasters735@gmail.com