УДК 004.8:004.42:681.518 DOi 10.30837/ bi.2025.2(103).15

В. В. Мяляренко[1], О. Ю. Чередніченко[2]

[1]НТУ ХПІ, м. Харків, Україна, vladyslav.maliarenko@cs.khpi.edu.ua,
ORCID iD: 0009-0009-6064-061X

[2]НТУ ХПІ, м. Харків, Україна, olga.cherednichenko@khpi.edu.ua,
ORCID iD: 0000-0002-9391-5220

МОДЕЛІ ОБРОБКИ ТЕКСТОВИХ БІЗНЕС-ПРАВИЛ У СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

У статті розглянуто проблему формалізації та автоматизованої обробки текстових бізнес-правил у системах підтримки прийняття рішень з урахуванням наявної структури даних та контексту предметної області. Запропоновано математичні моделі, що охоплюють три ключові етапи обробки правил: визначення структурних компонентів текстового бізнес-правила, побудову формальної таблиці рішень у нотації DMN та валідацію синтаксису й семантики згенерованої моделі за правилами логічної узгодженості. Модель структурного аналізу забезпечує формальне виділення умов, дій та залежностей із урахуванням доступних даних; модель генерації DMN визначає відповідність текстових конструкцій елементам таблиці рішень і враховує контекст системи підтримки рішень; модель валідації дозволяє виявляти логічні суперечності, неповноту та помилки узгодженості у формальній моделі. Представлено прототип програмної системи, що реалізує запропоновані моделі та дозволяє проводити експериментальне тестування їхньої ефективності. Результати експериментів демонструють коректність формалізації, повноту відображення бізнес-правил у DMN-таблиці та дієвість автоматичного виявлення структурних, синтаксичних і семантичних помилок.

БІЗНЕС-ПРАВИЛА, DMN, СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, ФОРМАЛЬНІ МОДЕЛІ, СТРУКТУРА ДАНИХ, СИНТАКСИЧНА ВАЛІДАЦІЯ, СЕМАНТИЧНА ВАЛІДАЦІЯ, АВТОМАТИЗОВАНЕ МОДЕЛЮВАННЯ

V. V. Maliarenko, O. Y. Cherednichenko. Models for Processing Textual Business Rules in Decision Support Systems. The paper addresses the problem of formalization and automated processing of textual business rules in decision support systems, taking into account the existing data structure and the contextual constraints of the decision-making domain. The study proposes mathematical models covering three key stages: identifying the structural components of a textual business rule, generating a formal DMN decision table, and validating the syntax and semantics of the resulting model in accordance with logical consistency rules. The structural analysis model extracts conditions, actions, and dependencies while incorporating the available data schema; the DMN generation model maps textual constructs to decision table elements with respect to the context of the decision support system; the validation model detects logical inconsistencies, incompleteness, and coherence errors in the formalized model. A prototype implementation is presented, enabling experimental evaluation of the proposed models. The results demonstrate correct formalization, completeness of business rule transformation into the DMN table, and effective detection of structural, syntactic, and semantic errors in the generated decision model.

BUSINESS RULES, DMN, DECISION SUPPORT SYSTEMS, FORMAL MODELS, DATA STRUCTURE, SYNTACTIC VALIDATION, SEMANTIC VALIDATION, AUTOMATED MODELING

Вступ

Сучасні системи підтримки прийняття рішень дедалі частіше спираються на явні бізнес-правила, які визначають логіку вибору альтернатив у складних організаційних і технічних системах. Стандарт Decision Model and Notation (DMN) надає формальну основу для подання такої логіки у вигляді виконуваних таблиць рішень, що сприяє прозорості, керованості та повторному використанню моделей рішень. Разом з тим на практиці бізнес-правила зазвичай формулюються у природній мові, а їх перетворення у формальні DMN-моделі потребує значних ручних зусиль і залишається джерелом помилок.

Останні дослідження демонструють потенціал використання штучного інтелекту для автоматизації подання логіки рішень і генерації DMN-артефактів з текстових описів [1]. Зокрема, показано, що поєднання мовних моделей зі структурними обмеженнями дозволяє значно скоротити час побудови таблиць рішень і підвищити їхню узгодженість зі схемою даних. Водночас проблема семантичної коректності згенерованих моделей, їхньої повноти та несуперечності залишається відкритою і потребує систематичного підходу до валідації.

У цій роботі пропонується комплексний підхід, який розглядає ідентифікацію семантики, генерацію DMN-таблиць і їх тестоорієнтовану валідацію як єдиний узгоджений процес.

1. Аналіз стану вирішеності задачі

Автоматизація формалізації бізнес-правил і побудови моделей рішень є активним напрямом досліджень у галузях систем підтримки прийняття рішень, бізнес-процесного моделювання та програмної інженерії. Застосування штучного інтелекту для генерації формальних представлень логіки рішень, зокрема у нотації DMN, продемонструвало свою перспективність

у зменшенні ручної роботи та підвищенні узгодженості моделей [1]. Водночас стандарт DMN визначає лише формат і семантику моделей рішень, не регламентуючи методів їх отримання з неформальних текстових описів [2].

У роботах, присвячених інтеграції DMN у бізнес-процеси, показано, що винесення логіки рішень у окремі DMN-артефакти підвищує прозорість і керованість процесів, зменшує складність BPMN-моделей і полегшує супровід рішень [3]. Водночас побудова DMN-таблиць у таких підходах, як правило, здійснюється вручну або напівавтоматично, що обмежує їх застосовність для великих і динамічних наборів правил.

Значна частина досліджень спрямована на вилучення логіки рішень із природномовних текстів за допомогою методів обробки природної мови. Запропоновані підходи використовують як правило-орієнтовані NLP-техніки, так і нейронні моделі для ідентифікації умов, дій і залежностей між ними [4], [5]. Такі методи демонструють здатність відновлювати елементи моделей рішень або діаграми вимог до рішень, однак зазвичай не забезпечують автоматичної генерації повністю виконуваних DMN-таблиць і не гарантують семантичної узгодженості з наявною схемою даних.

Поява великих мовних моделей (LLM) суттєво розширила можливості автоматичної генерації структурованих артефактів із тексту. Дослідження показують, що сучасні LLM здатні генерувати табличні структури та формалізовані правила, у тому числі у форматі, наближеному до DMN [6], [7]. Водночас такі моделі схильні до галюцинацій, логічних суперечностей і порушення структурних обмежень, що робить неможливим їх пряме використання без додаткових механізмів контролю.

Для підвищення структурної коректності результатів генерації запропоновано schema-aware підходи, які передбачають ін'єкцію схем даних, використання формальних форматів виводу та обмежень на синтаксис згенерованих структур [8]. Хоча ці методи покращують відповідність результатів формальним вимогам, вони не вирішують проблему семантичної коректності логіки рішень і не гарантують повноти покриття можливих комбінацій вхідних умов.

Окремим напрямом досліджень є Retrieval-Augmented Generation (RAG), що поєднує мовні моделі з пошуком релевантного зовнішнього контексту. Залучення доменних документів і правил дозволяє зменшити кількість пропусків і підвищити повноту згенерованих моделей [9], [10]. Разом з тим у літературі відзначається, що без додаткових механізмів семантичної фільтрації RAG може вводити суперечливі або нерелевантні знання, що негативно впливає на якість рішень [11].

Валідація DMN-таблиць є добре дослідженою задачею. Існуючі підходи охоплюють виявлення неповноти, перекриття правил, суперечностей і надлишковості за допомогою формальних методів та виконуваних рушіїв DMN [12], [13]. Проте більшість таких робіт розглядають валідацію як ізольований етап, який застосовується після побудови моделі, і не інтегрують її безпосередньо в процес автоматизованої генерації.

Отже, аналіз сучасних досліджень свідчить про наявність розриву між методами інтерпретації текстових бізнес-правил, генерацією формальних DMN-структур і їх систематичною семантичною валідацією. Це обґрунтовує необхідність комплексного підходу, у якому ідентифікація семантики, генерація моделей рішень і їх тестоорієнтована валідація розглядаються як єдиний узгоджений процес.

Текстові бізнес-правила, що використовуються у системах підтримки прийняття рішень, за своєю природою є неформальними специфікаціями логіки рішень. Вони формулюються природною мовою, орієнтовані на експертів предметної області та призначені для комунікації намірів і політик. Водночас практичне використання таких правил у програмних системах потребує їхнього перетворення у формальні, машинно-інтерпретовані моделі, зокрема у вигляді таблиць рішень DMN. Таким чином, ключовою проблемою є розрив між неформальним текстовим описом правила та його формальною реалізацією у вигляді виконуваної моделі прийняття рішень.

З точки зору теорії програмної семантики, текстове бізнес-правило можна розглядати як неформальну специфікацію програми, що задає відображення з простору вхідних станів у простір результатів рішення. Важливою особливістю цієї задачі є те, що відновлення семантики не може виконуватися у відриві від контексту використання правила. Інтерпретація тексту суттєво обмежується наявною схемою даних, яка визначає допустимі вхідні змінні, їх типи та домени, а також можливі результати рішення. Таким чином, постановку задачі ідентифікації бізнес-правил доцільно розглядати як задачу відновлення формальної семантики з неформального текстового опису з урахуванням типової та доменної семантики системи.

Метою даного дослідження є розроблення та експериментальна оцінка формалізованого підходу до автоматизованої обробки текстових бізнес-правил, який забезпечує відновлення їх формальної семантики та генерацію виконуваних DMN-таблиць рішень з контролем коректності, повноти та несуперечності результатів.

2. Методологія дослідження

Дослідження побудовано за принципом послідовного уточнення семантики. На першому етапі виконується ідентифікація бізнес-правил, у межах якої текстові описи інтерпретуються як джерело абстрактної семантики рішення. Цей етап формалізується як задача відновлення формального правила з тексту за

наявності обмежень схеми даних і доменного контексту. Для зменшення неоднозначності інтерпретації використовується підхід RAG, який дозволяє інтегрувати зовнішній доменний контекст у процес семантичного аналізу тексту. У результаті формується множина формальних правил, що визначають умови застосування та очікувані результати рішення.

Другий етап полягає у генерації таблиць рішень DMN як семантичного уточнення ідентифікованих формальних правил. Генерація розглядається не як механічне відображення логічних формул у табличну форму, а як процес побудови операційної семантики, що є семантично еквівалентною абстрактній специфікації. На цьому етапі виконується декомпозиція умов правил, визначення вхідних і вихідних атрибутів, а також фіксація політики виконання рішень відповідно до стандарту DMN. Згенеровані таблиці проходять синтаксичну перевірку, що гарантує їхню виконуваність.

Третій етап присвячено валідації згенерованих DMN-таблиць. На відміну від традиційних підходів, орієнтованих на ручний аналіз або формальну верифікацію, у роботі запропоновано тестоорієнтовану методологію валідації, яка відтворює практики, що застосовуються експертами предметної області. Ключовою особливістю цього етапу є незалежна генерація тестового покриття на основі тих самих джерел знань, що й генерація таблиць, але без використання їхньої структури. Тестові випадки формуються як конкретизації очікуваної семантики рішення для репрезентативних вхідних станів з урахуванням схеми даних і доменних обмежень.

Процес валідації організовано ітеративно. Після успішної синтаксичної перевірки DMN-таблиця тестується на згенерованому наборі тестів, а результати порівнюються з очікуваними значеннями, отриманими з референтної семантичної моделі. Якщо таблиця не задовольняє задані порогові критерії точності та покриття, ініціюється цикл уточнення, який може включати повторну генерацію тестів або повторну генерацію DMN-таблиці. Такий підхід дозволяє поступово зменшувати семантичні розбіжності між абстрактною специфікацією та операційною моделлю за обмеженого бюджету обчислювальних спроб.

Запропоновано інтегрований конвеєр, який поєднує стохастичні компоненти, засновані на великих мовних моделях, із детермінованими процедурами перевірки та виконання рішень. Це дозволяє зберегти інтерпретованість і керованість процесу, водночас зменшуючи обсяг ручної роботи та ризик помилок. Такий підхід забезпечує відтворювану основу для експериментального оцінювання та розроблення промислових систем керування бізнес-правилами та підтримки прийняття рішень.

Формалізація задачі спирається на інтерпретацію текстового правила та розглядає процес ідентифікації як відновлення формального семантичного представлення, узгодженого з наявною схемою даних і контекстом предметної області. Для цього необхідно чітко визначити вхідні дані, простір допустимих формальних правил та критерії коректності інтерпретації.

Нехай $t \in \Sigma^*$ позначає текстове бізнес-правило, сформульоване природною мовою. Вважається, що текст сам по собі не визначає однозначної формальної семантики, а задає множину потенційних інтерпретацій, допустимих з точки зору мовного формулювання. Інтерпретація цього тексту здійснюється не ізольовано, а з урахуванням формальної схеми даних системи підтримки прийняття рішень та контексту предметної області. Схема даних задається як трійка $\mathcal{S} = (\mathcal{A}, \mathcal{D}, \tau)$, де $\mathcal{A}$ є множиною атрибутів, $\mathcal{D}$ – множиною доменів значень, а $\tau : \mathcal{A} \to \mathcal{D}$ є відображенням типізації атрибутів. Контекст предметної області $\mathcal{C}$ розглядається як сукупність доменних тверджень, обмежень і визначень, які уточнюють семантику термінів, що використовуються у тексті правила, та накладають додаткові обмеження на допустимі інтерпретації.

Метою ідентифікації є побудова формального бізнес-правила. Простір формальних бізнес-правил визначається як множина об'єктів вигляду $r = \Phi, \Psi, \kappa$, де Φ є логічною формулою умов застосування правила, визначеною над атрибутами зі схеми $\mathcal{S}$, Ψ є множиною дій або результатів рішення, а κ містить метадані, зокрема інформацію про джерело правила, його пріоритет або ступінь впевненості. Формула умов Φ інтерпретується як предикат над простором вхідних станів, тоді як дії Ψ визначають відображення з простору вхідних значень у простір результатів рішення.

З формальної точки зору задача ідентифікації полягає у знаходженні такого правила r^*, яке є допустимою інтерпретацією тексту t за наявних обмежень схеми та контексту. Це відображення можна подати у вигляді функції

$$f : (t, \mathcal{S}, \mathcal{C}) \to r^*,$$

де r^* належить простору формальних правил і задовольняє умови семантичної та типової коректності. Допустимість формального правила визначається, по-перше, узгодженістю зі схемою даних, що означає використання виключно атрибутів, визначених у $\mathcal{S}$, та коректність усіх логічних і обчислювальних операцій відносно їхніх типів. По-друге, правило повинно бути узгодженим з контекстом предметної області, тобто не порушувати політики, зафіксовані в $\mathcal{C}$.

Таким чином, формальна постановка задачі ідентифікації бізнес-правил створює основу для побудови моделей генерації та валідації рішень, а також для аналізу якості й повноти бізнес-логіки.

3. Формальні моделі обробки текстових бізнес-правил

Для побудови математичної моделі необхідно формально визначити простори вхідних даних, простір допустимих семантичних інтерпретацій та критерії вибору оптимального правила.

Нехай $\mathcal{A}$ є скінченною множиною атрибутів, визначених у схемі даних системи підтримки прийняття рішень, а $\mathcal{D}$ – множиною доменів значень, асоційованих з цими атрибутами. Типізація атрибутів задається відображенням $\tau : \mathcal{A} \to \mathcal{D}$. Простір вхідних станів рішення визначається як декартів добуток доменів вхідних атрибутів і позначається $\mathcal{X} = \prod_{a \in \mathcal{A}_\mathcal{I}} \tau(a)$, де $\mathcal{A}_I \subseteq \mathcal{A}$ – множина атрибутів, що використовуються як умови. Аналогічно, простір результатів рішення визначається як $\mathcal{Y} = \prod a \in \mathcal{A}_\mathcal{O} \tau(a)$, де $\mathcal{A}_\mathcal{O} \subseteq \mathcal{A}$ – множина атрибутів, що використовуються як результати.

Формальне бізнес-правило задається у вигляді пари $r = (\Phi, \Psi)$, де Φ є логічною формулою над змінними з $\mathcal{A}_\mathcal{I}$, а Ψ – відображенням, що кожному допустимому вхідному стану ставить у відповідність результат рішення. Формула умов Φ інтерпретується як предикат

$$\Phi : \mathcal{X} \to \text{true}, \text{false},$$

тоді як дія Ψ визначається як часткова функція

$$\Psi : \mathcal{X} \rightharpoonup \mathcal{Y}.$$

Відповідно, семантика правила r задається як часткова функція

$$[\![r]\!] : \mathcal{X} \rightharpoonup \mathcal{Y},$$

яка визначена для тих і лише тих вхідних станів, для яких виконується умова Φ.

Текстове бізнес-правило t розглядається як джерело семантичної інформації, яке визначає функцію подібності між текстом та формальним правилом. Нехай $sim(t,r)$ є числовою мірою семантичної відповідності формального правила r та тексту t, що відображає ступінь узгодженості логічної структури та термінів правила з природномовним описом з урахуванням контексту $\mathcal{C}$. Тоді задача ідентифікації формулюється як задача оптимального вибору:

$$r^* = \arg\max_{r \in \mathcal{R}_{\text{valid}}} \text{sim}(t,r),$$

де оптимальне правило r^* є найкращою семантичною апроксимацією текстового опису серед усіх допустимих формальних інтерпретацій.

У практичній постановці ідентифікації бізнес-правил функція семантичної відповідності $sim(t,r)$ повинна одночасно відображати близькість формального правила r до змісту тексту t, узгодженість із схемою даних $\mathcal{S}$ та несуперечність доменному контексту $\mathcal{C}$. У підході на основі RAG ці три компоненти інтегруються в єдину цільову функцію через явне моделювання релевантного підконтексту $K \subseteq \mathcal{C}$, який вибирається механізмом пошуку й подається на вхід мовній моделі разом із текстом правила та інформацією про схему. Це дозволяє розглядати ідентифікацію як задачу максимізації апостеріорної правдоподібності формальної інтерпретації за умов обмеженого контексту.

Нехай оператор пошуку для заданого правила t повертає множину фрагментів контексту $K = \{k_j\}_{j=1}^m$, відібраних з $\mathcal{C}$. K визначається як результат максимізації семантичної релевантності в ембединг-просторі, тобто

$$K = \arg\max_{|K|=m} \sum_{k \in K} sim_{emb}\left(e(t), e(k)\right),$$

де $e(\cdot)$ – відображення у векторний простір, а sim_{emb} – міра близькості (наприклад, косинусна). Важливо, що ця подібність використовується лише для формування K, тоді як цільова функція ідентифікації оперує вже семантикою кандидата r.

Далі вводиться ймовірнісна інтерпретація генератора. Нехай $P_\theta(rt, \mathcal{S}, K)$ – умовний розподіл, що реалізується мовною моделлю з параметрами θ, яка генерує кандидатні правила r за умови тексту t, схеми $\mathcal{S}$ та контексту K. Тоді природною конкретизацією є визначення

$$sim(t,r) \equiv \log P_\theta(rt, \mathcal{S}, K) - \Omega(r; \mathcal{S}, K),$$

де Ω – штрафний функціонал.

Щоб зробити інтерпретованим і контрольованим, його доцільно розкласти на компоненти, що відповідають ключовим вимогам до правила: схемній узгодженості, типовій коректності, контекстній узгодженості та структурній придатності до подальшої компіляції в DMN. У цьому випадку можна записати

$$\Omega(r; \mathcal{S}, K) = \lambda_1, \pi_{\text{link}}(r; \mathcal{S}) + \lambda_2, \pi_{\text{type}}(r; \mathcal{S}) + \\ + \lambda_3, \pi_{\text{ctx}}(r; K) + \lambda_4, \pi_{\text{struct}}(r),$$

де кожний $\pi(\cdot) \geq 0$ є мірою порушення відповідної властивості, а λ_i задають ваги.

Компонент π_{link} формалізує вимогу, що всі змінні, використані у формулі умов і діях правила, мають бути зіставлені з атрибутами зі схеми. Компонент π_{type} відповідає за типову коректність: усі предикати в Φ та всі присвоєння у Ψ мають бути визначені для типів, що задані τ. У контексті DMN це також інтерпретується як умова коректності FEEL-виразів на рівні типів. Компонент π_{ctx} моделює узгодженість правила з контекстом. Його можна визначати як штраф за порушення доменних інваріантів або як штраф за відсутність логічної підтримки ключових тверджень у K. Нарешті, π_{struct} відповідає за структурну придатність правила до подальшої компіляції в DMN. Ідея полягає в тому, що навіть семантично коректне правило може бути сформульоване у вигляді, що ускладнює нормалізацію (наприклад, надлишкова вкладеність, неявні залежності, змішування кількох рішень в одному правилі). Типовим прикладом формалізації є штраф за складність формули умов, зокрема за кількість кон'юнктів/диз'юнктів та глибину дерева.

Після введення цієї конкретизації явно визначається, що LLM у зв'язці з RAG виконує роль стохастичного семантичного "декодера" формального правила, тоді як штрафний функціонал включає формальні вимоги схеми даних і предметної області, перетворюючи задачу на керований процес відновлення семантики, придатний до подальшої генерації та валідації DMN-моделей.

Після етапу ідентифікації бізнес-правила результатом обробки текстового опису є формальне правило, яке задає абстрактну семантику рішення у вигляді логічної умови застосування та відповідної дії або множини дій. Таке правило є машинно-інтерпретованим, однак воно ще не є безпосередньо виконуваною моделлю у складі системи підтримки прийняття рішень. Наступним кроком є перетворення цієї абстрактної семантики у конкретну операційну форму, сумісну зі стандартами виконання рішень, зокрема у вигляді таблиці рішень Decision Model and Notation. З концептуальної точки зору ця задача полягає у семантичному уточненні формального правила, тобто у переході від декларативного опису логіки до структурованої, детермінованої та виконуваної моделі.

У термінах програмної семантики формальне бізнес-правило, отримане на попередньому етапі, можна розглядати як абстрактну специфікацію програми прийняття рішення. Воно визначає, за яких умов повинні виконуватися певні дії, але не фіксує конкретний спосіб організації цієї логіки у вигляді окремих альтернативних випадків, порядку перевірки умов чи механізму розв'язання конфліктів. Натомість DMN-таблиця рішень задає операційну семантику цієї специфікації, явно визначаючи множину рядків правил, структуру вхідних та вихідних змінних, а також політику обробки ситуацій, коли кілька правил можуть бути застосовними одночасно. Таким чином, генерація DMN-таблиці є процесом уточнення семантики, у якому усувається недовизначеність, притаманна абстрактному правилу, і вводяться всі необхідні деталі для виконання.

Якісно задача генерації таблиці рішень полягає у побудові такої DMN-структури, яка є семантично еквівалентною ідентифікованому формальному правилу в межах допустимого простору вхідних станів, визначеного схемою даних і контекстом предметної області. Це означає, що для кожного допустимого набору вхідних значень таблиця рішень повинна або породжувати той самий результат, що й абстрактне правило, або, у випадках неповної специфікації, явно фіксувати відсутність застосовного рішення. Генерація таблиці не повинна вводити нових рішень, не передбачених семантикою правила, і не повинна втрачати жодних рішень, які правило дозволяє.

Водночас семантичне уточнення не є тривіальним механічним перетворенням. Формальне правило може містити складні логічні формули з кон'юнкціями, диз'юнкціями та неявними залежностями між умовами, тоді як DMN-таблиця вимагає явної декомпозиції логіки на набір дискретних рядків, кожен з яких відповідає певній комбінації вхідних умов. Тому генерація таблиці рішень передбачає нормалізацію умов, виділення релевантних вхідних атрибутів, розгортання складних логічних виразів у множину альтернативних випадків і, за потреби, доповнення таблиці службовими рядками для забезпечення повноти або коректної обробки винятків.

Суттєвим аспектом цієї задачі є вибір та фіксація політики виконання рішень, зокрема політики обробки збігів правил. У формальному правилі порядок або спосіб застосування умов може бути неявним або взагалі не визначеним, тоді як у DMN він має бути зафіксований через відповідну hit policy. Таким чином, генерація таблиці рішень включає не лише структурну декомпозицію логіки, а й прийняття семантичних рішень щодо того, як інтерпретувати потенційні конфлікти або перекриття умов, спираючись на метадані правила та доменний контекст.

Отже, задачу генерації DMN-таблиці доцільно розглядати як задачу побудови операційної семантики, яка є коректним і повним семантичним уточненням абстрактної семантики ідентифікованого бізнес-правила. Така постановка створює концептуальне підґрунтя для подальшої математичної формалізації процесу генерації таблиць рішень, у якій буде чітко визначено умови семантичної еквівалентності, критерії повноти та обмеження, накладені стандартом DMN і структурою даних системи підтримки прийняття рішень.

Математична модель генерації таблиці рішень ґрунтується на трактуванні формального бізнес-правила як абстрактної семантики рішення та DMN-таблиці як її семантичного уточнення. Метою генерації є побудова такої таблиці рішень, семантика якої є еквівалентною семантиці ідентифікованого правила в межах допустимого простору вхідних станів, визначеного схемою даних і доменними обмеженнями.

Нехай формальне правило задане у вигляді $r = (\Phi, \Psi)$, де Φ є логічною формулою над атрибутами зі схеми даних, а Ψ – відображенням, що визначає результат рішення для станів, у яких умова Φ виконується. Семантика цього правила визначається як часткова функція

$$[\![r]\!] : \mathcal{X} \rightharpoonup \mathcal{Y},$$

яка для кожного вхідного стану $x \in \mathcal{X}$ визначена тоді й лише тоді, коли $x \vDash \Phi$.

DMN-таблиця рішень формалізується як скінченна структура

$$\mathcal{D} = \langle I, O, R, H, \sigma \rangle,$$

де $I \subseteq \mathcal{A}$ є множиною вхідних атрибутів, $O \subseteq \mathcal{A}$ – множиною вихідних атрибутів, $R = 1,\ldots,m$ – множиною рядків таблиці, H – політика обробки збігів правил

(hit policy), а σ задає для кожного рядка $j \in R$ пару (Φ_j, Ψ_j), що відповідає умовам і результатам цього рядка. Кожна формула Φ_j є кон'юнкцією локальних обмежень над вхідними атрибутами, а Ψ_j – конкретизацією значень вихідних атрибутів.

Семантика DMN-таблиці визначається як відображення

$$[\![\mathcal{D}]\!] : \mathcal{X} \rightharpoonup \mathcal{Y}.$$

Задача генерації DMN-таблиці формулюється як задача побудови такої структури $\mathcal{D}$, що семантика таблиці є еквівалентною семантиці формального правила на допустимій області визначення. Формально це означає виконання умови семантичної еквівалентності

$$\forall x \in \mathcal{X}_C : \ [\![r]\!](x) = [\![\mathcal{D}]\!](x),$$

де $\mathcal{X}_C \subseteq \mathcal{X}$ – множина вхідних станів, допустимих з точки зору контексту предметної області.

Таким чином, математична модель генерації DMN-таблиці формалізує цей етап як задачу побудови операційної семантики, що є коректним семантичним уточненням формального бізнес-правила.

Після генерації DMN-таблиці бізнес-правило отримує форму, придатну до виконання у складі системи підтримки прийняття рішень. Проте сам факт успішної генерації таблиці не гарантує її коректності з точки зору семантики рішення, повноти охоплення допустимих ситуацій або узгодженості з доменними обмеженнями. Тому необхідним завершальним етапом конвеєра є валідація таблиці рішень, яка має на меті перевірити, що отримана операційна модель дійсно реалізує очікувану логіку рішення та не містить прихованих дефектів.

Якісно задачу валідації доцільно розглядати як перевірку семантичної коректності DMN-таблиці відносно трьох взаємопов'язаних джерел вимог. По-перше, таблиця повинна бути узгодженою з формальним бізнес-правилом, яке слугувало основою для її генерації. Це означає, що таблиця не повинна породжувати результатів, не передбачених абстрактною семантикою правила, і не повинна втрачати допустимі результати для жодного вхідного стану, на якому правило визначене. По-друге, таблиця повинна відповідати схемі даних системи підтримки прийняття рішень, зокрема використовувати лише визначені атрибути, коректно оперувати їхніми типами та доменами і не створювати внутрішніх типових суперечностей. По-третє, таблиця повинна бути узгодженою з контекстом предметної області, що включає доменні інваріанти, політики та обмеження, які не завжди явно представлені у формальній структурі правила.

З точки зору семантики програм, валідацію DMN-таблиці можна інтерпретувати як перевірку того, що операційна семантика, реалізована таблицею, є коректним семантичним уточненням абстрактної семантики бізнес-правила. Це передбачає аналіз поведінки таблиці на всьому допустимому просторі вхідних станів і виявлення ситуацій, у яких операційна модель поводиться неочікувано або неоднозначно. Зокрема, навіть за наявності семантичної еквівалентності на рівні окремих правил, таблична форма може містити конфлікти між рядками, перекриття умов або неохоплені комбінації вхідних значень, що призводять до непередбачуваної або некоректної поведінки під час виконання.

Суттєвим аспектом валідації є перевірка повноти таблиці рішень. У контексті систем підтримки прийняття рішень повнота означає, що для кожного допустимого вхідного стану, визначеного схемою даних і доменними обмеженнями, таблиця або породжує однозначний результат, або явно сигналізує про відсутність застосовного рішення. Неповні таблиці створюють ризик неявних помилок під час експлуатації системи, оскільки поведінка в непокритих випадках часто залежить від реалізації виконавчого рушія або зовнішньої логіки обробки помилок.

Іншим критично важливим аспектом є виявлення внутрішніх суперечностей і неоднозначностей. Таблиця рішень може містити кілька рядків, умови яких перекриваються, але результати є несумісними або обробляються некоректно з точки зору обраної політики виконання. Навіть якщо така таблиця формально виконується рушієм DMN, її семантика може бути неочевидною або суперечливою з точки зору бізнес-логіки. Тому якісна валідація повинна виявляти не лише явні синтаксичні помилки, а й глибші семантичні дефекти, пов'язані з конфліктами правил.

Отже, задачу валідації таблиць рішень DMN доцільно розглядати як завершальний етап семантичного аналізу, який перевіряє відповідність операційної моделі рішень абстрактній семантиці бізнес-правил, структурі даних системи та доменним обмеженням. Така постановка створює основу для подальшої математичної формалізації процедур валідації, зокрема для формального означення повноти, несуперечності та семантичної коректності DMN-таблиць рішень.

Нехай $\mathcal{S} = (\mathcal{A}, \mathcal{D}, \tau)$ – схема даних, $\mathcal{C}$ – доменний контекст, $T = \{t_i\}_{i=1}^{N}$ – множина текстових правил. За результатом етапу ідентифікації отримуємо формальну семантичну специфікацію рішення у вигляді множини правил $\mathcal{R} = \{r_i\}_{i=1}^{N}$, яка створює «еталонну» семантику $F_{\mathcal{R}} : \mathcal{X} \rightharpoonup \mathcal{Y}$ на допустимій області $\mathcal{X}_C \subseteq \mathcal{X}$. Далі генератор будує DMN-таблицю $\mathcal{D}$, яка задає операційну семантику $F_{\mathcal{D}} : \mathcal{X} \rightharpoonup \mathcal{Y}$.

Ключовий елемент підходу – незалежний генератор тестів, який використовує лише $(T, \mathcal{S}, \mathcal{C})$ або еквівалентно $(\mathcal{R}, \mathcal{S}, \mathcal{C})$, але не залежить від конкретної таблиці $\mathcal{D}$. Формально тест визначимо як пару $\xi = (x, \mathcal{O})$, де $x \in \mathcal{X}_C$ – вхідний стан, а $\mathcal{O} \subseteq \mathcal{Y}$ – множина припустимих очікуваних результатів. Використання

множини $\mathcal{O}$, а не одного значення, дозволяє коректно працювати з випадками недовизначеності або альтернатив у текстових правилах, а також із ситуаціями, коли бізнес-логіка визначає лише частину виходів. Тоді тестовий набір $\Xi = \{\xi_k\}_{k=1}^{m}$ є скінченною емпіричною специфікацією очікуваної поведінки рішення.

Генерацію тестів можна подати як стохастичний оператор

$$TestGen_{\eta} : (T, \mathcal{S}, \mathcal{C}) \Rightarrow \Xi,$$

параметризований η (наприклад, розмір набору, стратегія покриття, баланс типових і граничних випадків). При цьому *TestGen* можна інтерпретувати як вибір розподілу $Q\eta$ над $\mathcal{X}_C$ та побудову тестів шляхом семплінгу $x \sim Q$ і виведення очікуваних результатів $\mathcal{O}(x)$ з первинних артефактів правил і контексту. Узагальнено,

$$\xi = (x, \mathcal{O}(x)),\ x \sim Q_{\eta}, \mathcal{O}(x) = Oracle(T, \mathcal{S}, \mathcal{C}; x),$$

де *Oracle* – процедура, що синтезує очікування (через формальну семантику $\mathcal{R}$, через LLM+RAG-інтерпретацію, або їх комбінацію з консервативними правилами).

На основі запропонованих моделей ідентифікації бізнес-правил, генерації таблиць рішень та тестової валідації можна сформулювати інтегрований алгоритм, який реалізує повний конвеєр автоматизованої обробки текстових бізнес-правил у виконувані та перевірені DMN-таблиці (рис. 1). Вхідними даними алгоритму є множина текстових бізнес-правил T, схема даних системи підтримки прийняття рішень $\mathcal{S}$, доменний контекст $\mathcal{C}$, а також параметри керування процесом, зокрема пороги прийнятності тестової валідації та бюджети ітерацій. Вихідним результатом є або прийнята DMN-таблиця рішень, семантично узгоджена з бізнес-правилами, або множина контрприкладів, які вказують на неможливість досягнення прийнятної моделі за заданих обмежень.

Algorithm 1: End-to-end pipeline for DMN generation and validation

Input: Textual business rules T, data schema $\mathcal{S}$, domain context $\mathcal{C}$, thresholds (α, γ), budgets (B_T, B_D)
Output: Accepted DMN decision table $\mathcal{D}$ or counterexamples $\mathcal{F}$

```
R ← IdentifyRules(T, S, C);
g ← 0;
while g < B_D do
    D ← GenerateDMN(R, S, C);
    if not SynValid(D, S) then
        g ← g + 1;
        continue;
    Ξ ← TestGen(R, S, C);
    k ← 0;
    while k < B_T do
        F ← ExecuteTests(D, Ξ);
        if Accept(D, F, α, γ) then
            return D;
        if UpdateTests(F) then
            Ξ ← RefineTests(Ξ, F);
            k ← k + 1;
        else
            break;
    g ← g + 1;
return F;
```

Рис. 1. Алгоритм обробки бізнес-правил

Алгоритм починається з етапу ідентифікації бізнес-правил, на якому текстові правила інтерпретуються як неформальні специфікації семантики рішення. За допомогою конвеєра LLM+RAG виконується побудова множини формальних правил $\mathcal{R}$, узгоджених зі схемою даних і доменним контекстом. На цьому етапі формується абстрактна семантика рішення, яка надалі використовується як основне джерело очікуваної поведінки системи. На наступному етапі з абстрактної семантики $\mathcal{R}$ генерується кандидатна DMN-таблиця рішень $\mathcal{D}$. Генерація здійснюється як семантичне уточнення формальних правил і включає вибір вхідних і вихідних атрибутів, декомпозицію умов у множину рядків та фіксацію політики виконання. Згенерована таблиця підлягає синтаксичній перевірці відповідно до стандарту DMN. Таблиці, що не проходять цей етап, відхиляються без подальшого тестування, і процес повертається до повторної генерації.

У практичних сценаріях використання DMN-таблиць рішень їхня коректність рідко перевіряється шляхом повного формального аналізу семантики. Натомість, поширеним підходом є валідація через тестування, коли експерт предметної області або аналітик формує набір тестових випадків, що відображають типові, граничні та критичні ситуації, і перевіряє поведінку таблиці на цих прикладах. Такий підхід інтуїтивно зрозумілий, добре масштабується на складні бізнес-правила та дозволяє виявляти практично значущі дефекти, навіть якщо формальна модель є складною або неповною. Водночас ручне формування тестів є трудомістким і залежить від досвіду експерта, що обмежує відтворюваність і автоматизацію процесу.

У цій роботі валідація таблиць рішень пропонується як автоматизований процес тестування, у якому тестове покриття генерується безпосередньо з тих самих джерел знань, що й сама таблиця рішень, а саме з текстових бізнес-правил, схеми даних та контексту предметної області. Ключовою ідеєю є принципова незалежність тестового покриття від конкретної згенерованої DMN-таблиці. Тести не виводяться з табличної структури і не повторюють її логіку, а натомість репрезентують очікувану семантику рішення, відновлену з первинних артефактів бізнес-логіки. Завдяки цьому тестування стає інструментом виявлення семантичних дефектів генерації, а не формальним підтвердженням внутрішньої узгодженості таблиці.

Якісно кожен тестовий випадок можна інтерпретувати як конкретизацію очікуваної поведінки бізнес-правила для певного вхідного стану. Вхідні значення тесту формуються з урахуванням схеми даних, типів атрибутів і доменних обмежень, тоді як очікуваний результат або множина допустимих результатів визначається семантикою правила та контекстом предметної області. Таким чином, тестовий набір фактично реалізує емпіричну специфікацію семантики рішення,

проти якої перевіряється операційна модель у вигляді DMN-таблиці.

Процес валідації передбачає, що згенерована DMN-таблиця спочатку проходить синтаксичну перевірку та перевірку типів, які гарантують її формальну виконуваність у рушії DMN. Після цього таблиця піддається тестуванню на згенерованому наборі тестових випадків. Для кожного тесту обчислюється результат виконання таблиці, який порівнюється з очікуваним результатом, визначеним тестом. Таблиця вважається семантично коректною, якщо вона проходить тестування з достатнім рівнем покриття та не демонструє критичних розбіжностей з очікуваною поведінкою.

Оскільки тестове покриття є скінченним наближенням до повної семантичної перевірки, процес валідації організовується ітеративно та з урахуванням порогових критеріїв. Для тестування встановлюється обмеження на кількість тестових спроб або на рівень досягнутого покриття простору допустимих вхідних станів. Якщо згенерована таблиця не проходить тестування в межах заданого порогу, система може ініціювати повторну генерацію тестів з метою уточнення перевірки або повторну генерацію DMN-таблиці з урахуванням виявлених невідповідностей. Таким чином формується замкнений цикл уточнення, у якому тестування виступає інструментом зворотного зв'язку між очікуваною семантикою правила та її операційною реалізацією.

Такий підхід дозволяє розглядати валідацію DMN-таблиць як процес стохастичної семантичної перевірки, що поєднує формальні обмеження схеми даних і контексту з практикою тестування, прийнятою у прикладних системах. Він зберігає інтерпретованість ідеї валідації для експертів предметної області, водночас створюючи основу для автоматизації та формального аналізу якості згенерованих моделей рішень. Це, у свою чергу, відкриває можливість подальшої математичної формалізації процесу тестового покриття, критеріїв прийнятності та ітеративного вдосконалення DMN-таблиць.

Центральною передумовою запропонованого підходу до валідації є наявність незалежного джерела очікуваної семантики рішення, яке використовується для побудови тестових випадків і перевірки згенерованих DMN-таблиць. Це джерело формалізується у вигляді референтної семантичної моделі, яка для заданого вхідного стану визначає допустимі результати рішення на основі текстових бізнес-правил, схеми даних і доменного контексту. Принципово важливо, що ця референтна модель не використовує структуру DMN-таблиці та не аналізує її рядки, що забезпечує незалежність тестового покриття від результату генерації.

Референтна семантична модель розглядається як процедура, що інтерпретує ідентифіковану абстрактну семантику бізнес-правил у конкретних точках простору вхідних станів. Для кожного допустимого вхідного набору значень вона повертає множину допустимих результатів, що відображає можливу альтернативність у початкових текстових правилах. Така постановка дозволяє коректно працювати з правилами, які частково визначають поведінку системи, наприклад задають лише обмеження або політики, але не фіксують точного значення виходу.

Генератор тестових випадків використовує референтну семантичну модель як допоміжний компонент, який виконує окрему задачу побудови репрезентативного тестового покриття. З точки зору семантики, кожен тестовий випадок можна інтерпретувати як точкову специфікацію поведінки правила, тобто як перевірку того, що для конкретного вхідного стану операційна модель рішення не суперечить абстрактній семантиці, відновленій з тексту. Сукупність таких тестів утворює скінченне емпіричне наближення до повної семантичної перевірки.

Отже, після успішної синтаксичної перевірки запускається етап валідації. Незалежно від структури таблиці генерується тестове покриття $\mathcal{T}$ на основі $(\mathcal{R},\mathcal{S},\mathcal{C})$ із використанням семантичної моделі очікувань. Кожен тестовий випадок виконується на таблиці $\mathcal{D}$, а отримані результати порівнюються з допустимими очікуваннями. За результатами тестування обчислюються метрики точності та покриття, на основі яких приймається рішення про прийнятність таблиці. Якщо таблиця задовольняє порогові критерії, алгоритм завершується успішно, повертаючи $\mathcal{D}$ як валідну модель рішення. Якщо ж ні, то алгоритм переходить до ітеративного режиму уточнення. Залежно від характеру виявлених помилок ініціюється або регенерація тестового покриття, або повторна генерація DMN-таблиці. Цей процес повторюється до досягнення прийнятного результату або до вичерпання заданого бюджету ітерацій. Узагальнений псевдокод алгоритму наведено на рис.1.

Отже, запропонований алгоритм узагальнює всі розглянуті вище моделі в єдиний формальний процес. Він поєднує інтерпретацію неформальних бізнес-правил, семантично вмотивовану генерацію DMN-таблиць та практично орієнтовану валідацію. Така структура створює основу для подальшого експериментального аналізу ефективності та якості запропонованого методу.

Таким чином, розроблена формалізація визначає абстрактну модель процесу семантичного уточнення текстових бізнес-правил і множину припущень щодо взаємодії її компонентів. Для перевірки практичної застосовності цих припущень і тестування ефективності підходу в наступному розділі подано експериментальну оцінку на репрезентативних бізнес-сценаріях.

4. Експериментальне тестування

Метою експериментів є оцінка запропонованого підходу, а також емпіричне порівняння двох стратегій тестоорієнтованої валідації: з використання фіксованого набору тест-кейсів та з незалежною генерацією тест-кейсів і DMN-таблиць на кожній ітерації валідаційного циклу. Експерименти виконувалися з використанням прототипу програмної системи. Архітектура системи (рис. 2) відображає поділ процесу на етапи ідентифікації семантики, генерації DMN-таблиць і валідації, у яких текстові бізнес-правила, схема даних і доменний контекст використовуються як спільне семантичне джерело.

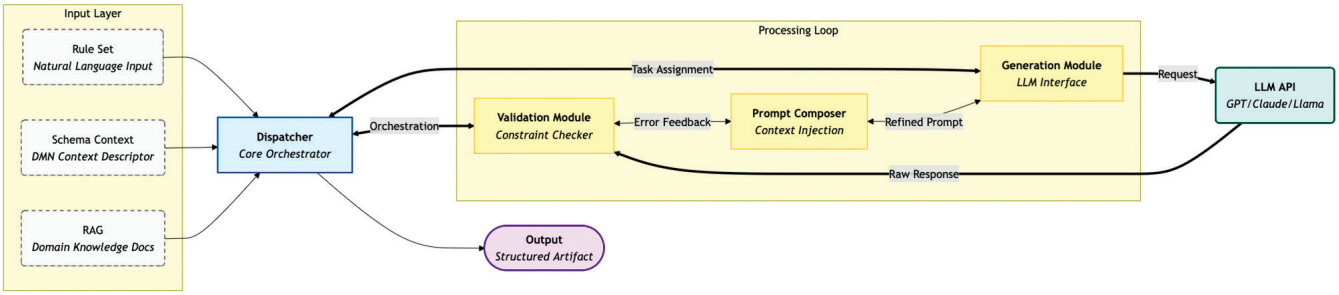

Рис. 2. Архітектура прототипу інтелектуального конвеєра обробки бізнес-правил

Прототип реалізовано з використанням LangChain для оркестрації мовних моделей і механізмів RAG та мовної моделі Claude Sonnet 4.5. За допомогою Prompt Composer формуються структуровані запити до мовної моделі. Перевірка виконується Validation Module, який послідовно здійснює синтаксичну, семантичну та логічну валідацію, включно з виконанням DMN-таблиць на згенерованих тест-кейсах.

Як репрезентативний сценарій використано набір бізнес-правил для автоматичного призначення продуктів іпотеки із політикою прийняття рішень FIRST. Правила визначають умови вибору між трьома можливими результатами: Fixed30, ARM7/ та ManualReview, на основі атрибутів заявника, зокрема кредитного рейтингу, співвідношення боргу до доходу (DTI), Loan-to-Value (LTV), стабільності доходу та поведінкових уподобань тощо. Для кожного експериментального прогону система генерувала до 200 тест-кейсів. Кожен тест-кейс містив вхідний об'єкт і множину очікуваних результатів, що дозволяло враховувати можливу недовизначеність семантики бізнес-правил.

Рисунок 3 ілюструє операційну інтерпретацію формальної моделі валідації DMN-таблиць, зокрема незалежну генерацію тестового покриття та моделей рішень. Було проведено два типи експериментів.

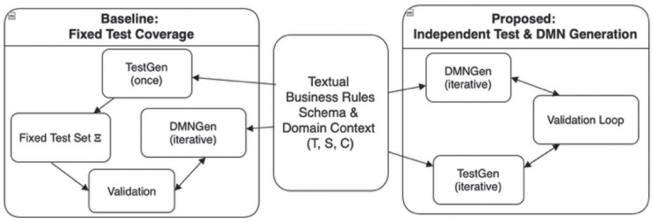

Рис. 3. Порівняння стратегій тестової валідації DMN-таблиць

У першому експерименті тест-кейси генерувалися один раз перед початком генерації DMN-таблиці та використовувалися як незмінний еталон для всіх подальших ітерацій валідації. Передбачалося, що такого набору буде достатньо для перевірки коректності багаторазово згенерованих DMN-таблиць. У 9 випадках система не змогла успішно пройти валідацію. Основні типи помилок включали некоректну множину можливих результатів (порожній список результатів) та семантичні невідповідності між очікуваним і фактичним значенням вихідної змінно product. Більшість успішних випадків завершувалися з першої спроби генерації DMN, однак наявність стабільних помилок вказала на обмеженість підходу з фіксованим тестовим покриттям .

У другому експерименті було застосовано альтернативну стратегію, за якої тест-кейси і DMN-таблиця генерувалися незалежно на кожній ітерації валідаційного циклу. Обидва артефакти формувалися з одного й того самого набору вхідних бізнес-правил, схемного та доменного контексту, але без взаємного використання структури. Результати показали, що за цієї стратегії всі 200 тестових прогонів завершилися успішно без жодного проваленого кейсу. Більшість DMN-таблиць проходили валідацію з першої або другої спроби, а максимальна кількість ітерацій не перевищувала трьох. Середній час обробки одного тестового кейсу склав 67 секунд, що є прийнятним для інтерактивного або напівавтоматизованого сценарію використання.

Порівняльний аналіз двох експериментів демонструє, що фіксований набір тест-кейсів не забезпечує достатнього покриття семантичного простору бізнес-правил. Натомість незалежна генерація тестів і таблиць створює ефект взаємної семантичної перевірки, за якого обидва артефакти повинні узгодитися з однією й тією ж абстрактною специфікацією. Отримані результати підтверджують, що тестове покриття повинно розглядатися як незалежне наближення

до референтної семантичної моделі рішення. Отже, доведена доцільність використання тестоорієнтованої валідації як ключового компонента автоматизованих систем формалізації бізнес-правил.

Висновки та обговорення

У цій роботі розглянуто задачу автоматизованої обробки текстових бізнес-правил як задачу відновлення формальної семантики рішень з неформального опису та її подальшого уточнення до виконуваних моделей у нотації DMN. На відміну від підходів, орієнтованих на безпосередню генерацію таблиць рішень, запропонований підхід розглядає ідентифікацію семантики, генерацію операційної моделі та її валідацію як єдиний узгоджений процес, формалізований у вигляді послідовних моделей і керованого ітеративного конвеєра.

Отримані результати підтверджують, що використання великих мовних моделей у поєднанні з RAG є ефективним засобом для інтерпретації текстових бізнес-правил і побудови початкових формальних представлень логіки рішень. Водночас експериментальна оцінка показала, що пряма генерація DMN-таблиць, навіть за умови врахування схемних обмежень, не гарантує семантичної коректності, повноти та несуперечності моделей рішень.

Ключовим результатом роботи є обґрунтування та експериментальне підтвердження ефективності тестоорієнтованої валідації, у межах якої тестове покриття генерується незалежно від DMN-таблиці, але з того самого семантичного джерела. Такий підхід дозволяє інтерпретувати тестовий набір як окрему проєкцію референтної семантичної моделі рішення і використовувати його для виявлення прихованих семантичних помилок, які не проявляються на рівні синтаксичної або структурної перевірки. Порівняння двох експериментальних стратегій показало, що незалежна генерація тестів і DMN-таблиць забезпечує стабільнішу конвергенцію та зменшує кількість некоректних моделей у фінальному результаті. Запропонований підхід розвиває і доповнює попередні результати з автоматизації подання логіки рішень, зокрема роботу [1], у якій було продемонстровано можливість використання штучного інтелекту для ефективної генерації DMN-артефактів.

Разом з тим робота має певні обмеження. По-перше, запропонований підхід покладається на якість текстових бізнес-правил і доменного контексту, що подаються на вхід системи. Неоднозначні або суперечливі формулювання можуть призводити до розширення множини допустимих результатів у референтній семантичній моделі. По-друге, використання великих мовних моделей зумовлює стохастичність процесу генерації та залежність від обчислювальних ресурсів, що може обмежувати застосування підходу в сценаріях із жорсткими вимогами до часу виконання.

Подальші дослідження доцільно спрямувати на формалізацію критеріїв зупинки ітеративного процесу валідації, автоматичну адаптацію бюджетів генерації та розширення підходу на складніші типи DMN-моделей, зокрема з багаторівневими діаграмами вимог до рішень і складними політиками агрегації результатів. Окремим перспективним напрямом є інтеграція формальних методів верифікації з тестоорієнтованою валідацією для подальшого підвищення гарантій семантичної коректності.

Список літератури:

[1] Cherednichenko O. AI-Based Efficient Automation of Decision Logic Representation / O. Cherednichenko, V. Maliarenko // Proc. of International Conference on Business Information Systems. – 2026. – P. 303–315. – doi: 10.1007/978-3-032-08614-3_19.

[2] Object Management Group. Decision Model and Notation (DMN), Version 1.6 // OMG Specification. – 2023.

[3] Batoulis K. Decision modeling and business process management / K. Batoulis, M. Weske // Business Process Management Journal. – 2018. – Vol. 24(2). – P. 451–470.

[4] Etikala S. Text2Dec: Extracting decision logic from natural language / S. Etikala, S. Goossens, J. De Smedt // Proc. of International Conference on Business Process Management (BPM). – 2020. – P. 303–319.

[5] Goossens S. Automatic extraction of decision models from textual descriptions / S. Goossens, J. De Smedt, J. Vanthienen // Decision Support Systems. – 2019. – Vol. 123. – P. 113–129.

[6] Goossens S. Evaluating large language models for DMN decision table generation / S. Goossens, J. De Smedt // Proc. of IEEE International Conference on Business Informatics. – 2023. – P. 1–8.

[7] Berkovitch S. Benchmarking large language models for structured table generation / S. Berkovitch, M. van der Aalst // Proc. of CAiSE. – 2023. – P. 145–160.

[8] Shorten R. Structured generation with schema constraints / R. Shorten, J. Kelleher // Proc. of EMNLP. – 2022. – P. 5621–5633.

[9] Li J. Retrieval-augmented generation: A survey / J. Li, Y. Liu, Z. Li // ACM Computing Surveys. – 2023. – Vol. 56(4). – P. 1–38.

[10] Jeong S. Schema-aware retrieval for enterprise decision systems / S. Jeong, H. Kim // Information Systems. – 2022. – Vol. 107. – P. 101–119.

[11] Liu P. On the challenges of contextual relevance in retrieval-augmented generation / P. Liu, X. Zhang // Proc. of ACL. – 2023. – P. 412–423.

[12] Calvanese D. Detecting incompleteness and inconsistency in DMN decision tables / D. Calvanese, M. Montali // Proc. of KR. – 2018. – P. 318–327.

[13] Corea C. Decision logic verification with Camunda DMN / C. Corea, M. Dumas // Software and Systems Modeling. – 2021. – Vol. 20(3) . – P. 965–985.

Надійшла до редколегії 12.11.2025