УДК / UDC: 004.89:351.86
DOI: https://doi.org/10.37750/2616-6798.2026.3(58).375616

Дмитро Володимирович Ланде
Державна наукова установа "Інститут інформації, безпеки і права Національної академії правових наук України"; КПІ ім. Ігоря Сікорського
Київ, Україна
ORCID: https://orcid.org/0000-0003-3945-1178
Володимир Миколайович Фурашев
Державна наукова установа "Інститут інформації, безпеки і права Національної академії правових наук України"
Київ, Україна
ORCID: https://orcid.org/0000-0001-7205-724X
Леонард Страшной
Університет Каліфорнії (UCLA), Лос-Анджелес, США
ORCID: https://orcid.org/0009-0008-5575-0286

ФРЕЙМВОРК AGENTFLOWLAW ДЛЯ БЕЗКОДОВОГО ПРОГРАМУВАННЯ ПРОЦЕДУР АНАЛІЗУ ПРАВОВИХ АКТІВ

Анотація. *У статті запропоновано AgentFlowLaw – спеціалізований фреймворк для безкодового програмування процедур аналізу правових актів із використанням великих мовних моделей і мультиагентного підходу. Фреймворк розроблено як предметно орієнтоване розширення AgentFlow, у якому збережено базові логічні примітиви та модель взаємодії агентів, але введено компоненти, що враховують специфіку правової інформації: правовий контекст, юрисдикцію, часову чинність норм, нормативний статус і перевірку джерел, доказову базу, невизначеність, простежуваність та контроль людини. Запропоновано підхід, за якого велика мовна модель не лише формує відповідь на правове питання, а генерує та виконує структуровану агентську процедуру його розв'язання. Така процедура передбачає декомпозицію задачі, динамічне формування спеціалізованих агентів, послідовне та паралельне виконання операцій, обмін результатами, їх незалежну перевірку та повторний аналіз за необхідності. Для суттєвих правових висновків забезпечується простежуваний ланцюжок «висновок – аргументація – правова норма – джерело». Практичну придатність AgentFlowLaw продемонстровано на прикладі аналізу проєкту Акта про штучний інтелект Європейського Союзу. У межах експерименту агентська система виявляла правові ризики, потенційні суперечності, неоднозначності та прогалини, індикатори корупційних ризиків, а також передбачені правовим актом запобіжні механізми. Показано, що поєднання великих мовних моделей, мультиагентної взаємодії та безкодового програмування дає змогу формалізувати професійні процедури правового аналізу, зробити їх відтворюваними, перевірюваними та доступними для використання фахівцями без необхідності традиційного програмування. Остаточна правова оцінка та прийняття юридично значущих рішень залишаються за людиною.*

Ключові слова: *AgentFlowLaw, AgentFlow, великі мовні моделі, мультиагентні системи, безкодове програмування, правові акти, правовий аналіз, правові ризики, верифікація, простежуваність, штучний інтелект.*

© Ланде Д.В., Фурашев В.М., Страшной Л. 2026

Dmytro V. Lande
State Scientific Institution "Institute of Information, Security and Law of the National Academy of Legal Sciences of Ukraine"; National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
Kyiv, Ukraine
ORCID: https://orcid.org/0000-0003-3945-1178
Volodymyr M. Furashev
State Scientific Institution "Institute of Information, Security and Law of the National Academy of Legal Sciences of Ukraine"
Kyiv, Ukraine
ORCID: https://orcid.org/0000-0001-7205-724X
Leonard Strashnoi
Senior Data Architect at the University of California (UCLA)
Los Angeles, USA
ORCID: https://orcid.org/0009-0008-5575-0286

AGENTFLOWLAW FRAMEWORK FOR CODELESS PROGRAMMING OF LEGAL ACT ANALYSIS PROCEDURES

Summary. *The article offers AgentFlowLaw – a specialized framework for code-free programming of legal act analysis procedures using large language models and a multi-agent approach. The framework was developed as a subject-oriented extension of AgentFlow, which preserves the basic logical primitives and model of agent interaction, but introduces components that take into account the specifics of legal information: legal context, jurisdiction, temporal validity of norms, regulatory status and verification of sources, evidence base, uncertainty, traceability and human control. An approach is proposed in which a large language model not only forms the answer to a legal question, but generates and executes a structured agency procedure for its solution. Such a procedure involves decomposition of the task, dynamic formation of specialized agents, sequential and parallel execution of operations, exchange of results, their independent verification and re-analysis as necessary. For significant legal conclusions, a traceable chain “conclusion – argumentation – legal norm – source” is provided. The practical suitability of AgentFlowLaw is demonstrated on the example of the analysis of the project of the Act on Artificial Intelligence of the European Union. Within the framework of the experiment, the agency system identified legal risks, potential contradictions, ambiguities and gaps, indicators of corruption risks, as well as preventive mechanisms provided for by the legal act. It is shown that the combination of large language models, multi-agent interaction and code-free programming makes it possible to formalize professional legal analysis procedures, make them reproducible, verifiable and available for use by specialists without the need for traditional programming. The final legal assessment and adoption of legally significant decisions remain with the person.*

Keywords: *AgentFlowLaw, AgentFlow, large language models, multi-agent systems, codeless programming, legal acts, legal analysis, legal risks, verification, traceability, artificial intelligence.*

Вступ

Актуальність теми. Стрімке поширення великих мовних моделей і агентських систем створює нові можливості для автоматизації роботи з правовими актами. Сучасні мовні моделі здатні опрацьовувати значні за обсягом тексти, виділяти нормативні положення, встановлювати змістові зв’язки, зіставляти формулювання та формувати аргументовані висновки. Водночас безпосереднє застосування великої мовної моделі за

схемою “правове питання – відповідь” є недостатнім для задач, у яких суттєвими є обґрунтованість результату, перевірка джерел, урахування юрисдикції та чинності правових норм, виявлення суперечностей і збереження альтернативних інтерпретацій.

Правовий аналіз за своєю природою є багатоетапною процедурою. Він передбачає структурне опрацювання документа, виділення правових норм, пошук пов’язаних нормативних положень, їх зіставлення, виявлення ризиків, колізій, неоднозначностей і прогалин, перевірку аргументації та формування підсумкового висновку. Тому перспективним є перехід від використання однієї мовної моделі до мультиагентної організації аналізу, за якої окремі функції виконуються спеціалізованими агентами, а їх результати передаються, зіставляються та незалежно перевіряються.

Разом із тим практичне застосування таких систем у правовій сфері не повинно вимагати від юриста володіння традиційними мовами програмування. Це зумовлює актуальність безкодового програмування, за якого професійна методика правового аналізу описується природною мовою у структурованій формі, а велика мовна модель виконує функції інтерпретатора сформованої процедури. Такий підхід дозволяє перенести акцент із програмування програмного коду на формалізацію логіки професійної діяльності.

Отже, актуальною є задача створення предметно орієнтованого фреймворку, який поєднує можливості великих мовних моделей, мультиагентних систем і безкодового програмування та враховує специфічні вимоги правового аналізу. Запропонований у роботі AgentFlowLaw спрямований на формування саме таких процедур – від постановки правової задачі та побудови системи спеціалізованих агентів до перевірки, простежування й представлення результатів із збереженням остаточного юридично значущого рішення за людиною.

Огляд літератури та стан дослідження. Застосування великих мовних моделей у правовій сфері є напрямом, що активно розвивається. У наукових роботах розглядаються задачі пошуку та узагальнення правової інформації, аналізу нормативних і судових документів, відповідей на правові запити, підготовки документів, виявлення змістових зв’язків між нормами та підтримки прийняття рішень [1]. Водночас дослідники відзначають низку проблем, пов’язаних із достовірністю сформованих відповідей, можливістю появи необґрунтованих тверджень, залежністю результату від формулювання запиту, складністю перевірки аргументації, урахуванням юрисдикції та необхідністю роботи з актуальними правовими джерелами [2, 3].

Окремим напрямом стало використання агентного та мультиагентного підходів, за яких складна правова задача поділяється на сукупність взаємопов’язаних підзадач [4]. Спеціалізовані агенти можуть виконувати пошук інформації, аналіз правових положень, формування аргументів, перевірку результатів та їх узагальнення [5]. Перехід від одиничного агента до мультиагентних архітектур розглядається як перспективний для складних правових задач, хоча він одночасно породжує проблеми координації агентів, узгодження результатів, контролю проміжних висновків та збільшення складності системи [6].

Важливим для цієї роботи є також розвиток засобів безкодового програмування, що дозволяють задавати логіку розв’язання задач без написання традиційного програмного коду [7]. У попередніх роботах авторів було запропоновано базовий фреймворк AgentFlow, у якому програма задається структурованим описом природною мовою, а велика мовна модель використовується для його інтерпретації та виконання [8]. Для формування складної поведінки агентів було введено набір логічних примітивів, модель агента та механізми організації взаємодії агентів. У подальших авторських дослідженнях

розвивалися питання оркестрації безкодових агентів, їх паралельного виконання та використання в різних середовищах [9].

Можливості запропонованого підходу були також досліджені авторами стосовно задач правової аналітики та парламентського контролю. Розглядалися процедури автоматизованого опрацювання законопроєктів, виявлення ризиків і суперечностей, аналізу потенційних корупційних загроз, а також застосування великих мовних моделей як інструменту реалізації складних аналітичних процедур. Відповідні результати створили основу для переходу від застосування універсального AgentFlow до окремих правових задач до побудови спеціалізованого фреймворку, в якому властивості правової інформації враховуються безпосередньо на рівні опису агентської процедури [10, 11].

Проведений аналіз стану досліджень показує, що наявні підходи охоплюють використання великих мовних моделей у праві, спеціалізованих правових агентів, мультиагентної взаємодії та багатоетапних процедур правового аналізу [12]. Зокрема, розробляються системи, в яких декілька агентів спільно розв’язують правові задачі, а також засоби оцінювання здатності агентів використовувати зовнішні знання та виконувати складні послідовності операцій [13, 14]. Проте залишається актуальною задача створення засобів, у яких сама процедура правового аналізу може бути сформована фахівцем без традиційного програмування, а специфічні властивості правових норм і джерел є явними складовими такої процедури [15].

Саме таким розвитком базового підходу є запропонований у цій роботі AgentFlowLaw. Його відмінність полягає не лише у використанні AgentFlow для роботи з юридичними текстами, а у предметній спеціалізації механізму безкодового програмування. До структури процедури вводяться правовий контекст, юрисдикція, чинність правових норм, нормативний статус джерел, доказова база, перевірка, робота з невизначеністю та простежуваність результату [16]. Таким чином, AgentFlowLaw програмує не наперед задану відповідь на правове питання, а процедуру її отримання, перевірки, простеження та представлення.

Розроблений AgentFlowLaw представлено у відкритому репозиторії GitHub [17], що містить опис фреймворку та матеріали, необхідні для його практичного застосування і відтворення наведених у роботі процедур. Посилання на репозиторій і відповідний бібліографічний опис буде додано до остаточної редакції статті. Такий спосіб представлення дозволяє розділити теоретичний опис підходу, наведений у статті, та його доступну для практичного використання реалізацію.

Отже, ця робота продовжує авторську лінію досліджень “AgentFlow → застосування AgentFlow у правовій аналітиці → AgentFlowLaw”, але переходить від використання універсального фреймворку в окремій предметній області до створення спеціалізованого засобу безкодового програмування процедур аналізу правових актів. Саме цей перехід визначає місце роботи серед наявних досліджень і формує основу для подальшого викладу архітектури та експериментальної перевірки AgentFlowLaw.

Метою статті є розроблення спеціалізованого фреймворку AgentFlowLaw для безкодового програмування процедур аналізу правових актів із використанням великих мовних моделей і мультиагентного підходу, що забезпечує формалізацію правової задачі, автоматичне формування та координацію спеціалізованих агентів, перевірку правових джерел і отриманих результатів, урахування невизначеності та формування простежуваних висновків для подальшого експертного оцінювання людиною.

Для досягнення цієї мети поставлено такі завдання: формалізувати принципи безкодового програмування процедур правового аналізу на основі базової концепції AgentFlow; розробити архітектуру AgentFlowLaw з урахуванням правового контексту,

юрисдикції, чинності та нормативного статусу правових джерел, доказової бази, невизначеності й простежуваності результатів; визначити модель спеціалізованого правового агента та принципи динамічного формування мультиагентної системи; розробити механізми взаємодії агентів і перевірки результатів правового аналізу; реалізувати AgentFlowLaw як метапромпт для генерації структурованих безкодових агентських процедур за описом правової задачі природною мовою; експериментально перевірити запропонований підхід на задачі аналізу правового акта; визначити особливості AgentFlowLaw порівняно з базовим AgentFlow та оцінити можливості поєднання великих мовних моделей, мультиагентного підходу і безкодового програмування у правовій аналітиці.

Виклад основного матеріалу:

В основу запропонованого підходу покладено ідею безкодового програмування, за якої складна задача розв'язується не шляхом безпосереднього написання програмного коду, а шляхом формування структурованого природномовного опису процедури її вирішення. Велика мовна модель у такому випадку використовується не лише як засіб генерації відповіді на запит користувача, а як середовище для створення та виконання агентських промптів. Користувач формулює задачу мовою предметної області, після чого на її основі створюється система спеціалізованих агентів, визначаються їх функції, порядок взаємодії, умови переходів між етапами, способи перевірки проміжних результатів та структура кінцевої відповіді.

Таким чином, принциповим є перехід від традиційної схеми

запит→велика мовна модель→відповідь

до схеми

задача→генерація агентської процедури→

→виконання процедури→перевірка→результат.

За такого підходу предметом програмування фактично стає не програмний код, а логіка розв'язання задачі. Її можна описати через ролі агентів, їх стани, функції, умови, цикли, переходи, інформаційні зв'язки та правила взаємодії. Це дає можливість користувачеві працювати на рівні професійних понять і процедур, не переводячи кожну предметну операцію в конструкції традиційної мови програмування.

Від універсального AgentFlow до предметно орієнтованого фреймворку

Реалізацією зазначеного підходу в попередніх роботах авторів став універсальний фреймворк AgentFlow. Його основною ідеєю є представлення складної агентської процедури у вигляді структурованого природномовного псевдокоду, який може інтерпретуватися великою мовною моделлю. AgentFlow не орієнтований на конкретну предметну область і задає загальні засоби формування та координації агентів.

Базова модель агента в AgentFlow може бути подана як

$$a=\langle R,S,L,C\rangle,$$

де R – роль агента, S – його стан, L – логіка поведінки, C – механізм комунікації з іншими агентами. Така модель дозволяє відокремити функціональне призначення агента від конкретної реалізації та будувати з окремих агентів складні процедури вирішення задач.

Для опису логіки AgentFlow використовує обмежений набір універсальних конструкцій, зокрема умову, цикл, функцію, мітку та перехід (Condition, Loop, Function, Label, Goto). На їх основі можуть задаватися послідовне виконання операцій, розгалуження залежно від отриманого результату, повернення до попереднього етапу, повторне виконання процедури, а також виклик окремих функціональних блоків. Важливо, що ці конструкції задаються у структурованій природномовній формі й тому не вимагають від користувача створення програмного коду традиційними засобами.

Іншою можливістю AgentFlow є формування не одного агента, а їх сукупності для розв'язання складної задачі. Загальна задача декомпозується на функціонально завершені підзадачі, для кожної з яких може створюватися окремий агент. Частина агентів виконується послідовно, коли результат попереднього етапу необхідний для наступного, а незалежні операції можуть виконуватися паралельно. Після цього їх результати об'єднуються та передаються на наступні етапи. Отже, AgentFlow дає змогу програмувати не лише поведінку окремого агента, а й структуру взаємодії мультиагентної системи.

Універсальність AgentFlow водночас визначає його обмеження при безпосередньому використанні у спеціалізованих предметних областях. Фреймворк визначає, як сформувати агентів, організувати їх взаємодію, реалізувати умови, цикли та переходи, але сам по собі не задає, які саме предметні характеристики повинні обов'язково враховуватися під час виконання конкретної професійної процедури. Для правової сфери це має особливе значення.

Необхідність спеціалізації для правового аналізу

Робота з правовими актами не може бути зведена лише до семантичного аналізу тексту. Правове положення має розглядатися в контексті юрисдикції, виду нормативного документа, його місця в системі правових джерел, часової чинності та співвідношення з іншими нормами. Навіть семантично коректно сформований великою мовною моделлю висновок не може вважатися достатньо обґрунтованим, якщо неможливо встановити правову норму, на якій він ґрунтується, та відповідне джерело.

Крім того, для правового аналізу суттєвим є розмежування різних типів результатів. Виявлений ризик не обов'язково означає порушення; потенційна суперечність ще не є встановленою правовою колізією; наявність дискреційних повноважень сама по собі не означає наявності корупції. Аналогічно, відсутність достатніх даних для підтвердження певного висновку не повинна компенсуватися генерацією моделлю найбільш правдоподібного твердження. Невизначеність у такому випадку є самостійним результатом аналізу, який необхідно зберегти та передати фахівцеві.

Ці особливості зумовили необхідність переходу від універсального AgentFlow до його предметно орієнтованого розвитку – AgentFlowLaw. Його призначенням є безкодове програмування не довільних агентських процедур, а саме процедур аналізу правових актів.

У формалізованому вигляді процедуру, сформовану AgentFlowLaw, можна подати як

$$P^{Law}=FAFL(T,C,S,R,O),$$

де T – поставлена правова задача; C – правовий контекст; S – сукупність доступних правових джерел; R – вимоги та обмеження процедури; O – вимоги до структури результату.

Таким чином, AgentFlowLaw виконує функцію метапромпту: на вході отримує опис правової задачі, а результатом його роботи стає структурована агентська процедура, придатна для подальшого виконання великою мовною моделлю. Отже,

Правова задача→AgentFlowLaw→агентська процедура→ВММ→результат ана лізу.

Ця схема відображає основну відмінність запропонованого підходу: AgentFlowLaw програмує не відповідь на правове питання, а процедуру отримання, перевірки, простеження та представлення цієї відповіді.

Нові елементи AgentFlowLaw

AgentFlowLaw успадковує базові механізми AgentFlow, але доповнює їх предметними елементами, необхідними для правового аналізу. До них належать правовий контекст, правове джерело, юрисдикція, чинність, нормативний статус, доказова база, простежуваність, невизначеність, перевірка та контроль людини. Тому співвідношення між двома фреймворками можна узагальнено подати як

AgentFlowLaw =

= AgentFlow+правовий контекст+правові джерела+правова перевірка+

+простежуваність+контроль людини.

Розширюється також модель агента. Якщо в AgentFlow агент визначається роллю, станом, логікою та комунікацією, то для правової предметної області до його характеристик додаються параметри, пов’язані з юрисдикцією, нормативним статусом, доказовою базою, чинністю та простежуваністю. Відповідно, спеціалізований агент може бути представлений як

$$a^{Law}=\langle R,S,L,C,J,A,E,V,T\rangle,$$

де J характеризує юрисдикцію, A – нормативний статус і авторитетність джерела, E – доказову базу, V – часову чинність, T – простежуваність отриманого результату.

Склад агентів при цьому не задається раз і назавжди. Він формується відповідно до конкретної правової задачі. Наприклад, можуть створюватися агенти структурного аналізу документа, виділення правових норм, пошуку пов’язаних джерел, аналізу правових ризиків, виявлення суперечностей, неоднозначностей і прогалин, пошуку індикаторів корупційних ризиків, перевірки доказової бази, формування правової аргументації та інтеграції результатів. Реалізується принцип “один агент – одна основна аналітична функція”, що полегшує контроль проміжних результатів і визначення джерела можливих помилок.

Важливою властивістю AgentFlowLaw є можливість поєднання послідовного та паралельного аналізу. Операції, що використовують спільний початковий правовий контекст, але вирішують різні задачі, можуть виконуватися незалежно. Наприклад, аналіз правових ризиків, суперечностей, неоднозначностей і корупційних ризиків доцільно здійснювати паралельно. Отримані результати після цього об’єднуються, однак сам факт об’єднання не означає автоматичного усунення розбіжностей. Якщо спеціалізовані агенти отримали різні інтерпретації, такі розбіжності можуть становити значущий результат і мають бути збережені для подальшої перевірки.

Перевірка, невизначеність і простежуваність

Одним із центральних елементів AgentFlowLaw є відокремлення первинного аналізу від перевірки його результатів. Для цього формується окремий контур

Аналіз→перевірка джерел→перевірка чинності→

→перевірка суперечностей→перевірка аргументації.

Якщо на одному з етапів виявлено недостатність доказів, суперечність або неможливість підтвердження правового положення, результат може бути повернуто на повторний аналіз. При цьому кількість повторень має бути скінченною, а невирішена проблема після завершення циклу не приховується, а включається до кінцевого результату із зазначенням відповідного рівня невизначеності.

Для правових джерел передбачається фіксація стану їх перевірки: підтверджене джерело, неперевірене джерело, суперечливе джерело або джерело, яке не вдалося знайти. Аналогічно для виявлених суперечностей можуть розрізнятися підтверджений, можливий, непідтверджений та невирішений стани. Такий механізм перешкоджає перетворенню припущення, сформованого великою мовною моделлю, на категоричний правовий висновок.

Особливе значення має простежуваність. Для кожного суттєвого результату має бути можливо відновити логіку його формування:

висновок→аргументація→правова норма→джерело.

Отже, кінцевий результат AgentFlowLaw є не лише текстом, сформованим великою мовною моделлю, а структурованою сукупністю висновків, аргументів, норм, джерел, статусів перевірки та зафіксованих невизначеностей.

Організація взаємодії агентів

У типовій процедурі AgentFlowLaw первинний документ передається агенту завантаження та підготовки, після чого виконується його структурний аналіз і виділення нормативно значущих положень. Далі здійснюється пошук і перевірка пов’язаних правових норм. Після формування спільного правового контексту можуть паралельно працювати спеціалізовані аналітичні агенти. Їх результати надходять до агента перевірки доказової бази, агента правової аргументації та, на завершальному етапі, до агента узагальнення результатів.

Відповідно до сформованої процедури AgentFlowLaw аналіз правового акта виконується не однією великою мовною моделлю в межах єдиного запиту, а системою спеціалізованих агентів із розподіленими функціями. Після завантаження та структурного опрацювання документа здійснюється пошук релевантних правових норм, а основні напрями аналізу – виявлення правових ризиків, суперечностей, неоднозначностей і прогалин, а також індикаторів корупційних ризиків – виконуються паралельно. Отримані результати проходять окрему перевірку доказової бази, після чого формується правова аргументація та узагальнений результат для подальшого оцінювання фахівцем. Схему взаємодії агентів наведено на рис. 1.

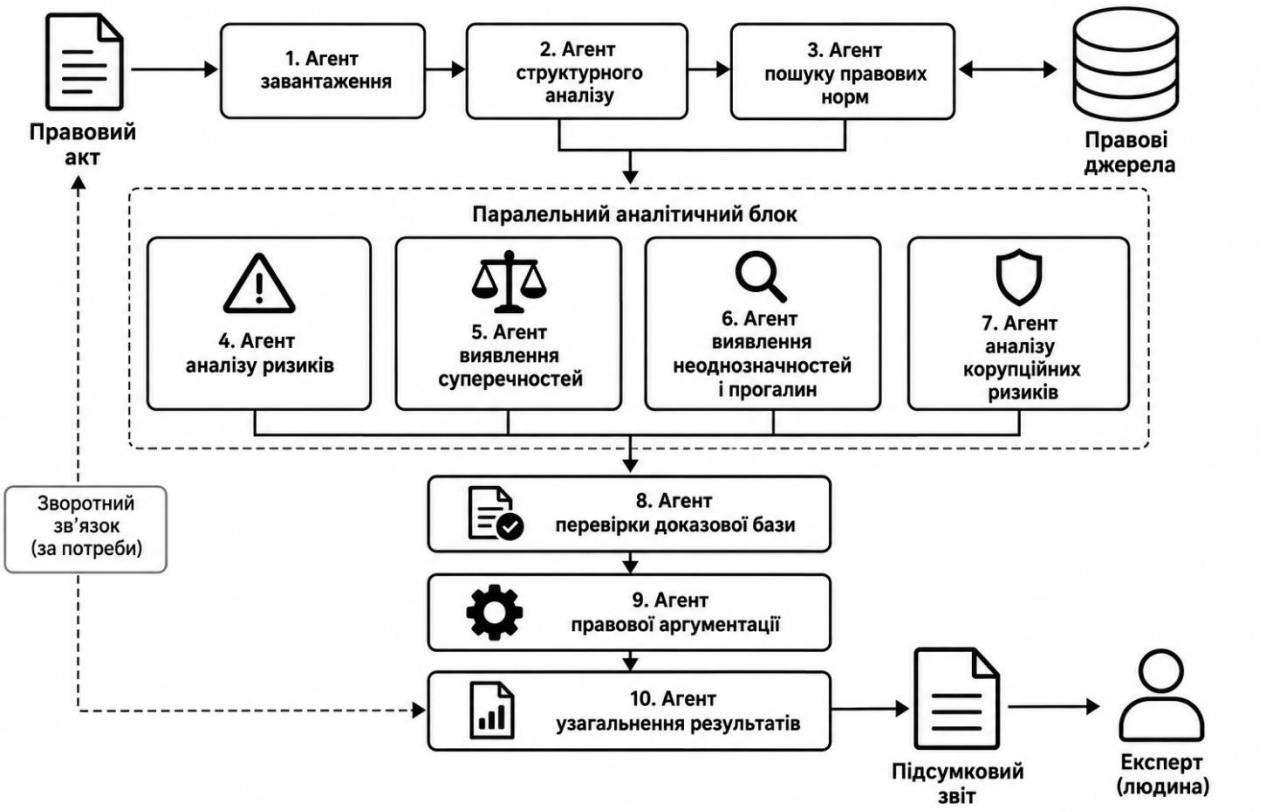


Рис. 1. Схема взаємодії агентів у фреймворку AgentFlowLaw

Як видно з рис. 1, архітектура AgentFlowLaw поєднує послідовні та паралельні процедури. Паралельне виконання спеціалізованих аналітичних функцій дає змогу незалежно виявляти різні класи проблем правового акта, не змішуючи їх на початковому етапі аналізу. Окремий агент перевірки доказової бази контролює обґрунтованість виявлених проблем і їх зв'язок із правовими джерелами, а агент правової аргументації формує простежуваний ланцюжок "висновок – аргументація – норма – джерело". Передбачений зворотний зв'язок дає змогу повторити окремі етапи у разі недостатності або суперечливості отриманих результатів. Підсумковий звіт не розглядається як остаточне юридичне рішення: він передається експерту для оцінювання та прийняття рішення.

Людина в контурі правового аналізу

Принциповою вимогою AgentFlowLaw є збереження людини в контурі прийняття рішення. Агентська система може знаходити норми, встановлювати зв'язки між ними, виявляти ознаки суперечностей, формувати альтернативні інтерпретації, перевіряти джерела та будувати аргументацію. Однак результат такого аналізу має характер інформаційно-аналітичної підтримки.

Це особливо важливо для категорій, які потребують правової кваліфікації. Наприклад, агент, що аналізує корупційні ризики, не повинен встановлювати факт корупції або приписувати суб'єктам неправомірні наміри. Його функцією є виявлення структурних і процедурних ознак, які можуть створювати умови для надмірної дискреції, непрозорості, вибіркового застосування норм або недостатнього контролю. Остаточна правова оцінка таких ознак належить фахівцеві.

У цьому сенсі загальну логіку AgentFlowLaw можна подати як замкнений контур:

правова задача→агентська процедура→аналіз→перевірка→

→простежуваний результат→експерт

Запропонована організація дозволяє поєднати переваги великих мовних моделей – роботу з природною мовою, контекстне опрацювання великих текстів і гнучке формування процедур – із функціональною декомпозицією мультиагентних систем та контролем, необхідним для правової предметної області. Безкодове представлення процедури, своєю чергою, робить її доступною для формування та модифікації безпосередньо фахівцем предметної області.

Таким чином, AgentFlowLaw є не набором заздалегідь визначених промптів для окремих юридичних операцій, а фреймворком генерації безкодових агентських процедур відповідно до поставленої правової задачі. Саме ця властивість створює основу для його застосування до різних типів правових актів і різних видів аналізу без необхідності щоразу програмувати нову мультиагентну систему традиційними засобами.

Для перевірки працездатності запропонованого підходу далі проведено експеримент, у якому AgentFlowLaw застосовано до конкретної задачі аналізу правового акта. Метою експерименту було перевірити, чи може сформована за допомогою фреймворку агентська процедура забезпечити декомпозицію поставленої задачі, організувати взаємодію спеціалізованих агентів, виявити різні класи правових проблем, перевірити отримані результати та сформувати простежуваний підсумковий звт для експерта.

Експериментальне застосування AgentFlowLaw для аналізу проєкту Акта про штучний інтелект

Для перевірки практичної придатності запропонованого фреймворку AgentFlowLaw проведено експеримент з автоматизованого аналізу проєкту нормативно-правового акта. Як досліджуваний документ обрано проєкт Акта про штучний інтелект Європейського Союзу, представлений Європейською Комісією 21 квітня 2021 р. як пропозицію Регламенту Європейського Парламенту та Ради щодо встановлення гармонізованих правил у сфері штучного інтелекту – COM(2021) 206 final, 2021/0106(COD).

Вибір цього документа зумовлений його складною структурою, значним обсягом, наявністю різних категорій правових норм, системи винятків, механізмів контролю та оцінювання ризиків. Сам проєкт побудований відповідно до ризик-орієнтованого підходу та передбачає диференціацію систем штучного інтелекту залежно від рівня потенційного ризику. Для систем високого ризику встановлюються спеціальні вимоги щодо даних, документації, простежуваності, прозорості, людського контролю, точності та стійкості.

Первинне завдання для AgentFlowLaw було сформульовано максимально стисло:

Автоматичне виявлення ризиків, суперечностей і корупційних загроз у тексті законопроєкту.

Відповідно до фреймворку AgentFlowLaw на основі цього завдання було сформовано безкодову агентську процедуру, що включала десять спеціалізованих агентів: завантаження та перевірки документа; структурного аналізу; пошуку правових норм; аналізу правових ризиків; виявлення суперечностей; виявлення неоднозначностей і прогалин; виявлення індикаторів корупційних ризиків; перевірки доказової бази; формування правової аргументації; узагальнення результатів.

Процедура передбачала послідовне виділення нормативних положень документа, паралельний аналіз різних типів ризиків, перевірку отриманих результатів і формування підсумкового висновку. Для кожного суттєвого результату мала підтримуватися простежуваність:

Висновок→Аргументація→Норма→Джерело.

Принципово важливо, що агент, який виявляв потенційні корупційні ризики, не мав права кваліфікувати певне положення як прояв корупції. Його функція полягала у виявленні нормативних конструкцій, здатних створювати умови для надмірної дискреції, непрозорості, вибіркового застосування норм або недостатнього контролю. Остаточна правова оцінка залишалася за людиною.

Результати аналізу

На першому етапі було встановлено загальну нормативну структуру документа. Проєкт передбачає заборону окремих практик застосування штучного інтелекту, спеціальний режим для систем високого ризику, вимоги прозорості для певних інших систем, механізми оцінювання відповідності, нагляду та застосування санкцій. Задекларованими цілями регулювання є безпечність систем штучного інтелекту та дотримання основоположних прав, забезпечення правової визначеності, посилення нагляду і формування єдиного ринку відповідних технологій.

Однією з перших виявлених проблем стало використання оціночних правових категорій, від тлумачення яких залежить правовий режим системи штучного інтелекту. Зокрема, це стосується визначення високого ризику, суттєвості ризику, достатньої прозорості, належного рівня точності, необхідності та пропорційності певних заходів. Характерно, що ще під час підготовки проєкту учасники консультацій звертали увагу на необхідність чіткішого визначення понять “штучний інтелект”, “ризик”, “високий ризик”, “віддалена біометрична ідентифікація” та “шкода”.

Другий виявлений ризик пов’язаний із динамічністю класифікації систем високого ризику. Проєкт передбачає можливість розширення переліку таких систем відповідно до розвитку технологій та появи нових способів їх застосування. Такий механізм забезпечує адаптивність регулювання, але водночас потенційно зменшує передбачуваність правового режиму для розробників, постачальників і користувачів систем штучного інтелекту.

Окремий блок результатів стосувався віддаленої біометричної ідентифікації. Проєкт виходить із заборони використання систем біометричної ідентифікації в реальному часі у публічно доступних місцях для правоохоронних цілей, але одночасно передбачає низку винятків. AgentFlowLaw класифікував цю конструкцію не як доведену правову суперечність, а як потенційну нормативну напруженість між захистом основоположних прав, з одного боку, та потребами правоохоронної діяльності – з іншого.

Аналіз цієї ж групи положень виявив індикатор потенційного корупційного ризику, пов’язаний із дискреційністю застосування винятків. Рішення уповноваженого суб’єкта може залежати від оцінювання тяжкості ситуації, ймовірності та масштабу шкоди, необхідності, пропорційності й терміновості втручання. Водночас наявність процедур авторизації, вимог щодо обґрунтування та інших запобіжників не дозволяє інтерпретувати таку дискрецію як безпосередню корупційну загрозу. Тому результат було позначено саме як індикатор ризику, що потребує експертної оцінки.

Ще одна група результатів пов’язана з процедурами оцінювання відповідності систем високого ризику. У певних випадках значна частина первинної перевірки

покладається на самого постачальника системи. Це створює потенційну суперечність інтересів між необхідністю об'єктивного контролю та комерційною зацікавленістю постачальника у виведенні системи на ринок. Разом із тим проєкт передбачає подальший ринковий нагляд, документування, реєстрацію та можливість перевірки з боку компетентних органів. У пояснювальній частині документа прямо передбачено поєднання попереднього оцінювання відповідності із подальшим контролем систем, що вже перебувають на ринку.

Агент аналізу неоднозначностей виявив окрему проблему операціоналізації якісних вимог. Зокрема, для практичного застосування таких критеріїв, як "достатня прозорість" або "належний рівень точності", необхідні додаткові технічні стандарти, методики та вимірювані показники. Сам по собі такий спосіб нормативного формулювання не свідчить про дефект правового акта, оскільки технологічно нейтральне регулювання повинно зберігати придатність у разі зміни технологій. Однак AgentFlowLaw правильно позначив ці положення як такі, що потребують подальшої нормативної або технічної конкретизації.

Водночас аналіз виявив значну кількість компенсуючих механізмів. До них належать управління ризиками, вимоги до якості даних, документування, реєстрація подій, прозорість, людський контроль, оцінювання відповідності, реєстрація систем та ринковий нагляд. Проєкт прямо пов'язує вимоги до систем високого ризику з даними, документацією, простежуваністю, прозорістю, людським контролем, стійкістю та точністю.

Особливе значення має передбачений проєктом людський контроль за функціонуванням систем високого ризику. Такий підхід є важливим і з погляду методології AgentFlowLaw: результат автоматизованого аналізу не замінює рішення фахівця, а створює структуровану доказову основу для його прийняття.

Узагальнені результати експерименту наведено в табл. 1.

Таблиця 1. Результати експерименту

№	Категорія	Виявлена проблема або особливість	Результат перевірки
1	Правовий ризик	Використання оціночних правових категорій	підтверджено
2	Правовий ризик	Динамічна зміна класифікації систем високого ризику	підтверджено
3	Потенційна суперечність	Заборона біометричної ідентифікації та система винятків	можлива
4	Індикатор корупційного ризику	Дискреційність застосування винятків	потенційний
5	Ризик управління	Внутрішнє оцінювання відповідності постачальником	підтверджено
6	Індикатор корупційного ризику	Поєднання самоконтролю та комерційного інтересу	потенційний
7	Неоднозначність	Критерій достатньої прозорості	підтверджено
8	Неоднозначність	Критерій належного рівня точності	підтверджено

9	Ризик реалізації	Практична перевірність абсолютних вимог до якості даних	можливий
10	Ризик зловживання	Доступ контролюючих органів до даних та програмного коду	контрольований
11	Ризик правозастосування	Варіативність реалізації контрольних процедур	підтверджено

Проведений експеримент демонструє, що AgentFlowLaw дає змогу перейти від звичайного запиту до великої мовної моделі до формалізованої безкодової процедури правового аналізу. Один короткий первинний запит автоматично декомпозується на сукупність спеціалізованих завдань, результати яких перевіряються, зіставляються та інтегруються окремим агентом.

Узагальнено процес можна подати як:

Правовий акт → { ризики, суперечності, неоднозначності, прогалини, індикатори корупційних ризиків, запобіжні механізми } → перевірка → експерт

Суттєвим результатом експерименту є те, що процедура не ототожнює потенційний ризик із встановленим порушенням. Наявність дискреційних повноважень не інтерпретується автоматично як корупційна норма; можливе протиріччя не перетворюється без додаткової перевірки на встановлену правову колізію; разом із ризиками фіксуються передбачені самим нормативним актом запобіжні механізми. Таким чином, у процедурі зберігається невизначеність і допускається існування альтернативних правових інтерпретацій.

Це дає підстави розглядати AgentFlowLaw не як засіб автоматичного формування остаточних юридичних висновків, а як інструмент побудови відтворюваних процедур попереднього аналізу правових актів. Результатом роботи агентської системи є структурований набір тверджень із простежуваним зв’язком між висновком, аргументацією, правовою нормою та джерелом. Остаточна правова кваліфікація при цьому залишається за фахівцем.

Експеримент також показує принципову відмінність AgentFlowLaw від застосування одиничного промпту для аналізу правового документа. У запропонованому підході процес аналізу стає явно заданою процедурою, в якій визначено функції агентів, порядок їх взаємодії, паралельні гілки аналізу, правила перевірки результатів, умови повторного аналізу та формат кінцевого результату. Саме ця властивість дозволяє розглядати AgentFlowLaw як засіб безкодового програмування процедур аналізу правових актів.

Обговорення

Для визначення місця запропонованого фреймворку доцільно порівняти AgentFlowLaw із базовим AgentFlow. AgentFlow є універсальним фреймворком безкодового програмування, у якому поведінка агентів задається структурованою природною мовою із застосуванням логічних примітивів Condition, Loop, Function, Label і Goto; агент характеризується роллю, станом, логікою та механізмом комунікації.

AgentFlowLaw зберігає ці принципи, але доповнює їх засобами, необхідними саме для формування процедур аналізу правових актів. Основні відмінності наведено в табл. 2.

Таблиця 2. Порівняння фреймворків AgentFlow та AgentFlowLaw

Характеристика	**AgentFlow**	**AgentFlowLaw**
Призначення	Універсальне безкодове програмування агентських процедур	Безкодове програмування агентських процедур аналізу правових актів
Предметна орієнтація	Не обмежена конкретною предметною областю	Правові та нормативно-правові документи
Основна роль фреймворку	Формування та виконання структурованих агентських процедур	Формування спеціалізованих процедур правового аналізу
Базові логічні примітиви	Умова, цикл, функція, мітка, перехід	Успадковуються від AgentFlow
Базова модель агента	Роль, стан, логіка, комунікація	Базова модель розширюється правовим контекстом, джерелами, чинністю, доказовістю та простежуваністю
Контекст виконання	Визначається конкретним завданням	Формалізований правовий контекст
Юрисдикція	Не є обов’язковим елементом	Явно задається або визначається під час формування процедури
Часова чинність інформації	Спеціально не регламентується	Перевіряється чинність правових норм на момент аналізу
Ієрархія джерел	Спеціально не визначається	Враховується нормативний статус і співвідношення правових джерел
Робота з джерелами	Залежить від конкретного завдання	Правове твердження пов'язується з нормою та джерелом
Склад агентів	Формується відповідно до довільного завдання	Динамічно формується відповідно до конкретного завдання правового аналізу
Спеціалізовані функції	Залежать від задачі	Пошук норм, аналіз ризиків, суперечностей, неоднозначностей, прогалин, індикаторів корупційних ризиків тощо
Перевірка результатів	Може визначатися процедурою	Виділена як окремий обов'язковий етап
Робота з невизначеністю	Визначається конкретною процедурою	Невизначеність явно фіксується та зберігається у результаті
Суперечності між агентами	Можуть узгоджуватися відповідно до сценарію	Значущі розбіжності зберігаються для подальшої експертної оцінки

Простежуваність	Загальний механізм передачі стану та результатів	Ланцюжок “висновок → аргументація → норма → джерело”
Роль людини	Залежить від конкретного сценарію	Остаточна правова оцінка та юридично значуще рішення залишаються за людиною
Тип результату	Структурований результат виконання агентської процедури	Перевірений і простежуваний результат правового аналізу для експерта

Як видно з табл. 2, AgentFlowLaw не замінює базову модель AgentFlow і не створює незалежну від неї парадигму безкодового програмування. Логічні примітиви, модель агента, механізми формування рою, послідовної та умовно-паралельної взаємодії залишаються спільними. Саме ці властивості становлять основу AgentFlow.

Принципова відмінність полягає у введенні в AgentFlowLaw предметно визначеного правового контексту. До стану агентської процедури додаються юрисдикція, момент часу, вид правового документа, нормативний статус джерел та їх чинність. Тому одна й та сама логічна конструкція в AgentFlowLaw виконується не лише над текстовими даними, а над правовими твердженнями, для яких необхідно встановити нормативне походження та умови застосування.

Друга суттєва відмінність стосується перевірки результатів. У загальному AgentFlow механізми перевірки можуть бути визначені відповідно до конкретної задачі, тоді як в AgentFlowLaw перевірка стає складовою архітектури. Первинний правовий аналіз відокремлюється від перевірки джерел, юрисдикції, часової чинності та простежуваності. Це особливо важливо для роботи великих мовних моделей, оскільки правова сфера вимагає підвищеної уваги до можливих необґрунтованих або помилково приписаних джерелам тверджень.

Ще однією відмінністю є трактування невизначеності та розбіжностей між агентами. AgentFlowLaw не вимагає обов'язкового досягнення єдиного результату. Якщо декілька правових інтерпретацій мають достатнє обґрунтування, вони можуть бути збережені у підсумковому результаті. Аналогічно, неможливість підтвердити норму або встановити її застосовність приводить не до генерування найбільш імовірної відповіді, а до фіксації невизначеного результату.

Таким чином, співвідношення двох фреймворків у найбільш узагальненому вигляді можна подати як

AgentFlowLaw=

AgentFlow+правовий контекст+правові джерела+

+верифікація+простежуваність+контроль людини.

Отже, розвиток AgentFlow у напрямі AgentFlowLaw полягає не просто в застосуванні універсального фреймворку до юридичних текстів. Йдеться про предметну спеціалізацію самого механізму формування агентських процедур, коли специфічні властивості правової інформації стають явними елементами безкодової програми. Це дозволяє розглядати AgentFlowLaw як спеціалізоване розширення AgentFlow, орієнтоване на відтворюваний, перевірюваний і контрольований людиною аналіз правових актів.

Висновки

У роботі розв’язано задачу розроблення спеціалізованого фреймворку AgentFlowLaw для безкодового програмування процедур аналізу правових актів із використанням великих мовних моделей та мультиагентної взаємодії. На відміну від підходу, за якого велика мовна модель безпосередньо формує відповідь на поставлене правове питання, запропонований фреймворк призначений насамперед для програмування процедури отримання, перевірки, обґрунтування та представлення такої відповіді.

AgentFlowLaw розроблено як предметно орієнтоване розширення універсального фреймворку AgentFlow. Збережено його базові принципи безкодового програмування, модель агента та логічні примітиви, але введено компоненти, що враховують специфіку правової інформації: правовий контекст, юрисдикцію, часову чинність норм, нормативний статус джерел, доказову базу, перевірку результатів, простежуваність висновків, явне подання невизначеності та контроль з боку людини. У найбільш узагальненому вигляді запропоноване розширення визначається співвідношенням

AgentFlowLaw=AgentFlow+правовий контекст+правові джерела+верифікація+простежуваність+контроль людини.$\mathrm{AgentFlowLaw} = \mathrm{AgentFlow} + \mathrm{правовий\ контекст} + \mathrm{правові\ джерела} + \mathrm{верифікація} + \mathrm{простежуваність} + \mathrm{контроль\ людини}$.

Наукова новизна роботи полягає у запропонованому підході до представлення аналізу правових актів як безкодової агентської програми, що генерується з опису правової задачі та визначає не лише склад спеціалізованих агентів, а й послідовність і паралельність їх роботи, інформаційні зв’язки, правила перевірки, умови повторного аналізу та структуру кінцевого результату. На відміну від безпосереднього застосування великих мовних моделей до юридичних текстів, об’єктом програмування в AgentFlowLaw стає сама процедура правового аналізу. Для суттєвих висновків задається простежуваний ланцюжок

висновок→аргументація→правова норма→джерело,$\text{висновок} \rightarrow \text{аргументація} \rightarrow \text{правова норма} \rightarrow \text{джерело}$,

а невизначені або недостатньо підтверджені результати не замінюються ймовірними твердженнями, а явно фіксуються для подальшої експертної оцінки.

Використання великих мовних моделей у запропонованому підході дає змогу працювати безпосередньо з природною мовою правових документів, інтерпретувати сформульовані людиною завдання, виділяти нормативні положення та їх взаємозв’язки, формувати аргументацію і генерувати спеціалізовані агентські процедури без написання традиційного програмного коду. При цьому велика мовна модель використовується не лише як засіб генерації текстової відповіді, а як середовище інтерпретації та виконання структурованого опису процедури аналізу.

Мультиагентний підхід забезпечує функціональне розділення складної правової задачі. Пошук норм, аналіз ризиків, виявлення суперечностей, неоднозначностей і прогалин, пошук індикаторів корупційних ризиків, перевірка доказової бази, побудова правової аргументації та узагальнення результатів виконуються спеціалізованими агентами. Це дозволяє поєднувати послідовні та паралельні процедури, незалежно перевіряти отримані результати, зберігати суттєві розбіжності між агентами та повертати окремі результати на повторний аналіз. Таким чином, мультиагентність у AgentFlowLaw є не лише способом декомпозиції обчислень, а механізмом організації контрольованого правового аналізу.

Експериментальне застосування AgentFlowLaw до проєкту Акта про штучний інтелект Європейського Союзу продемонструвало можливість автоматизовано виділяти різні класи результатів: правові ризики, потенційні суперечності, неоднозначності та прогалини, індикатори корупційних ризиків і передбачені самим правовим актом запобіжні механізми. Важливим результатом експерименту стало розмежування встановленого факту та потенційного ризику: дискреційність не ототожнювалася з корупцією, можлива суперечність – із доведеною правовою колізією, а виявлення ризику супроводжувалося пошуком норм, що його обмежують або компенсують.

Практичне значення запропонованого підходу полягає також у зниженні порогу використання агентських технологій у правовій сфері. Юрист або аналітик може задавати необхідну процедуру термінами предметної області, не реалізуючи програмно маршрути взаємодії агентів, цикли перевірки та механізми передачі результатів. Безкодове програмування в цьому випадку виступає засобом формалізації професійної методики правового аналізу та її перетворення на відтворювану агентську процедуру.

Отримані результати дозволяють розглядати AgentFlowLaw як спеціалізований інструментарій на перетині великих мовних моделей, мультиагентних систем, безкодового програмування та правової аналітики. Його призначення полягає не в автоматичній заміні юридичної експертизи, а у формуванні структурованої, перевірюваної та простежуваної інформаційно-аналітичної основи для роботи фахівця. Тому завершальним елементом процедури залишається людина, яка оцінює сформовані аргументи, невизначеності та альтернативні інтерпретації і приймає остаточне юридично значуще рішення.

Подальший розвиток роботи доцільно спрямувати на формування бібліотеки типових процедур AgentFlowLaw для різних класів правових задач, інтеграцію з актуалізованими базами законодавства та судової практики, розроблення формальних показників якості й простежуваності агентського аналізу, а також проведення порівняльних експериментів із одиничними промптами та іншими способами застосування великих мовних моделей у правовій аналітиці.

Декларація щодо використання генеративного штучного інтелекту

Під час підготовки цієї роботи автори використовували систему ChatGPT для перевірки граматики та орфографії, а також перефразування і редагування змісту. Після застосування цих інструментів автори переглянули та відредагували матеріал і несуть повну відповідальність за зміст цієї публікації. Також система ChatGPT використовувалась як процессор для створення агентського промпту по заданню авторського фреймворку AgentFlowLaw.

ПОДЯКИ: Немає

КОНФЛІКТ ІНТЕРЕСІВ: Немає

Використана література

1. Ariai, F., Mackenzie, J. and Demartini, G., 2025. Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges. ACM Computing Surveys, 58(6), pp.1-37. DOI: https://doi.org/10.1145/3777009

2. Magesh, V., Surani, F., Dahl, M., Suzgun, M., Manning, C.D. and Ho, D.E., 2025. Hallucination–free? Assessing the reliability of leading AI legal research tools. Journal of empirical legal studies, 22(2), pp.216-242. DOI: https://doi.org/10.1111/jels.12413

3. Dehghani, F., Dehghani, R., Naderzadeh Ardebili, Y. and Rahnamayan, S., 2025. Large language models in legal systems: A survey. Humanities and Social Sciences Communications, 12(1), p.1977. DOI: https://doi.org/10.1057/s41599-025-05924-3

4. Li, H., Chen, J., Yang, J., Ai, Q., Jia, W., Liu, Y., Lin, K., Wu, Y., Yuan, G., Hu, Y. and Wang, W., 2025, July. Legalagentbench: Evaluating llm agents in legal domain. In Proceedings of the 63rd annual meeting of the association for computational linguistics (volume 1: long papers) (pp. 2322-2344). DOI: https://doi.org/10.18653/v1/2025.acl-long.116

5. Sadowski, A. and Chudziak, J.A., 2025, November. On verifiable legal reasoning: A multi-agent framework with formalized knowledge representations. In Proceedings of the 34th ACM International Conference on Information and Knowledge Management (pp. 2535-2545). DOI: https://doi.org/10.1145/3746252.3761057

6. Chen, B., Li, G., Wu, J., Li, J., Chen, M. and Wang, J., 2026. Agentchain: Blockchain-empowered multi-agent coordination for trustworthy llm question-answering systems. IEEE Transactions on Dependable and Secure Computing. DOI: https://doi.org/10.1109/TDSC.2026.3685256

7. Abendroth, A., 2026. The democratization of software engineering: evolution, definition, and the future of low-code and no-code platforms. Management Review Quarterly, pp.1-36. DOI: https://doi.org/10.1007/s11301-026-00603-2

8. Ланде Д.В., Фурашев В.М. Застосування фреймворку безкодового програмування при вирішенні завдань парламентського контролю // Інформація і право, 2025. - N 2(53). - С. 88-103. DOI: https://doi.org/10.37750/2616-6798.2025.2(53).334055

9. Lande Dmytro, Strashnoy Leonard. AgentFlow - No-Code Agent Framework Based on Logical Primitives. SSRN Preprint 5285664, DOI: https://dx.doi.org/10.2139/ssrn.5285664 (Jun 22, 2025). - 12 p.

10. Інтелектуальна система парламентського контролю : Від семантичного аналізу до інтелектуальної координації в державному управлінні з використанням штучного інтелекту : Монографія / Ланде Д.В., Фурашев В.М., Даник Ю.Г., Страшной Л.Л. - Київ: ТОВ "Інжиніринг", 2025. - 278 с. ISBN 978-617-8180-03-4

11. Застосування фреймворку безкодового програмування на основі великих мовних моделей у правовій сфері. // Трансформаційні процеси правотворчої діяльності та парламентського контролю : матеріали наук.-практ. конф. (Київ, 30 жовт. 2025 р.) [Електронне видання] / упоряд.: В. М. Фурашев, С. О. Дорогих. Київ; Одеса : Фенікс, 2025. -С. 38-40. Режим доступу: https://ippi.org.ua/sites/default/files/zbirnik_tez_30.10.2025.pdf ISBN 978-617-8430-98-6

12. Fernández-Llorca, D., Gómez, E., Sánchez, I. and Mazzini, G., 2025. An interdisciplinary account of the terminological choices by EU policymakers ahead of the final agreement on the AI Act: AI system, general purpose AI system, foundation model, and generative AI. Artificial Intelligence and Law, 33(4), pp.875-888. DOI: https://doi.org/10.1007/s10506-024-09412-y

13. Chi, X., Wang, W., Zhang, Z., Li, A., Huang, Y., Wu, Y., Kuang, K., Sun, C., Liu, X., Wu, F. and Xiong, M., 2026. Legalai research in llm era: data, modeling and evaluation. Artificial Intelligence Review, 59(4), p.118. DOI: https://doi.org/10.1007/s10462-026-11514-9

14. Linna, E. and Linna, T., 2026. Challenges for generative AI in legal reasoning. Discover Artificial Intelligence, 6(1), p.170. DOI: https://doi.org/10.1007/s44163-026-00902-3

15. Bui, L.V., 2025. Advancing patent law with generative AI: Human-in-the-loop systems for AI-assisted drafting, prior art search, and multimodal IP protection. World Patent Information, 80, p.102341. DOI: https://doi.org/10.1016/j.wpi.2025.102341

16. Fatouros, G., Mavrogiorgos, K., Makridis, G., Soldatos, J. and Kyriazis, D., 2026. LLM-Based Agents for Cybersecurity: A Systematic Review of Architectures, Applications, and Open Challenges. Journal of Cybersecurity and Privacy, 6(5), p.159. DOI: https://doi.org/10.3390/jcp6050159

17. Lande D., Strashnoy L. AgentFlowLaw: A No-Code Multi-Agent Framework for Legal Act Analysis. GitHub repository, 2026. https://github.com/DmytroLande/AgentFlowLaw/

Дмитро Володимирович Ланде
доктор технічних наук, професор
керівник Наукового центру Державної наукової установи “Інститут інформації, безпеки і права Національної академії правових наук України”
04053, Україна, м. Київ, пров. Несторівський, 4
завідувач кафедри Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”.
03056, Україна, м. Київ, проспект Берестейський, 37
email: dwlande@gmail.com

Володимир Миколайович Фурашев
кандидат технічних наук, старший науковий співробітник
заступник директора Державної наукової установи “Інститут інформації, безпеки і права Національної академії правових наук України”
04053, Україна, м. Київ, пров. Несторівський, 4
email: vfurashev@gmail.com

Леонард Страшной
старший фахівець з архітектури даних (Senior Data Architect), Університет Каліфорнії (UCLA), Лос-Анджелес, США
ORCID: https://orcid.org/0009-0008-5575-0286

Dmytro V. Lande
Doctor of Technical Sciences, Professor
Head of the Research Center of the State Scientific Institution "Institute of Information, Security and Law of the National Academy of Legal Sciences of Ukraine"
4 Nestorivskyi Lane, Kyiv, 04053, Ukraine
Head of the Department of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
37, Prospect Beresteiskyi, Kyiv, Ukraine, 03056
email: dwlande@gmail.com

Volodymyr M. Furashev
PhD in Technical Sciences, Senior Research Fellow
Deputy Director of the State Scientific Institution "Institute of Information, Security and Law of the National Academy of Legal Sciences of Ukraine"
4 Nestorivskyi Lane, Kyiv, 04053, Ukraine
email: vfurashev@gmail.com

Leonard Strashnoi
Senior Data Architect at the University of California (UCLA), Los Angeles, USA

Рекомендоване цитування: Ланде Д.В., Фурашев В.М., Старшной Л.Л., Фреймворк AgentFlowLaw для безкодового програмування процедур аналізу правових актів *Інформація і право.* № 3(58)/2026. 2026. С. 51-70. https://doi.org/10.37750/2616-6798.2026.3(58).375616

Suggested Citation: Lande D., Furashev V., Strashnoi L. (2026) AgentFlowLaw Framework for Codeless Programming of Legal Act Analysis Procedures. *Information and Law*. 3(58)/2026. 51-70. https://doi.org/10.37750/2616-6798.2026.3(58).375616

Дата надходження статті до редакції: 14.08.2026 р.
Дата прийняття статті до друку після рецензування: 15.08.2026 р.
Дата публікації (оприлюднення): 09.09.2026 р.

~~~~~~~~~~~~~~ * * * ~~~~~~~~~~~~~~
~~~~~~~~~~~~~~