

БАГАТОМОДЕЛЬНЕ ПРЕДСТАВЛЕННЯ ДАНИХ ДЛЯ ОНТОЛОГО-КЕРОВАНОЇ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Інститут кібернетики імені В.М. Глушкова Національної академії наук України

Вступ

Сьогодні результати розв'язання задач прийняття рішень є результатом поєднання та інтеграції знань, розуміння та ідей розв'язання множин взаємозв'язаних задач з різних предметних областей, кожна з яких має свої специфічні умови та природу. В рамках такого прийняття рішень людині, що приймає рішення (ЛПРу), необхідно врахувати множину властивостей, що визначаються та використовуються одночасно. Це вимагає розгляду процесів, структур, ресурсів, навколишнього середовища, а також взаємодії між акторами процесу прийняття рішень.

При цьому необхідно враховувати, що прийняття рішень відбувається в умовах інформаційної та реалізаційної неоднорідності, розподіленості та автономності інформаційних ресурсів системи.

Інформаційна неоднорідність ресурсів полягає в різноманітності їх прикладних контекстів. Реалізаційна неоднорідність джерел проявляється у використанні різноманітних комп'ютерних платформ, засобів управління базами даних, моделей даних і знань тощо. Дані для прийняття рішень природно організовані в різних форматах та моделях, що включають структуровані, напівструктуровані та неструктуровані дані. Для багатьох задач прийняття рішень використання тільки однієї моделі даних, як правило реляційної, є недостатньо.

Інформаційна підтримка прийняття рішень є невід'ємною складовою в реалізації сучасних програмних засобів знаннеорієнтованої підтримки прийняття рішень.

Ця підтримка реалізується через кількісну та якісну частини інформаційної підтримки. При цьому процес прийняття рішень може бути по-різному визначений із застосуванням різних даних.

Аналіз останніх досліджень та постановка завдання

Сьогодні для розв'язання проблеми «різноманітності» даних пропонується використовувати багатомодельне представлення даних. Існують публікації, наприклад огляд [1], які стосуються багатомодельного управління даними та надають загальний опис і характеристики багатомодельності об'єктно-реляційних (реляційних) СУБД, баз даних *NoSQL*, порівняльні дослідження тощо. Представлене в них багатомодельне представлення даних реалізується через дві основні концепції багатомодельної системи управління даними: багатоваріантне зберігання (*polyglot persistence*) [2] та мультимодельне зберігання (*multi-model*) [3]. Використання багатоваріантного зберігання означає, що дані зберігаються в різних базах даних, кожна з яких підтримує власні моделі даних. Мультимодельна база даних розроблена для підтримки декількох моделей даних на одній інтегрованій системі управління даними.

Але як правило ці публікації стосуються розгляду особливостей СУБД, особливостей форматів даних і їх використання та, як правило, не стосуються особливостей інформаційної підтримки прийняття рішень.

Мета

Метою дослідження є розроблення моделі багатомодельної інформаційної підтримки прийняття рішень, що базується на онтологічних методах представлення та обробки знань.

Основна частина

В роботі будемо розуміти під підтримкою прийняття рішень інтелектуальну комп'ютерну технологію посилення можливостей ЛППР в процесі спостереження за станом проблемної області, діагностики проблемних ситуацій і цілей дій, планування дій та генерацію способів їх реалізації, формування раціональних варіантів рішень з використанням експертних знань і методів моделювання та оптимізації.

При цьому необхідно врахувати наступні аспекти прийняття рішень:

1. Дані та інформація. Як зберегти високий рівень узгодженості між даними, інформацією і знаннями, коли вони перетворюються в різні формати різної природи.

2. Використання контекстів та інтеграція точок зору, використання та адаптація кількісних та якісних методів.

3. Створення процесу прийняття рішень на різних рівнях прийняття рішень в умовах напівструктурованих та неструктурованих задач.

4. Розв'язання проблеми представлення процесу прийняття рішень та візуалізації результатів прийняття рішень.

5. Проблеми організації процесу прийняття рішень.

Сьогодні в якості основи для реалізації перерахованих аспектів прийняття рішень розглядаються онтології, як засіб явного розуміння, представлення та використання знань проблемних областей та процесів прийняття рішень з врахуванням особливостей горизонтальних/ вертикальних зв'язків між задачами, моделями, методами, реалізаціями та різними шарами прийняття рішень. При цьому всі знання, що використовуються, розглядаються в розрізі знань, що описують контент, та знань, що описують контекст.

Під онтологією [4] будемо розуміти систему, що описує структуру певної проблемної області або множини проблемних областей та складається з множини класів понять, зв'язаних відношеннями, їх визначень та аксіом, що задають обмеження на інтерпретацію цих понять в рамках проблемної області або їх множини.

Така онтологія для підтримки прийняття рішень базується на взаємопов'язаній множині онтологій, що представляє собою багаторівневу асоціативну структуру [4]:

$$O = \langle O_{meta}, O_{core}, O_{cntxt}, \{O_{DM}\}, O_R, O_{user}, Inf \rangle$$

де O_{meta} – мета-онтологія; O_{core} – базова онтологія; O_{cntxt} – контекстна онтологія; $\{O_{DM}\}$ – множина онтологій представлення процесу прийняття рішень, що включає представлення задач та їх розв'язання на рівні проблемної області, онтологій предметно-формального та формального представлення та реалізацій цього процесу; O_R – онтологія реалізацій, що включає опис програмного забезпечення для підтримки прийняття рішень; O_{user} – онтологія представлення користувача та взаємодії з ним; Inf – модель машини виведення, що асоціюється з множиною онтологій O .

Мета онтології полягає в тому, щоб забезпечити інтегровану концептуальну основу прийняття рішень для того, щоб вона була визначена, зрозуміла, структурована, представляла явища при прийнятті рішень та могла бути використаною в системі підтримки прийняття рішень (СППР).

Для реалізації онтологічного представлення може бути використані декларативна мова *RDF* разом із своїм онтологічним розширенням *RDFS* (*RDF Schema*) та мова *OWL* (онтологічна мова, що заснована на логіці опису та сумісна з *RDF*). Такий вибір визначається з наступних причин. По-перше, ці мови набагато виразніші порівняно з іншими мовами онтологій. По-друге, ці мови мають можливість підтримувати семантичну сумісність для обміну та спільного використання знань в рамках СППР і дозволяють автоматизованим

міркуванням використовувати автоматизовані процеси. Це дає змогу представити онтологію, як певну базу знань, що реалізується як спрямований позначений граф. *RDF* можна використати для створення онтологічного представлення певних проблемних областей та відповідних технологій прийняття рішень для забезпечення автоматизованої обробки такого онтологічного представлення.

Модель графа представляється набором вузлів, відношень, властивостей та міток: $G = (V, E, \Sigma, L)$, де V – скінченний набір вузлів або вершин; $E \subseteq V \times V$ – це набір ребер, що представляє бінарний зв'язок між елементами в V ; Σ – набір міток; L є функцією відображення, що визначає позначені ребра з множини $V \times \Sigma \times V$.

Для роботи з даними *RDF* використовують мову запитів *SPARQL* (*SPARQL Protocol and RDF Query Language*). Мова *SPARQL* дає можливість реалізувати функції маніпуляції з даними графових баз даних у форматі *RDF*, тобто баз даних, що зберігають дані у графоподібній структурі, де вузли представляють сутності, а ребра – зв'язки між ними.

При прийнятті рішень необхідно враховувати, що використання інформації та знань в процесі прийняття рішень, як правило, відбувається в контексті складної структури процесу прийняття рішень, який часто формується за допомогою цілого ряду чинників [5]. В цьому випадку прийняття рішень розглядається через один або декілька взаємопов'язаних контекстів (моделі деякого контексту), в яких хтось (актор) щось робить (дія) з деяких причин (цілі) в певних умовах (середовище) для якогось (об'єкт) за допомоги деякого (засоби) у вигляді (представлення), використовуючи щось (можливості), десь (місце розташування) та колись (час). Представлення контексту складається з змісту, що базується на онтологіях, які охоплюють певну частину моделі контексту.

Для опису контексту необхідно визначити поняття та конструкції, які визначають природу, структуру та

представлення процесу формування та прийняття рішень і відповідних складових областей, які описують такий процес.

Для конкретизації контексту необхідно використовувати відповідні атрибутивні дані, які повинні бути представлені в форматі певної реляційної (об'єктно-реляційної) системи управління базами даних (СУБД). В цьому випадку реалізується інформаційна частина процесу прийняття рішень, що базується на реляційній моделі даних, тобто через множину взаємозв'язаних таблиць. Реляційна модель даних працює з структурованими даними. Для таких даних характерно, що вони мають визначені типи даних, існують у попередньо визначених форматах, зберігаються в таблицях реляційної бази даних. Відповідно реляційна модель даних визначає вимоги щодо програмних засобів реалізації збереження даних, організації доступу до цих даних та обробку цих даних. При цьому кожному джерелу даних, що представляє інформаційну частину та описується реляційною моделлю, ставиться у відповідність набір зв'язків з іншими джерелами (будь-якого рівня), набір обмежень цілісності та набір допустимих операцій.

Сьогодні при прийнятті рішень традиційно відокремлюють дані для прийняття рішень від оперативних даних. Тобто, оперативні дані зберігаються в системі обробки транзакцій в реальному часі (*OLTP*). Потім ці дані обробляються та перетворюються для зберігання в системі прийняття рішень, що використовує сховища даних та засоби оперативної аналітичної обробки (*OLAP*) і засоби інтелектуального аналізу (*Data Mining*).

Для роботи з реляційними (атрибутивними) даними можна використати, наприклад, мову запитів *SQL*, технологію роботи з даними *Entity Framework*.

Процес прийняття рішень базується на трьох стратегіях прийняття рішень: створення, інтеграція та адаптація [6]. Створення означає "абсолютно новий процес" прийняття рішень або прийняття рішень "на порожньому місці". Така концепція прийняття рішень реалізує прийняття

рішень в проблемній ситуації, коли вона виникла вперше та не існує відповідних визначених моделей, методів та реалізацій, які би можна використати як основу для процесу прийняття рішень. Інтеграція означає концепцію реалізації прийняття рішень, згідно якої процес прийняття рішень об'єднує в собі компоненти різних існуючих процесів прийняття рішень. Адаптація означає концепцію реалізації прийняття рішень, згідно якої вже було побудовано процес прийняття рішень і для розв'язання задачі можна змінити деяку частину(и) існуючого процесу або розширити існуючий процес деякою новою частиною(ами). Стратегію адаптація можна реалізувати за допомогою використання прецедентів та прецедентного підходу в процесі прийняття рішень.

Це визначає необхідність в використанні засобів зберігання результатів реалізацій прийняття рішень у вигляді або сесій прийняття рішень у разі не закінченого процесу прийняття рішень, що зберігають опис проблеми чи проблемної ситуації, моделі, методу (сценарію) розв'язання та з відповідними характеристиками прийняття рішень, або прецедентів, що зберігають опис проблеми чи проблемної ситуації разом із докладним описом дій, що виконували при розв'язанні даної проблеми чи у цій проблемній ситуації, та описом отриманого кінцевого результату.

В цьому випадку можна використати документи у вигляді множин пар ключ-значення. Документи мають динамічну схему. Динамічна схема означає, що документи не обов'язково повинні мати однаковий набір полів або структуру, а загальні поля в документах можуть містити дані різних типів.

З представленого можна зробити висновки, що при онтолого-керованому прийнятті рішень в рамках системи підтримки прийняття рішень необхідно підтримати роботу з:

- графовою інформацією;
- атрибутивною інформацією;
- інформацією у вигляді певних документів.

Це вказує на необхідність управління гетерогенними даними при прийнятті рішень.

Для того, щоб використати та інтегрувати реляційне представлення (атрибутивна інформація), графове представлення (графова інформація), представлення у вигляді документів використовують концепцію багатомодельного представлення даних, тобто для роботи з даними необхідно використовувати декілька моделей даних, які мають різну природу.

Як вже зазначалось, що для реалізації багатомодельного представлення даних використовують багатоваріантне зберігання (*polyglot persistence*) [2] та мультимодельне зберігання (*multi-model*) [3]. Сьогодні до них можна ще віднести хмарне зберігання (*cloud-based*) [7], до них також можна віднести і представників СУБД з мультимодельного зберігання, наприклад *Couchbase*, *MS SQL Server*, *Oracle Database* та інші.

Polyglot persistence (багатоваріантне зберігання) базується на принципі, що у межах однієї системи підтримки прийняття рішень доводиться зберігати даних і розв'язувати різні задачі за допомоги різних СУБД, кожний з яких підтримує свою модель даних. Використання *Polyglot persistence* в рамках підтримки прийняття рішень на основі декількох баз даних з різними логічними моделями має наступні переваги: хороша масштабованість застосування, ефективне управління різнорідними даними, більш швидкий час відгуку і як наслідок більш висока продуктивність. Але це вимагає розробки проміжного програмного забезпечення для інтеграції всіх баз даних, що веде до складності та підвищення вартості експлуатаційних витрат.

Multi-model бази даних – це спосіб розв'язання проблеми управління *Polyglot persistence* і реалізує переваги керування даними без недоліків використання різних технологій баз даних для кожного типу даних.

Multi-model бази даних спрямовані на об'єднання різних логічних моделей даних в єдиний інтегрований засіб з

уніфікованою мовою запитів і *API* для всіх моделей, що підтримуються. Такі бази даних можна розуміти як бази даних, в яких зберігаються дані в різних форматах (таблиці, графи, документи, графіки, об'єкти тощо) та управляються однією системою управління даними. *Multi-model* бази даних можуть реалізовуватися в рамках архітектури складного управління (*Complex engine architecture*) (наприклад, *CouchBase*). В цьому випадку СУБД перетворює всі типи даних, що підтримуються в єдину модель ядра СУБД. Або такі бази даних реалізуються в рамках рівневої архітектури (*Layer-based architecture*) (наприклад, *Oracle Database 12c/19c/21c*). В цьому випадку СУБД підтримує різні моделі даних за допомогою механізму різних рівнів надбудов над ними. Дані зберігаються у відповідній моделі даних. Кожна модель даних має власний компонент, який взаємодіє з механізмом управління даними. *Multi-model* СУБД (*MS SQL Server 2019/2022, Oracle Database 12c/19c/21c*) пропонують декілька моделей даних – реляційну та нереляційну – у складі єдиної платформи.

Хмарне (*cloud-based*) зберігання визначається як база даних, яка працює на платформі хмарних обчислень. У випадку традиційної (*traditional cloud-based*) хмарної моделі бази даних компанія купує простір для віртуальної машини у постачальника хмарних послуг, а база даних розгортається в хмарі. Ця модель дозволяє користувачам встановлювати та підтримувати власні бази даних. Користувачі самостійно запускають бази даних у хмарі, використовуючи образ віртуальної машини. У цьому випадку хмарна база даних може бути звичайною базою даних, такою як *MS SQL Server* і *MySQL*, і ця традиційна база даних може бути встановлена, налаштована та підтримувана користувачем самостійно на хмарному сервері. У випадку використання хмарних баз даних як сервісу (*DBaaS*) компанія власник служби несе відповідальність за надання системи управління базами даних, конфігурацію, масштабування, налаштування продуктив-

ності, резервне копіювання, конфіденційність і контроль доступу. Користувачі купують доступ до служби бази даних, яка підтримується постачальником хмарних баз даних. Хмарні бази даних можуть підтримувати реляційні, *NoSQL* та багатомодельні моделі баз даних.

В Інституті кібернетики імені В.М. Глушкова НАН України при виконанні проєктів [6], що пов'язані з реалізацією процедур підтримки прийняття рішень на основі системної оптимізації та онтологічних засобів представлення та обробки знань, вибрано в якості багатомодельного представлення та управління даними концепцію багатоваріантного зберігання (*polyglot persistence*).

Використання багатоваріантного зберігання дозволяє представити дані в їхній найбільш природній моделі даних, використати найбільш ефективні засоби маніпулювання даними та реалізувати еволюційний принцип до вже створеного програмного забезпечення для управління даними різної природи.

Для представлення онтологій підтримки прийняття рішень, що створені в програмному середовищі *Protégé* та представляють собою графову базу даних, використовується *RDF*, а в якості мови запитів до такої графової бази даних використовується мова *SPARQL*. Такі графові бази даних *RDF* спрямовані на семантичну обробку, часто з можливістю комбінування інформації між структурованими та неструктурованими даними. Графові бази даних *RDF* можуть бути сумісними з *ACID* (*atomicity, consistency, isolation, durability*) [8].

В якості засобу для реалізації роботи з реляційною моделлю даних та для роботи з відповідними інформаційним об'єктами використовується засоби *MS SQL Server* та технологія роботи з даними *Entity Framework*. *Entity Framework Core* (*EF Core*) є об'єктно-орієнтованою, розширюваною технологією від компанії *Microsoft* для доступу до даних. *EF Core* є *ORM*-інструментом (*object-relational mapping* – відображення даних на реальні

об'єкти). *EF Core* дозволяє абстрагуватися від самої бази даних та її таблиць і працювати з даними як з об'єктами та незалежно від типу бази даних. *EF Core* надає універсальний *API* для роботи з даними.

Для реалізації зберігання даних у вигляді певних документів, зокрема *JSON*-документів, використовується *NoSQL* СУБД *MongoDB* [9]. В основі роботи *MongoDB* лежать концепції документів та колекцій – ієрархічних структур, що містять пари "ключ-значення" (поля). Колекція є групою документів *MongoDB*. Колекція існує в одній базі даних. Документи у колекції можуть мати різні поля. Інформація в документах може бути подана у різних типах даних. *MongoDB* використовується для зберігання сесій прийняття рішень, збереження прецедентів розв'язання задач прийняття рішень з метою повторного використання при прийнятті рішень.

Таким чином, розглянуте багатомодельне представлення даних для підтримки прийняття рішень дає змогу зберігати та використовувати різні моделі даних, такі як реляційна (атрибутивна), документна (ключ-значення), графічна, в одній системі підтримки прийняття рішень, підтримувати зміну джерел даних (наприклад, баз даних) у відповідь на зміну предметних областей, що є характерною для прийняття рішень та вимагає незалежного розгляду від програмного забезпечення, повторно використовувати дані для підтримки різноманітних проблемних ситуацій.

Висновки

Представлений підхід до реалізації багатомодельного представлення для підтримки прийняття рішень дає можливість реалізувати інформаційну частину інтелектуальної знаннє-орієнтованої підтримки прийняття рішень, що дозволяє виробляти найбільш прийнятні рішення в певних проблемних ситуаціях та базується на принципах інженерії знань для сукупності проблемних областей.

Версію такого багатомодельного представлення даних було розроблено в рамках науково-дослідної роботи "Розробити типові онтологокеровані процедури

системної оптимізації для розв'язання прикладних задач".

Література

1. Jiaheng L., Holubova I. Multi-model Databases: A New Journey to Handle the Variety of Data. *ACM Computing Surveys*. 2019. Vol. 52. P. 1–38.
2. Fowler M., Sadalage P. J. *NoSQL DISTILLED. A Brief Guide to the Emerging World of Polyglot Persistence*. 1st ed. Crawfordsville : Addison-Wesley Professional, 2012. 192 p.
3. Liu Z. H. et al. Multi-Model Database Management Systems – a Look Forward. *Lecture Notes in Computer Science. Vol. 11470. Heterogeneous Data Management, Polystores, and Analytics for Healthcare. VLDB 2018 Workshops, Poly and DMAH Rio de Janeiro, Brazil, August 31, 2018 Revised Selected Papers* / ed. by V. Gadepally et al. Cham, 2019. P. 16–29.
4. Чаплінський Ю. П. Онтологічні складові підтримки прийняття управлінських рішень. *Наукові праці НУХТ*. 2013. № 48. С. 65–68.
5. Чаплінський Ю. П., Субботіна О. В. Онтологія та контекст при розв'язанні прикладних задач прийняття рішень. *Штучний інтелект*. 2016. № 2. С. 147–155.
6. Чаплінський Ю. П. Контекстно-онтологічна системна оптимізація проблемно-орієнтованої підтримки прийняття рішень. *Нові інформаційні технології, моделювання та автоматизація* : монографія / за заг. ред. С. В. Котлика. Одеса, 2022. С. 6–44.
7. Neu W. et al. *An Introduction to Cloud Databases*. 1st ed. Sebastopol : O'Reilly Media, Inc., 2019. 48 p.
8. Hoffer J. A., Ramesh V., Topi H. *Modern database management*. 13th ed. Harlow : Pearson, 2019. 600 p.
9. Bradshaw S., Eoin B., Chodorow K. *MongoDB: The Definitive Guide*. 3rd ed. Sebastopol : O'Reilly Media, Inc., 2019. 511 p.

Чаплінський Ю.П., Субботіна О.В

БАГАТОМОДЕЛЬНЕ ПРЕДСТАВЛЕННЯ ДАНИХ ДЛЯ ОНТОЛОГО-КЕРОВАНОЇ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Прийняття рішень з використанням онтолого-керованих технологій сьогодні є результатом поєднання та інтеграції знань та даних різної природи і призначення, розв'язання множин взаємозв'язаних задач прийняття рішень, синтезу різних точок зору на проблему та процес прийняття рішень. Для багатьох задач прийняття рішень використання тільки однієї моделі даних, як правило реляційної, є недостатньо. Сьогодні для розв'язання проблеми «різноманітності» даних використовують концепцію багатомодельне управління даними. В статті розглянуті сучасні аспекти прийняття рішень, що базуються на використанні онтологій, як засобу явного розуміння та представлення областей та процесів прийняття рішень, що інтегрує методи системного, процесного та ситуаційного аналізу. В статті визначено можливі види представлень (моделей) даних, які необхідні для реалізації онтолого-керованого прийняття рішень, також визначено відповідні види інформації. В роботі для реалізації багатомодельного управління даними розглянуто та проаналізовано багатоваріантне зберігання (polyglot persistence), мультимодельне зберігання (multi-model) та хмарне зберігання (cloud-based). В роботі зроблено висновок про вибір багатоваріантне зберігання, як засобу багатомодельного представлення даних в рамках проектів, які виконувалися авторами статті. В статті описано вибрані програмні засоби для реалізації визначеного багатомодельного представлення даних.

Ключові слова: система підтримки прийняття рішень; прийняття рішень; онтологія; багатомодельне управління даними; багатоваріантне зберігання даних.

Chaplinsky Y.P., Subbotina O.V.

MULTI-MODEL DATA REPRESENTATION FOR ONTOLOGY-DRIVEN DECISION-MAKING

Nowadays ontology-driven decision-making is the result of the combination and integration of knowledge and data of different nature and purpose, solving multiple interconnected decision-making problems, synthesizing different points of view on the problem and the decision-making process. For many decision-making problems, the use of only one data model, usually a relational one, is not enough. Today, to solve the problem of "diversity" of data, the concept of multi-model data management is used. Modern aspects of decision-making based on the use of ontologies as a tool of understanding and presentation of decision-making areas and processes, which integrates the methods of system, process and situational analysis are discussed. The possible types of data representations (models) that are necessary for the implementation of ontology-guided decision-making, and also defines the corresponding types of information are defined. In the work, polyglot persistence storage, multi-model storage and cloud-based storage were considered and analyzed for the implementation of multi-model data management. The conclusion is made about the choice of multi-variant storage as a means of multi-model presentation of data within the framework of projects carried out by the authors of the article. The selected software tools for implementing the defined multi-model data representation are described.

Keywords: decision-support system; decision-making; ontology; multi-model data management; polyglot persistence.