УДК 004.5:004.8 **DOI: https://doi.org/10.30837/PMW.2026.T1.094**

ПРОЕКТУВАННЯ КОГНІТИВНО-АДАПТИВНИХ ІНТЕРФЕЙСІВ ДЛЯ СИСТЕМ ІНТЕРАКТИВНОГО МОДЕЛЮВАННЯ СЦЕНАРІЇВ НА ОСНОВІ ШІ

Олійник В.М.
аспірант, кафедра «Медіасистеми та технології»,
Харківський національний університет радіоелектроніки
ORCID ID: 0000-0002-5584-1531
Бізюк А.В.
кандидат технічних наук, доцент,
професор, кафедра «Медіасистеми та технології»,
Харківський національний університет радіоелектроніки
ORCID ID: 0000-0001-9830-9206

Анотація. *У розділі обґрунтовано концепцію когнітивно-адаптивного інтерфейсу для систем інтерактивного моделювання сценаріїв на основі ШІ. Показано обмеження Chat UI та доцільність переходу до human-in-the-loop архітектури з візуальним контролем, вузлами фіксації, маркуванням аномалій і локальною регенерацією сценарію.*

Ключові слова: *когнітивно-адаптивний інтерфейс, штучний інтелект, великі мовні моделі, human-in-the-loop.*

Вступ

Сучасний етап розвитку інформаційних технологій у сфері інтерактивних медіа характеризується переходом від інструментальних середовищ до систем інтелектуальної генерації контенту. У сфері цифрових ігор, інтерактивних симуляцій та сценарного моделювання ця тенденція безпосередньо пов’язана з розвитком процедурної генерації контенту, що передбачає автоматизоване створення рівнів, правил, карт, місій, персонажів та інших елементів ігрового або симуляційного середовища [1]. Окремим напрямком є генерація процедурного контенту на основі досвіду, у межах якого автоматизована генерація розглядається не лише як технічне створення об’єктів, а як адаптація контенту до досвіду, дій і когнітивних характеристик користувача [2].

Традиційне проєктування складних інтерактивних середовищ, зокрема систем моделювання сценаріїв та варгеймів, в основному базувалося на детермінованих методах. Цей підхід передбачав фіксовані правила пересування ігрових одиниць, визначені штрафи й бонуси, строгі правила «if-then» замість імовірнісних переходів, а також жорсткі сценарні дерева, де кожна гілка сценарію задається явно. У професійній традиції варгеймінгу саме формалізація правил, ролей, умов і процедур є однією з передумов використання гри як інструменту аналізу, навчання або підтримки рішень [3]. Така методологія

забезпечує високий рівень контролю над балансом системи, водночас створюючи бар'єри для масштабованості. Баланс у детермінованих системах не є лінійним: додавання одного фактора, наприклад нового типу техніки, зміни її атрибутів або введення нового правила взаємодії, може порушити логіку всієї системи.

Впровадження великих мовних моделей як центральних генеративних рушіїв змінює парадигму створення медіаконтенту. Замість покрокового моделювання окремих елементів структури користувач переходить до декларативного опису цільового стану системи природною мовою. Великі мовні моделі здатні інтерпретувати високорівневі наративні описи і перетворювати їх на структуровані набори даних, що визначають ландшафт, тактичні характеристики об'єктів, правила взаємодії та логіку сценарію [4]. У сучасних дослідженнях процедурної генерації контенту зазначається, що інтеграція LLM розширює межі класичних PCG-підходів, оскільки мовні моделі можуть генерувати не лише окремі об'єкти, а й правила, рівні та комплексні сценарні структури [5]. Це дозволяє суттєво скоротити цикл розробки прототипу медіапродукту завдяки спрощенню процесу проєктування складних симуляцій.

Проте, прискорення генерації за використанням великих мовних моделей виявило недолік: зростання швидкості створення контенту супроводжується деградацією керованості його внутрішньою логікою. Великі мовні моделі, за своєю стохастичною природою, не гарантують дотримання жорстких топологічних, математичних і балансних обмежень, які є критичними для варгеймів та інших систем інтерактивного моделювання. У дослідженнях галюцинацій LLM підкреслюється, що такі моделі можуть продукувати правдоподібний, але фактично або структурно некоректний зміст, що особливо проблемно для задач, де результат має відповідати формальним правилам і обмеженням [6]. Як наслідок, виникає конфлікт між високою лінгвістичною якістю згенерованого сценарію та його можливою структурною неспроможністю: галюцинаціями об'єктів, порушенням балансу сил, нелогічним розміщенням сутностей на сітці або суперечностями між правилами і наративним описом.

Особливо гостро ця проблема проявляється у традиційних інтерфейсах взаємодії з ШІ, що здебільшого представлені у вигляді вікна чату. Chat UI є зручним для послідовного діалогу природною мовою, однак він є недостатнім для редагування багатовимірних просторово-наративних структур. Якщо користувач бачить помилку у згенерованому сценарії, наприклад аномальну концентрацію юнітів у секторі карти, невідповідність типу юніта місцевості або порушення логіки переміщення, текстовий промпт на виправлення часто призводить не до локальної корекції, а до часткової або повної регенерації даних. У такій ситуації виправлення одного елемента може зруйнувати суміжні, вже коректні структури, що змушує користувача повторно перевіряти весь сценарій.

Отже, основна проблема полягає не лише у здатності ШІ генерувати сценарії, а у здатності користувача контролювати, перевіряти й локально

редагувати результати цієї генерації. Для складних інтерактивних медіасистем сценарій не є лінійним текстом: він одночасно містить просторову топологію, параметри об’єктів, правила взаємодії, часову логіку, балансні обмеження та наративні умови. Подання такого масиву у вигляді тексту, таблиці або сирої JSON/XML-структури створює значне когнітивне навантаження, оскільки користувач змушений самостійно реконструювати просторову модель, виявляти приховані суперечності та оцінювати наслідки кожної зміни.

Це зумовлює необхідність розробки нових моделей людино-машинної взаємодії, які б забезпечили «семантичний міст» між неструктурованим текстом ШІ та жорсткою структурою інтерактивної медіасистеми. Такий інтерфейс має не просто відображати результат генерації, а трансформувати його у зрозумілі для користувача візуальні й семантичні шари: карту, сутності, правила, параметри, конфлікти, зони ризику та можливі точки локального втручання. У межах підходу human-centered artificial intelligence наголошується, що ефективні ШІ-системи мають поєднувати високий рівень автоматизації з високим рівнем людського контролю, а не витісняти користувача з процесу прийняття рішень [7]. Аналогічно, принципи взаємодії людини та штучного інтелекту передбачають, що система повинна підтримувати прозорість стану, можливість корекції, зрозумілий зворотний зв’язок і контроль користувача над результатами автоматизованих дій [8].

У цьому розділі когнітивно-адаптивний інтерфейс розглядається як спосіб подолання обмежень класичного Chat UI у задачах інтерактивного моделювання сценаріїв на основі ШІ. Його ключова функція полягає у збереженні балансу між швидкістю генерації та точністю моделювання: система має дозволяти користувачу бачити структуру сценарію, виявляти аномалії, фіксувати коректні елементи, втручатися у локальні ділянки та запускати часткову корекцію без повної регенерації контексту. Саме така постановка проблеми визначає подальшу логіку дослідження: від аналізу когнітивних обмежень роботи з LLM-сценаріями до проєктування human-in-the-loop архітектури, семантичного мапінгу та механізмів локальної корекції інтерактивних сценаріїв.

Мета та задачі дослідження

Метою дослідження є теоретичне обґрунтування та розробка концепції когнітивно-адаптивного інтерфейсу для систем інтерактивного моделювання сценаріїв на основі штучного інтелекту. Такий інтерфейс має забезпечувати візуальний контроль над результатами генерації, зменшувати когнітивне навантаження користувача при роботі зі складними масивами даних і підтримувати поелементне редагування сценарію без необхідності повної регенерації контексту.

Об’єктом дослідження є процеси людино-машинної взаємодії, що виникають під час проєктування та генерації складних просторово-наративних сценаріїв із використанням великих мовних моделей. У межах цього

дослідження особлива увага приділяється сценаріям варгеймів, оскільки вони поєднують наративний опис, просторову топологію, параметричні сутності, правила взаємодії та балансні обмеження. Розгляд взаємодії у цьому контексті фокусується на трансформації неструктурованого семантичного вводу у формалізовану структуру ігрового поля, систему об'єктів та набір правил, придатних для візуальної перевірки й подальшого редагування.

Предметом дослідження є сукупність інтерфейсних рішень, методів візуалізації та когнітивних моделей, що забезпечують верифікацію, контроль і локальну корекцію результатів LLM-генерації. Особлива увага приділяється механізмам подолання ефекту «чорної скриньки» через впровадження проміжних шарів маніпуляції даними, які дозволяють людині виступати активним контролером у human-in-the-loop процесі формування інтерактивного сценарію.

Для досягнення поставленої мети визначено такі **задачі дослідження**.

1. Проаналізувати обмеження традиційного чат-інтерфейсу у задачах генерації, перевірки та редагування складних просторово-наративних сценаріїв.
2. Обґрунтувати доцільність застосування human-in-the-loop підходу як принципу збереження контролю користувача над результатами LLM-генерації.
3. Визначити когнітивні чинники, що ускладнюють роботу користувача зі згенерованими сценаріями, зокрема зовнішнє когнітивне навантаження, просторово-семантичний дисонанс і перевантаження сирими даними.
4. Запропонувати модель семантичного мосту між природномовним описом сценарію, структурованою моделлю даних і візуальним інтерфейсом користувача.
5. Описати ключові елементи когнітивно-адаптивного інтерфейсу, що забезпечують візуальну перевірку, фіксацію коректних елементів, виявлення аномалій і локальну корекцію окремих фрагментів сценарію.
6. Здійснити аналітичне порівняння класичного Chat UI та запропонованого когнітивно-адаптивного інтерфейсу за критеріями керованості, когнітивного навантаження, локальності редагування та ризику каскадної регенерації.

Основна частина

1 Когнітивні обмеження роботи з LLM-сценаріями

Проєктування медіасистем, що інтегрують великі мовні моделі, потребує врахування не лише технічних можливостей генерації, а й обмежень людського сприйняття. У системах інтерактивного моделювання сценаріїв користувач працює не з окремим текстом, а з багаторівневою структурою, що поєднує наративний опис, просторову топологію, параметри об'єктів, правила взаємодії, часові умови та балансні обмеження. Тому основним ризиком стає не сам факт наявності великого обсягу даних, а спосіб їх подання користувачу. Теорія когнітивного навантаження виходить із того, що робоча пам'ять людини має

обмежену пропускну здатність, а ефективність виконання складних завдань залежить від того, як саме інформація організована й представлена [9]. У подальших роботах ця теорія була розвинена як модель проєктування інформаційних середовищ, що мають знижувати зайве навантаження на користувача й підтримувати побудову коректних ментальних схем [10].

У контексті LLM-сценаріїв загальне когнітивне навантаження користувача можна умовно представити як суму трьох компонентів: внутрішнього, зовнішнього та доречного навантаження (табл. 1). Внутрішнє навантаження визначається об'єктивною складністю сценарію: кількістю типів юнітів, правилами їхньої взаємодії, топологією карти, кількістю умов перемоги, тригерів і балансних обмежень. Це навантаження не може бути повністю усунуте інтерфейсом, оскільки воно є властивістю самої предметної області. Зовнішнє навантаження виникає через спосіб подання інформації: надлишковий текст, неструктуровані відповіді моделі, потребу ручної перевірки JSON/XML-структур, дублювання параметрів або необхідність шукати приховані суперечності у великому масиві згенерованих даних. Доречне навантаження пов'язане з корисною когнітивною роботою користувача: побудовою ментальної моделі сценарію, оцінкою стратегічної логіки, ухваленням рішень щодо балансу та корекції структури.

Таблиця 1 – Компоненти когнітивного навантаження користувача під час роботи з LLM-сценаріями

Компонент навантаження	**Джерело виникнення у сценарному моделюванні**	**Роль інтерфейсу**
Внутрішнє навантаження	Складність правил, кількість юнітів, топологія карти, сценарні умови	Не усувається повністю, але може бути структуроване через візуальні шари
Зовнішнє навантаження	Сирий текст, JSON/XML, неструктуровані відповіді LLM, потреба ручної перевірки помилок	Має бути мінімізоване через візуалізацію, фільтрацію та маркування аномалій
Доречне навантаження	Побудова ментальної моделі сценарію, аналіз балансу, прийняття рішень	Має підтримуватися інтерфейсом як продуктивна аналітична робота

Проблема LLM-сценаріїв полягає в тому, що користувач часто змушений витрачати основну частину когнітивних ресурсів не на творче або аналітичне моделювання, а на перевірку цілісності згенерованого результату. Наприклад, сценарій може бути лінгвістично переконливим, але містити логічні або просторові помилки: розміщення важкої техніки на непрохідній місцевості, дублювання одного й того самого юніта в різних секторах, порушення правил видимості або несумісність між описом місії та фактичними умовами перемоги. Такі помилки пов'язані з ширшою проблемою галюцинацій великих мовних моделей, коли модель формує правдоподібний, але фактично або структурно некоректний результат [6]. У сценарних системах ця проблема набуває особливої

ваги, оскільки помилка може бути неочевидною в тексті, але критичною для балансу симуляції.

Фундаментальним когнітивним обмеженням у такій ситуації є просторово-семантичний дисонанс. Йдеться про конфлікт між послідовним характером текстового подання і паралельним характером просторового сприйняття. Текст або JSON-структура змушують користувача послідовно зчитувати окремі параметри, зіставляти їх між собою і внутрішньо реконструювати карту сценарію. Наприклад, запис виду

`{“side”: “player”, “unit”: “tank”, “coord”: “2:3”, “terrain”: “forest”}`

не показує проблему безпосередньо. Користувач має спочатку інтерпретувати тип юніта, потім координату, потім тип місцевості, а після цього співвіднести ці параметри з правилами сценарію. У візуальному інтерфейсі така сама суперечність може бути сприйнята майже миттєво, якщо юніт відображений на карті, а несумісна ділянка підсвічена як конфлікт.

Ця відмінність узгоджується з класичним висновком про те, що діаграмні й просторові репрезентації можуть бути ефективнішими за текстові, оскільки вони роблять частину відношень безпосередньо видимими, а не прихованими в послідовності символів [11]. Для варгейм-сценаріїв це означає, що топологічні дані, зони контролю, радіуси дії, маршрути руху, концентрація сил і конфлікти правил не повинні залишатися лише у текстовому або табличному вигляді. Якщо інтерфейс змушує користувача виконувати «віртуальний рендеринг» карти у власній пам’яті, він переносить на людину ту частину роботи, яку має виконувати UI-шар.

Окремим джерелом перевантаження є надмірна деталізація. Один юніт у сценарній системі може мати десятки параметрів: тип, сторону конфлікту, координати, стан, рівень укріплення, боєздатність, дальність дії, запас ходу, мораль, обмеження видимості, зв’язок із тригерами та залежність від сценарних умов. Якщо всі ці параметри показані одночасно, користувач отримує формально повну, але практично малокеровану картину. У такому випадку зростає ризик «паралічу аналізу», коли кількість доступної інформації ускладнює ухвалення рішення замість того, щоб його підтримувати.

Для зниження такого навантаження доцільно застосовувати стратегію прогресивного розкриття. Її сутність полягає в тому, що інтерфейс показує користувачу не весь масив даних одразу, а лише той рівень деталізації, який потрібний для поточного завдання. У сучасних дослідженнях складних людино-машинних інтерфейсів progressive disclosure розглядається як спосіб зменшити навчальне й операційне навантаження через подання функцій та інформації шарами. У контексті LLM-сценаріїв це означає, що дані мають бути організовані щонайменше на трьох рівнях.

Перший рівень - візуальний дескриптор. На цьому рівні користувач бачить основну структуру сценарію: карту, типи сутностей, їхню належність до сторін, базовий стан і просторове розміщення. Другий рівень - контекстні атрибути, які

відкриваються при фокусуванні, виборі юніта або виділенні області. Саме тут доцільно показувати параметри боєздатності, модифікатори місцевості, локальні правила або попередження про конфлікти. Третій рівень - семантичне ядро, тобто повний опис згенерованої структури: вихідний промпт, JSON/XML-представлення, історія змін і службові параметри моделі. Цей рівень має бути доступний для експертного редагування, але не повинен бути основним режимом взаємодії.

Отже, когнітивне обмеження роботи з LLM-сценаріями полягає не в тому, що користувач не здатний аналізувати складні моделі, а в тому, що традиційні текстові інтерфейси змушують його виконувати зайві операції перетворення, пошуку й верифікації. Когнітивно-адаптивний інтерфейс має зменшити зовнішнє навантаження, переносячи частину перевірки структури на рівень UI: візуалізувати просторові відношення, приховувати другорядні параметри до моменту запиту, підсвічувати потенційні аномалії та підтримувати поетапне занурення в деталі. Саме ця логіка створює підґрунтя для переходу до human-in-the-loop архітектури, у якій користувач не перевіряє весь згенерований масив вручну, а взаємодіє з ним через керовані, візуально зрозумілі й когнітивно структуровані точки контролю.

2 Human-in-the-Loop як принцип контролю генерації

Обмеження, описані у попередньому підрозділі, безпосередньо вказують на необхідність зміни моделі взаємодії користувача з генеративною системою. Якщо LLM-сценарій подається як готовий результат, сформований у відповідь на один промпт, користувач фактично працює з «чорною скринькою»: він бачить вхідний запит і фінальний текст або набір даних, але не має доступу до проміжної логіки формування просторових, параметричних і балансних рішень. Для простих текстових завдань така модель може бути прийнятною, однак для інтерактивного моделювання сценаріїв вона створює критичну проблему контролю. Користувач не лише не бачить, як саме модель розподілила об’єкти, правила й обмеження, а й не може втрутитися у процес до моменту появи помилки.

Альтернативою є перехід від моделі «чорної скриньки» до моделі «скляної скриньки», у якій інтерфейс розкриває користувачу не внутрішні параметри нейромережі як такі, а структурну логіку згенерованого сценарію: які сутності створені, де вони розміщені, які правила до них застосовано, які обмеження порушено і які елементи потребують перевірки. У підході human-centered artificial intelligence підкреслюється, що ефективні ШІ-системи мають поєднувати високі можливості автоматизації з високим рівнем людського контролю [7]. Отже, прозорість у цьому випадку означає не повне пояснення роботи LLM, а створення інтерфейсних умов, за яких користувач може контролювати результат генерації на рівні сценарних структур.

У такій архітектурі ШІ доцільно розглядати не як автономного автора сценарію, а як керований генеративний інструмент або партнера у змішаній

ініціативі. Дослідження mixed-initiative co-creativity у сфері ігрового дизайну показують, що продуктивна взаємодія людини й алгоритму виникає тоді, коли обидві сторони беруть участь у створенні рішення, але людина зберігає можливість спрямовувати, обмежувати й оцінювати машинну генерацію [12]. Для LLM-сценаріїв це означає, що модель може пропонувати варіанти ландшафту, складу сил, тригерів або параметрів юнітів, але остаточна валідація, фіксація та корекція мають залишатися за користувачем.

Для формалізації такого розподілу доцільно звернутися до класичних моделей рівнів автоматизації. Шкала Sheridan-Verplank описує автоматизацію як континуум від повністю ручного керування до повністю автономного виконання дій системою [13]. Подальша модель Parasuraman, Sheridan і Wickens уточнює, що автоматизація може застосовуватися до різних стадій діяльності: збору інформації, аналізу, вибору рішення та виконання дії [14]. Це важливо для ШІ-інтерфейсів, оскільки проблема полягає не в самому факті автоматизації, а в тому, на якому етапі вона забирає контроль у користувача.

Класичний Chat UI у роботі з LLM фактично піднімає автоматизацію до високого рівня одразу на кількох стадіях. Модель не лише аналізує запит, а й самостійно обирає структуру сценарію, генерує фінальний набір об'єктів і подає результат як неподільний масив. Користувач отримує лише постфактум-можливість оцінити відповідь. Для сценарного моделювання це створює деструктивну модель бінарного вибору: «прийняти весь сценарій» або «відхилити/перегенерувати весь сценарій». Такий підхід суперечить самій природі складних інтерактивних систем, де частина елементів може бути коректною, частина - потребувати уточнення, а частина - містити критичні помилки.

Наприклад, LLM може згенерувати загалом прийнятну карту, логічну структуру місії та збалансований склад сторін, але помилково розмістити один тип техніки у зоні, де за правилами він не може діяти. У чат-інтерфейсі користувач змушений описувати цю локальну помилку текстом: вказувати координати, тип об'єкта, характер порушення і бажану зміну. Однак новий промпт може змінити не лише проблемний фрагмент, а й сусідні ділянки, параметри інших юнітів або навіть загальну логіку сценарію. У результаті виправлення одного елемента запускає каскадну регенерацію, після якої користувач має повторно перевіряти вже узгоджені частини системи.

Саме тому для когнітивно-адаптивного інтерфейсу принциповим є перенесення автоматизації з рівня повного авторства на рівень підтримки локальних рішень. LLM може виконувати високорівневу генерацію, пропонувати варіанти, аналізувати обмеження й виявляти потенційні суперечності, але інтерфейс має забезпечувати користувачу можливість мікровтручання. Мікровтручання означає локальну дію користувача щодо конкретного елемента або області сценарію: зафіксувати юніт, заблокувати шар карти, змінити параметр, виділити проблемну зону, запросити альтернативу лише для одного фрагмента або підтвердити запропоновану корекцію.

Такий підхід відповідає принципам human-AI interaction, згідно з якими система має давати користувачу можливість контролювати результат, коригувати помилки, розуміти стан системи та отримувати зворотний зв'язок щодо дій ШІ [8]. У сценарному моделюванні це означає, що користувач повинен працювати не з одним фінальним текстом, а з набором керованих структур: картою, шарами, сутностями, правилами, попередженнями та точками локальної корекції. Тоді ШІ перестає бути автономним генератором, який щоразу створює новий сценарій, і стає інструментом ітеративного уточнення вже наявної моделі.

Отже, Human-in-the-Loop у системах LLM-сценарного моделювання слід розуміти не як формальну присутність людини після завершення генерації, а як архітектурний принцип розподілу контролю. Генеративна модель бере на себе швидке створення варіантів і первинний аналіз, тоді як користувач зберігає контроль над прийняттям рішень, фіксацією коректних елементів і локальною корекцією помилок. Саме такий розподіл створює основу для подальшого проєктування семантичного мосту між природномовним описом, структурованою моделлю даних і візуальним інтерфейсом сценарію.

3 Семантичний міст між наративом ШІ та структурою сценарію

Після визначення когнітивних обмежень і необхідності human-in-the-loop контролю ключовим завданням стає перетворення результату LLM-генерації на форму, придатну для візуальної перевірки та локального редагування. Природномовний опис сценарію є зручним для користувача, але недостатнім для інтерактивної симуляції, оскільки він не гарантує однозначного визначення об'єктів, координат, правил, тригерів і обмежень. Тому між текстом, який генерує або вводить користувач, і візуальним інтерфейсом сценарію має існувати проміжний рівень - семантичний міст.

У межах цього дослідження семантичний міст розглядається як механізм переходу від природномовного наративу до структурованої моделі сценарію. Його функція полягає не лише у технічному перетворенні тексту на JSON/XML або інший формат даних, а у формалізації змісту: виділенні сутностей, їхніх властивостей, просторових координат, правил взаємодії та сценарних залежностей. Такий підхід узгоджується з принципами онтологічного моделювання, де онтологія визначається як формальна специфікація понять предметної області та зв'язків між ними [15]. Для генеративних систем це означає, що LLM-результат має бути не фінальним текстом для читання, а джерелом структурованих елементів, які можуть бути перевірені, візуалізовані й змінені користувачем.

Перехід від природномовного опису до структурованої моделі можна подати як послідовність трьох рівнів (рис. 1). На першому рівні користувач або LLM формує наративний опис: наприклад, *«механізована група просувається через лісисту місцевість, де противник підготував засідку»*. На другому рівні система виділяє з цього опису формалізовані сутності: тип місцевості, сторони конфлікту, юніти, тактичні ролі, умови видимості, можливі тригери й

обмеження. На третьому рівні ці сутності перетворюються на елементи інтерфейсу: ділянки карти, маркери юнітів, зони контролю, правила взаємодії та параметри сценарію. Дослідження формалізації текстових промптів до систем штучного інтелекту також підкреслюють важливість переходу від неструктурованого запиту до формалізованих параметрів, придатних для подальшої машинної обробки [4].

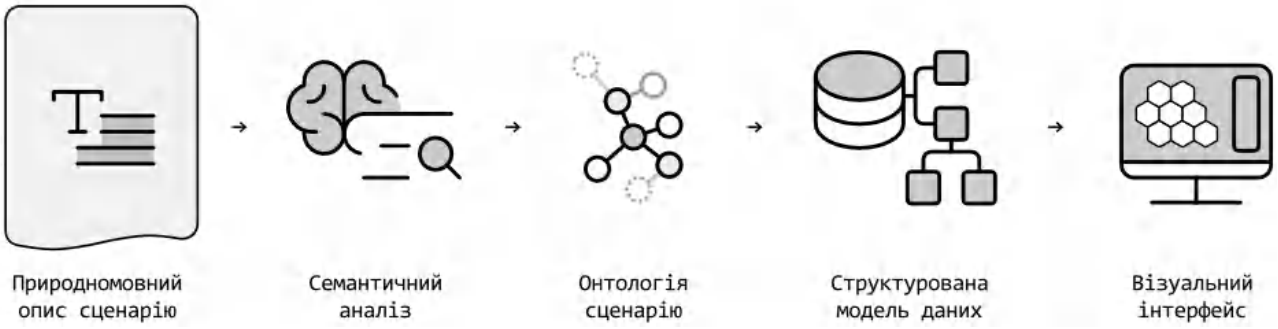

Рисунок 1 – Модель семантичного мосту між природномовним описом і візуальною структурою сценарію

Основою такого переходу є базова онтологія сценарію. Вона задає набір класів сутностей, які можуть бути присутні у сценарії, їхні властивості та допустимі відношення. У випадку варгеймів, мінімальна онтологія має містити щонайменше шість груп: ландшафт, юніти, правила, тригери, цілі та обмеження. Ландшафт визначає типи місцевості й умови просторового переміщення. Юніти описують активні об’єкти сценарію: сторони, типи сил, бойові характеристики, статуси та доступні дії. Правила фіксують допустимі взаємодії між об’єктами. Тригери задають події, які запускаються за певних умов. Цілі визначають бажані стани сценарію для кожної сторони. Обмеження встановлюють межі допустимих змін, наприклад заборону розміщення певного типу техніки на непрохідній місцевості (табл. 2).

Таблиця 2 – Базові компоненти онтології сценарію для інтерактивного моделювання

Компонент онтології	Приклад сутностей	Функція у сценарії
Ландшафт	ліс, дорога, міст, висота, річка	Визначає умови руху, видимості, укриття та доступності ділянок
Юніти	піхота, бронетехніка, артилерія	Формують активні об’єкти сценарію з параметрами й діями
Правила	дальність ураження, штраф руху, зона контролю	Обмежують і регулюють взаємодію між об’єктами
Тригери	вхід у сектор, втрата юніта, досягнення цілі	Запускають події або зміну стану сценарію
Цілі	утримати позицію, прорвати оборону, евакуювати групу	Визначають логіку оцінювання успішності сценарію
Обмеження	заборона руху, ліміт ресурсів, баланс сторін	Запобігають структурним і балансним помилкам

Наступним етапом є просторове співставлення. Структурована модель сценарію має бути перенесена на гексагональну або іншу тактичну сітку, оскільки саме просторове розміщення визначає значну частину логіки симуляції. Якщо LLM генерує сутність *«танковий підрозділ на північному фланзі»,* інтерфейс має перетворити цей опис на конкретний об'єкт із координатами, зоною дії, допустимими маршрутами й обмеженнями місцевості. У цьому процесі природномовні відношення «поруч», «позаду», «на фланзі», «у зоні видимості» мають бути переведені у формальні просторові відношення: координати, відстані, сусідство клітин, сектори огляду або радіуси дії.

Саме на цьому рівні виникає потреба у поділі даних на шари. Замість єдиного неподільного масиву сценарій має бути представлений як сукупність взаємопов'язаних, але окремо контрольованих шарів. Terrain layer містить інформацію про місцевість і її властивості. Entity layer відповідає за розміщення юнітів, об'єктів і ресурсів. Rule layer зберігає правила, які визначають допустимі дії та взаємодії. Event/trigger layer описує події, що активуються за певних умов.

Такий поділ дозволяє користувачу працювати не з усім сценарієм одночасно, а з тим рівнем структури, який потребує перевірки або редагування.

Практична цінність шарової моделі полягає у тому, що вона створює умови для локального контролю. Якщо помилка стосується ландшафту, користувач може редагувати terrain layer без зміни складу сил. Якщо проблема полягає у неправильному розміщенні юніта, корекція відбувається в entity layer. Якщо порушено логіку активації події, змінюється event/trigger layer. Отже, семантичний міст не лише структурує дані, а й готує їх до подальшого застосування вузлів фіксації, локальної корекції та контрольованої регенерації.

У цій логіці інтерфейс виконує роль семантичного перекладача між LLM і користувачем. Для LLM сценарій є набором текстових і структурних залежностей, тоді як для користувача він має бути представленим як зрозумілий простір дій: карта, об'єкти, попередження, правила та можливі точки втручання. Інтерфейс повинен приховувати технічну складність моделі даних, але не приховувати змістові зв'язки, які впливають на коректність сценарію. Тому користувач має бачити не сирий результат генерації, а інтерпретовану структуру, де кожен елемент має візуальне представлення, семантичне значення і зв'язок із правилами сценарію.

Таким чином, семантичний міст є центральним елементом когнітивно-адаптивного інтерфейсу. Він забезпечує перехід від природномовної генерації до керованої інтерактивної моделі, у якій LLM створює початковий матеріал, а UI перетворює його на структуру, доступну для аналізу, перевірки й локального редагування. Без такого проміжного рівня користувач залишається залежним від текстового результату моделі; з ним же сценарій стає системою взаємопов'язаних шарів, сутностей і правил, над якими можна здійснювати точковий human-in-the-loop контроль.

4 Архітектура когнітивно-адаптивного інтерфейсу

Архітектура когнітивно-адаптивного інтерфейсу має будуватися не як розширення чат-вікна, а як окреме середовище керування згенерованим сценарієм. Його основна функція полягає у перетворенні результатів LLM-генерації на структуру, яку користувач може візуально перевіряти, локально редагувати й частково фіксувати. Такий підхід узгоджується з принципами human-AI interaction, згідно з якими система має підтримувати видимість стану, можливість корекції, зрозумілий зворотний зв'язок і контроль користувача над результатами автоматизованих дій [8]. У межах сценарного моделювання це означає, що інтерфейс має показувати не лише фінальний текст сценарію, а й структуру його просторових, семантичних і правилових залежностей.

Центральним елементом такої архітектури є візуальна карта сценарію. Вона виконує роль основного простору контролю, у якому користувач бачить розміщення юнітів, типи місцевості, зони дії, маршрути, конфлікти правил і потенційні аномалії. Для варгеймів природним форматом є гексагональна сітка, оскільки вона дозволяє формалізувати сусідство клітин, відстані, напрямки руху та зони контролю. Проте запропонований принцип не обмежується лише гексагональними картами: він може бути застосований до квадратних сіток, графових карт, часових сценарних діаграм або інших тактичних представлень. Важливим є те, що карта стає не ілюстрацією до тексту, а головним інтерфейсним шаром, через який користувач здійснює контроль над згенерованою моделлю.

Поряд із картою має функціонувати панель семантичного контексту. Її завдання полягає у поясненні того, що саме означає вибраний елемент сценарію: до якої сторони він належить, які параметри має, які правила на нього впливають, які обмеження з ним пов'язані та які фрагменти вихідного промпту або LLM-відповіді стали підставою для його створення. Така панель забезпечує зв'язок між візуальним об'єктом і його семантичним походженням. У контексті візуальної аналітики це відповідає підходу семантичної взаємодії, за якого користувач взаємодіє з візуальним представленням даних, а система перетворює ці дії на сигнали для аналітичної моделі (рис. 2, табл. 3) [16].

Щоб уникнути перевантаження користувача, архітектура має використовувати progressive disclosure параметрів. На базовому рівні карта показує лише ключові візуальні дескриптори: тип юніта, сторону, стан, тип місцевості та основні попередження. При фокусуванні або виборі об'єкта відкриваються контекстні атрибути: бойові параметри, модифікатори місцевості, обмеження руху, пов'язані тригери й локальні ризики балансу. Повний технічний опис, зокрема JSON/XML-представлення, історія генерації або службові параметри моделі, має бути доступний лише у спеціальному режимі експертного редагування. Такий поділ дозволяє користувачу зберігати загальне розуміння сценарію, не занурюючись у всі параметри одночасно.

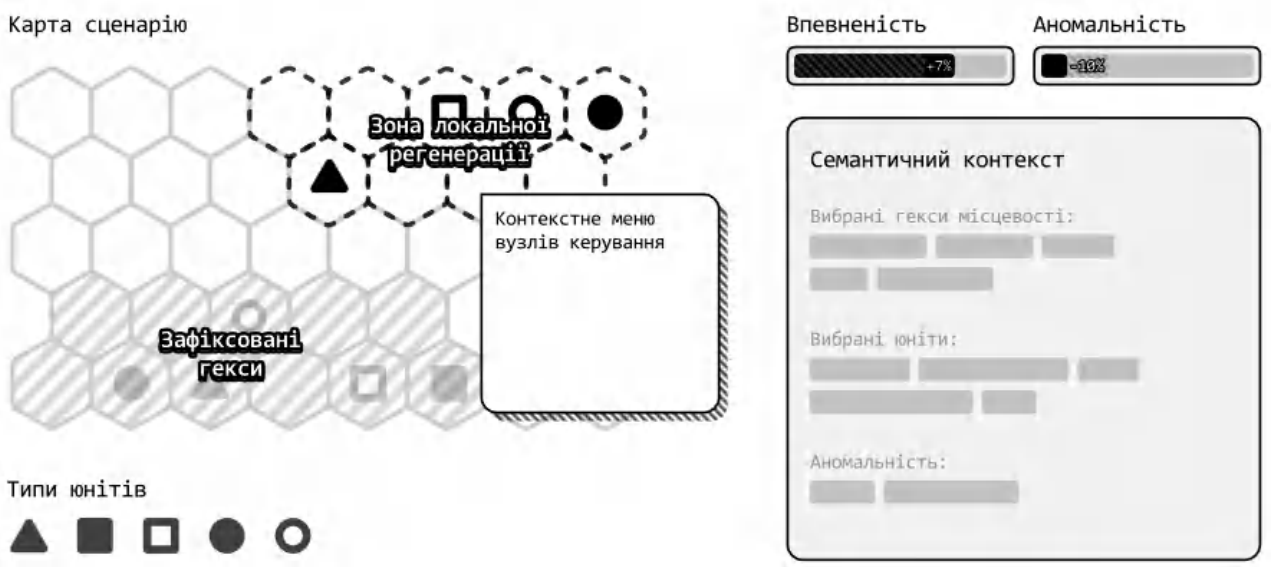

Рисунок 2 – Архітектура когнітивно-адаптивного інтерфейсу для контролю LLM-сценарію

Таблиця 3 – Ключові компоненти когнітивно-адаптивного інтерфейсу

Компонент	Функція	Очікуваний ефект
Візуальна карта	Просторове представлення сценарію, юнітів, зон і маршрутів	Зменшення потреби у ручній реконструкції сценарію з тексту
Панель семантичного контексту	Пояснення параметрів, правил і походження вибраного елемента	Підвищення прозорості LLM-результату
Поступове розкриття інформації	Поступове відкриття параметрів залежно від дії користувача	Зниження інформаційного перевантаження
Індикатори надійності	Позначення елементів із різним рівнем надійності	Пріоритезація перевірки
Теплові карти аномалій	Виявлення зон ризику, концентрації або конфлікту	Швидке виявлення проблемних ділянок
Контрольні вузли (control nodes)	Фіксація коректних елементів сценарію	Запобігання каскадній регенерації
Семантичне відновлення (semantic inpainting)	Локальна регенерація вибраного фрагмента	Корекція без руйнування всього сценарію

Окремим компонентом є індикатори впевненості. Вони мають показувати, наскільки система «впевнена» у коректності певного елемента або рішення. Наприклад, низький рівень впевненості може бути присвоєний юніту, розміщення якого суперечить частині правил, або тригеру, який має нечітку умову активації. Візуалізація невизначеності є важливою для складних систем прийняття рішень, оскільки допомагає користувачу розрізняти стабільні дані та елементи, що потребують додаткової перевірки [17]. У LLM-сценаріях такі індикатори не повинні подаватися як абсолютна математична істина; їхня роль полягає у пріоритезації уваги користувача.

Для виявлення просторових проблем доцільно застосовувати теплові карти аномалій. Вони можуть показувати зони надмірної концентрації юнітів, ділянки з підвищеним ризиком дисбалансу, конфлікти між типом місцевості й розміщеними об’єктами, а також сектори, де правила сценарію застосовуються неоднозначно. Наприклад, якщо в одному секторі карта містить надто багато одиниць однієї сторони або LLM розмістила артилерію у зоні, де вона не має лінії вогню, інтерфейс має не приховувати цю проблему в тексті, а візуально маркувати її на карті. Такий підхід знижує зовнішнє когнітивне навантаження, оскільки користувач бачить проблемну область без потреби вручну шукати її у структурі даних.

Поряд із тепловими картами мають використовуватися механізми підсвічування конфліктів правил. Якщо певний об’єкт порушує обмеження руху, несумісний із типом місцевості або активує тригер у неправильний момент, інтерфейс має явно позначити цей конфлікт. Важливо, щоб попередження було пов’язане не лише з об’єктом, а й із правилом, яке порушено. Наприклад, користувач має бачити не загальне повідомлення «помилка сценарію», а конкретне пояснення: *«бронетехніка розміщена на непрохідній ділянці»* або *«тригер евакуації активується до досягнення цільової зони».* Це робить верифікацію сценарію змістовною, а не формальною.

Додаткову роль виконують локальні підказки щодо ризиків балансу. Вони мають попереджати користувача про потенційні наслідки зміни ще до запуску регенерації. Наприклад, якщо користувач додає новий підрозділ у сектор, система може показати, що це змінює співвідношення сил у зоні, збільшує щільність оборони або робить одну з цілей надто легкою для досягнення. Такі підказки не повинні автоматично забороняти дію, оскільки остаточне рішення залишається за користувачем, але вони мають надавати інформацію для усвідомленого втручання.

Найважливішим архітектурним елементом є контрольні вузли. Вони позначають елементи сценарію, які користувач визнав коректними й не хоче змінювати під час наступних ітерацій генерації. Такими вузлами можуть бути окремі гекси, групи клітин, юніти, параметри, правила, тригери або цілі сценарію. Контрольні вузли перетворюють вибраний елемент на обмеження для наступного запиту до LLM: модель має враховувати його як незмінну частину контексту. Це дозволяє уникнути ситуації, коли виправлення одного локального елемента випадково змінює вже перевірені частини сценарію.

Механічно це реалізується через freezing або pinning. Freezing означає тимчасове блокування шару, області або параметра від змін. Pinning означає явне закріплення елемента як важливого для сценарної логіки. Наприклад, користувач може зафіксувати terrain layer, щоб LLM змінювала лише розташування юнітів; закріпити конкретний підрозділ, щоб він залишався у вибраному секторі; або заблокувати параметр дальності дії, якщо він уже узгоджений із правилами. Усі ці дії мають бути видимими в інтерфейсі, щоб користувач розумів, які частини сценарію відкриті для регенерації, а які захищені від змін.

Локальна регенерація реалізується через принцип семантичного відновлення. За аналогією з редагуванням зображень, де змінюється лише вибрана область, у сценарній системі змінюється лише визначений фрагмент семантичної структури. Користувач виділяє проблемну область на карті або обирає конкретний об'єкт, після чого формулює короткий мікропромпт: наприклад, «замінити цей підрозділ на легшу піхоту», «зменшити щільність сил у секторі» або «зробити тригер активації пізнішим». Система передає LLM не весь сценарій як відкритий простір для перегенерації, а локальний контекст: вибрану область, сусідні незмінні елементи, зафіксовані вузли та правила, які не можна порушувати.

У результаті когнітивно-адаптивний інтерфейс працює як середовище керованого редагування, а не як пасивний переглядач відповіді ШІ. Його архітектура поєднує візуальну карту, семантичний контекст, індикатори невизначеності, маркування аномалій, вузли фіксації та локальну регенерацію. Саме така комбінація дозволяє перейти від глобальної регенерації сценарію до ітеративного уточнення окремих фрагментів. Це зберігає переваги LLM як швидкого генеративного інструмента, але повертає користувачу контроль над структурною логікою, балансом і просторовою коректністю інтерактивної симуляції.

5 Механізм локальної корекції сценарію

Механізм локальної корекції є практичним продовженням описаної архітектури когнітивно-адаптивного інтерфейсу. Його головна функція полягає у тому, щоб замінити глобальну регенерацію сценарію керованим циклом часткового редагування. У класичному Chat UI користувач змушений формулювати текстове уточнення до всієї відповіді моделі, навіть якщо помилка стосується одного юніта, одного сектору карти або одного правила. У когнітивно-адаптивному інтерфейсі корекція має бути прив'язана до конкретної області, конкретного шару або конкретної сутності сценарію. На рис. 3 схематично показано етапи роботи із запропонованим інтерфейсом.

Першим етапом такого циклу є генерація початкового сценарію. LLM отримує природномовний опис ситуації, наприклад склад сторін, тип місцевості, цілі, обмеження та бажану динаміку подій. На цьому етапі модель формує первинну структуру сценарію: опис карти, перелік юнітів, правила взаємодії, тригери, умови перемоги та інші параметри. Однак цей результат не повинен одразу розглядатися як фінальний. Він є початковою гіпотезою системи, яка потребує інтерфейсної інтерпретації, візуалізації та перевірки.

Другий етап полягає у перетворенні згенерованого сценарію на візуальну структуру. Інтерфейс розкладає результат LLM на окремі шари: місцевість, сутності, правила, тригери та обмеження. Після цього ці дані відображаються у вигляді карти, панелі семантичного контексту, списку подій і параметрів. Саме на цьому етапі користувач перестає працювати з лінійним текстом і отримує

доступ до просторової моделі сценарію. Це зменшує потребу у ручній реконструкції зв'язків між описом, координатами, об'єктами та правилами.

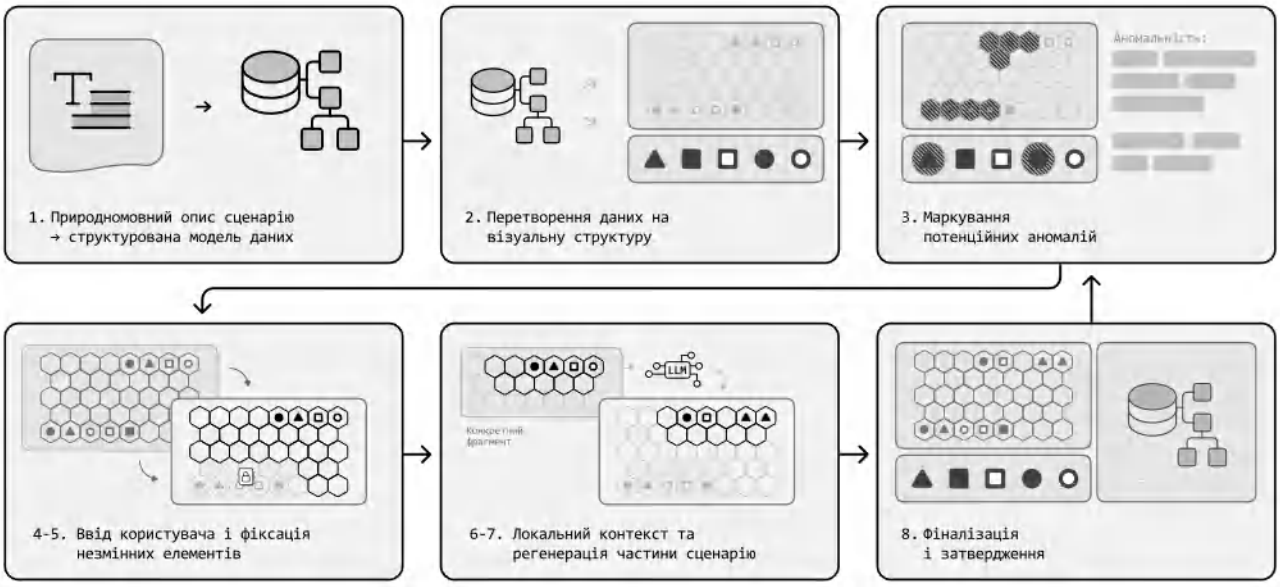

Рисунок 3 – Цикл локальної корекції LLM-сценарію в когнітивно-адаптивному інтерфейсі

Третім етапом є автоматичне маркування потенційних аномалій. Система перевіряє згенеровану структуру за набором правил і евристик: чи відповідає тип юніта типу місцевості, чи не перевищена щільність сил у секторі, чи не порушені умови видимості, чи не активується тригер передчасно, чи не створює розташування сторін очевидного дисбалансу. Результатом такої перевірки є візуальні позначки на карті, індикатори впевненості або попередження у панелі контексту. Важливо, що система не обов'язково має автоматично виправляти всі проблеми; її завдання – зробити їх видимими для користувача.

Четвертий етап передбачає дію користувача: він виділяє проблемну область або конкретний елемент сценарію. Це може бути один гекс, група клітин, юніт, зона контролю, тригер або правило. Виділення області є принципово важливим, оскільки саме воно задає межі корекції. На відміну від текстового промпту, де користувач змушений словесно описувати координати й контекст, візуальне виділення дозволяє безпосередньо вказати, яка частина сценарію має бути змінена.

П'ятий етап – фіксація незмінних елементів. Перед запуском локальної регенерації UI визначає, які частини сценарію мають залишитися стабільними. Це можуть бути сусідні гекси, зафіксовані юніти, правила руху, балансні обмеження або вже перевірені тригери. Такі елементи передаються у запит до LLM як обмеження, а не як відкриті для редагування дані. Завдяки цьому система запобігає каскадній зміні сценарію, коли виправлення одного локального фрагмента випадково змінює інші коректні частини моделі.

На шостому етапі LLM отримує локальний контекст. На відміну від глобального запиту, локальний контекст містить лише ті дані, які потрібні для конкретної корекції: вибрану область, її поточний стан, сусідні незмінні елементи, релевантні правила та коротку інструкцію користувача. Наприклад,

замість промпту *«перероби розміщення сил на карті»* система формує запит типу: *«у зоні X змінити склад підрозділів так, щоб бронетехніка не перебувала на непрохідній місцевості; сусідні гекси, цілі сценарію та баланс сторін залишити незмінними».* Така структура запиту зменшує простір невизначеності для моделі.

Сьомий етап полягає у генерації тільки обмеженої частини сценарію. LLM не створює сценарій заново, а пропонує зміну для вибраного фрагмента. Це може бути заміна типу юніта, перенесення об'єкта на суміжну клітину, зміна параметра, уточнення тригера або створення альтернативного локального розміщення. Після цього інтерфейс має порівняти новий фрагмент із зафіксованими елементами й перевірити, чи не виникли нові конфлікти. Якщо корекція не відповідає обмеженням, вона не повинна автоматично вбудовуватися у сценарій.

Восьмий етап – перевірка та інтеграція результату у загальний стан. Користувач переглядає запропоновану зміну у візуальному середовищі, оцінює її вплив на баланс, правила й просторову логіку, після чого приймає, відхиляє або додатково уточнює результат. Прийнята зміна вбудовується у загальну модель сценарію як новий стан, тоді як попередній стан має зберігатися для можливості скасування або порівняння. Таким чином, кожна локальна корекція стає контрольованою ітерацією, а не непередбачуваною регенерацією всього сценарію.

Отже, механізм локальної корекції реалізує human-in-the-loop підхід на рівні конкретної дії користувача. Він поєднує генеративні можливості LLM із візуальним вибором області, фіксацією незмінних елементів, локальним контекстом і перевіркою результату перед інтеграцією. Такий цикл знижує когнітивне навантаження, зменшує ризик каскадної регенерації та дозволяє підтримувати структурну цілісність сценарію протягом кількох ітерацій редагування.

Результати дослідження

1 Порівняння Chat UI та когнітивно-адаптивного інтерфейсу

Порівняння класичного Chat UI та когнітивно-адаптивного інтерфейсу доцільно здійснювати не за зручністю введення промпту, а за здатністю підтримувати контроль над складною сценарною структурою. Для простих генеративних завдань чат-інтерфейс є достатнім, оскільки результат може бути перевірений шляхом послідовного читання. Проте у системах інтерактивного моделювання сценарій складається з кількох взаємопов'язаних рівнів: просторової карти, юнітів, параметрів, правил, тригерів, цілей і балансних обмежень. Тому ключовим критерієм стає не швидкість генерації першої відповіді, а керованість подальшої перевірки й редагування. У таблиці 4 наведено порівняння Chat UI та когнітивно-адаптивного інтерфейсу в задачах LLM-сценарного моделювання.

Наведене порівняння показує, що Chat UI є ефективним переважно на етапі формування початкового запиту або загального опису сценарію. Його слабкість проявляється на етапах перевірки, уточнення та часткового редагування. Користувач змушений працювати з лінійною текстовою відповіддю, тоді як сама структура сценарію є нелінійною і просторово розподіленою. Через це зростає ризик пропустити приховану помилку або спровокувати небажану зміну вже узгоджених фрагментів.

Таблиця 4 – Порівняння Chat UI та когнітивно-адаптивного інтерфейсу в задачах LLM-сценарного моделювання

Критерій	Chat UI	Когнітивно-адаптивний інтерфейс
Рівень контролю	Контроль здійснюється переважно через текстові уточнення після генерації. Користувач впливає на результат опосередковано.	Контроль здійснюється через карту, шари, вузли фіксації, локальні обмеження та візуальні точки втручання.
Локальність редагування	Локальна зміна описується текстом, але модель може змінити інші частини сценарію.	Користувач виділяє конкретну область, шар або об’єкт, а корекція обмежується заданим контекстом.
Ризик каскадної регенерації	Високий: виправлення одного фрагмента може спричинити зміну вже коректних елементів.	Низький: зафіксовані елементи й контрольні вузли обмежують простір регенерації.
Когнітивне навантаження	Високе: користувач має самостійно зіставляти текст, координати, правила й параметри.	Нижче: інтерфейс візуалізує просторові відношення, аномалії та релевантні параметри.
Прозорість логіки	Обмежена: результат подається як суцільна відповідь без явного розділення на шари й залежності.	Вища: сценарій подається як структура з ландшафтом, юнітами, правилами, тригерами й обмеженнями.
Підтримка балансу	Баланс перевіряється вручну або через повторні промпти, що не гарантує стабільності результату.	Баланс підтримується через попередження, індикатори ризику, теплові карти та локальні обмеження.
Придатність для складних сценаріїв	Обмежена: ефективність знижується зі зростанням кількості об’єктів, правил і залежностей.	Вища: інтерфейс підтримує багаторівневу структуру, поступове розкриття інформації та локальні ітерації.

Когнітивно-адаптивний інтерфейс, навпаки, переносить основну взаємодію з рівня текстового уточнення на рівень структурного керування. Його перевага полягає не у відмові від природної мови, а у поєднанні промптингу з візуальною картою, семантичними шарами, маркуванням аномалій і локальною регенерацією. У такій моделі природна мова використовується для формулювання наміру, а інтерфейс визначає межі, контекст і обмеження зміни.

Отже, у задачах інтерактивного моделювання сценаріїв когнітивно-адаптивний інтерфейс має вищу придатність, оскільки краще відповідає природі самого об’єкта редагування. Складний сценарій потребує не лише генерації, а й постійної верифікації, фіксації коректних частин і контрольованого внесення змін. Саме ці функції залишаються слабким місцем Chat UI і становлять основну перевагу запропонованого підходу.

2 Аналітична модель часових витрат

Для аналітичного порівняння двох моделей взаємодії доцільно використати скорочену Keystroke-Level Model (KLM). У класичному вигляді KLM застосовується для прогнозування часу виконання користувацьких дій в інтерактивних системах через сумування окремих операторів: введення, вказування, ментальної підготовки, перемикання режимів та очікування реакції системи. У межах цього дослідження модель використовується не для точного емпіричного вимірювання, а для аналітичного порівняння двох сценаріїв редагування LLM-сценарію.

Скорочену модель часових витрат можна подати так:

$$T_{\text{execute}} = T_K + T_P + T_H + T_M + T_R, \quad (1)$$

де T_K – час на введення тексту або параметрів;

T_P – час на вказування, вибір області або об’єкта;

T_H – час на перемикання між режимами взаємодії;

T_M – час на когнітивну оцінку результату;

T_R – час очікування відповіді системи.

Для LLM-сценарного моделювання особливо важливими є T_M і T_R, оскільки користувач витрачає значний час на перевірку згенерованої структури, а система – на повторну генерацію тексту або фрагментів даних.

Розглянемо типову задачу (табл. 5): у згенерованому сценарії виявлено помилку, коли бронетехніка розміщена на ділянці, що за правилами є непрохідною.

У сценарії А користувач працює через класичний Chat UI. Він має знайти помилку в тексті або структурі даних, сформулювати уточнювальний промпт, описати координати, характер помилки й бажану зміну. Після цього LLM генерує нову відповідь, яка може змінити не лише проблемну ділянку, а й інші елементи сценарію. Отже, після отримання результату користувач змушений повторно перевіряти значну частину карти, правил і параметрів.

У сценарії Б користувач працює через когнітивно-адаптивний інтерфейс. Помилка маркується на карті як конфлікт між типом юніта і типом місцевості. Користувач виділяє проблемний гекс або групу гексів, фіксує сусідні незмінні елементи та формулює коротку локальну інструкцію. LLM отримує не весь сценарій як відкритий простір для перегенерації, а обмежений контекст: вибрану

область, правила, зафіксовані елементи й необхідну зміну. У результаті генерується лише локальний фрагмент, який потім перевіряється й інтегрується у загальний стан.

Таблиця 5 – Порівняння часових витрат у двох сценаріях редагування

Оператор	**Сценарій А: Chat UI**	**Сценарій Б: когнітивно-адаптивний інтерфейс**
T_K	Високий: потрібно описати проблему, координати, обмеження й бажану зміну текстом	Нижчий: вводиться короткий мікропромпт до вже виділеної області
T_P	Низький або відсутній, оскільки взаємодія переважно текстова	Вищий: користувач виділяє область або об’єкт на карті
T_H	Середній: перемикання між читанням відповіді, промптом і перевіркою структури	Середній або нижчий: основна взаємодія зосереджена у візуальному середовищі
T_M	Високий: потрібно самостійно знайти помилку й повторно перевірити наслідки регенерації	Нижчий: помилка візуально маркована, а зона зміни обмежена
T_R	Високий: можлива регенерація значної частини сценарію	Нижчий: генерується лише локальний фрагмент

З таблиці 5 видно, що когнітивно-адаптивний інтерфейс може збільшувати T_P, оскільки користувач виконує додаткову дію вибору області на карті. Проте це збільшення є функціонально виправданим, оскільки воно зменшує два найвагоміші джерела втрат – T_M і T_R. Візуальне маркування аномалії скорочує час когнітивного пошуку та оцінки, а локальний контекст зменшує обсяг генерації й подальшої перевірки.

Отже, аналітична модель показує, що перевага когнітивно-адаптивного інтерфейсу полягає не в усуненні всіх операцій користувача, а в перерозподілі часових витрат. Частина текстового введення і повторної ментальної перевірки замінюється точним візуальним вибором області та контрольованою локальною регенерацією. Це робить процес редагування сценарію більш передбачуваним, зменшує ризик каскадних змін і підвищує ефективність роботи зі складними LLM-сценаріями.

3 Узагальнені результати концептуальної моделі

Проведений аналіз дозволяє сформулювати узагальнені результати запропонованої концептуальної моделі. Насамперед, у межах дослідження обґрунтовано необхідність розглядати інтерфейс для LLM-сценарного моделювання не як допоміжний засіб введення промптів, а як повноцінну архітектуру контролю генерації. Запропонована модель когнітивно-адаптивного інтерфейсу поєднує візуальну карту, семантичний контекст, progressive disclosure параметрів, індикатори невизначеності, маркування аномалій, вузли фіксації та локальну регенерацію. У сукупності ці елементи створюють

середовище, у якому користувач може не лише отримувати результат від ШІ, а й керувати його перевіркою, уточненням і частковим редагуванням.

Другим результатом є обґрунтування переходу від класичного Chat UI до human-in-the-loop взаємодії. Показано, що чат-інтерфейс є придатним для формування початкового опису сценарію, але недостатнім для контролю складної просторово-наративної структури. Його основне обмеження полягає в тому, що користувач отримує фінальний масив даних і може впливати на нього переважно через нові текстові запити. На відміну від цього, HITL-підхід передбачає постійну участь користувача у валідації, виборі, фіксації та корекції елементів сценарію.

Третім результатом є визначення ролі семантичного мапінгу між природномовним текстом і просторовою структурою сценарію. У запропонованій моделі LLM-результат не використовується безпосередньо як фінальний сценарій, а проходить через проміжний шар формалізації. На цьому рівні наративний опис перетворюється на систему сутностей: ландшафт, юніти, правила, тригери, цілі та обмеження. Далі ці сутності розподіляються за візуальними шарами інтерфейсу, що дозволяє користувачу працювати не з абстрактним текстом, а з керованою моделлю сценарію.

Четвертим результатом є опис механізму контрольних вузлів, як засобу запобігання каскадній регенерації. Вузли фіксації дозволяють позначати вже перевірені елементи сценарію як незмінні для наступних ітерацій. Завдяки цьому локальна корекція одного фрагмента не повинна призводити до зміни сусідніх, уже узгоджених частин моделі. Такий механізм є принциповим для складних симуляцій, де навіть незначна зміна параметра може вплинути на баланс, просторову логіку або сценарну послідовність.

П’ятим результатом є аналітичне обґрунтування потенційного зниження когнітивних і часових витрат при локальному редагуванні. Скорочена KLM-модель показує, що когнітивно-адаптивний інтерфейс не усуває всі дії користувача, але перерозподіляє їх у більш продуктивний спосіб. Замість довгого текстового уточнення, повної регенерації та повторної перевірки сценарію користувач виконує точний вибір області, фіксує незмінні елементи й запускає локальну регенерацію. Це зменшує передусім витрати на когнітивну оцінку результату та очікування відповіді системи.

Отже, запропонована концептуальна модель демонструє, що ефективність ШІ у сценарному моделюванні залежить не лише від якості генеративної моделі, а й від архітектури взаємодії з нею. Когнітивно-адаптивний інтерфейс забезпечує перехід від одноразової генерації тексту до ітеративного, візуально контрольованого процесу створення сценарію, у якому користувач зберігає контроль над логікою, балансом і структурною цілісністю інтерактивної системи.

Висновки

У результаті проведеного дослідження встановлено, що використання великих мовних моделей у системах інтерактивного моделювання сценаріїв відкриває значні можливості для прискорення створення медіаконтенту, але водночас формує нову проблему керованості. LLM здатні швидко генерувати наративні описи, варіанти структури сценарію, параметри об'єктів і загальну логіку подій, однак сам факт генерації не гарантує структурної коректності результату. Для складних просторово-наративних систем, зокрема варгеймів, критичними залишаються не лише швидкість створення сценарію, а й можливість перевірити баланс, просторову топологію, відповідність правил, коректність тригерів і логіку розміщення сутностей.

Показано, що класичний Chat UI є недостатнім інтерфейсом для роботи з такими сценаріями. Його основне обмеження полягає у лінійному текстовому характері взаємодії, тоді як сам сценарій є багатовимірною структурою. Користувач змушений перевіряти просторові, параметричні та правилові залежності через текст, таблиці або сирі структури даних, що створює надмірне зовнішнє когнітивне навантаження. Крім того, виправлення локальної помилки через текстовий промпт часто не гарантує локальності зміни: модель може перегенерувати не лише проблемний фрагмент, а й інші, вже коректні частини сценарію.

Отже, основною проблемою LLM-сценарного моделювання є не стільки здатність ШІ створювати початковий сценарій, скільки здатність користувача контролювати, верифікувати й уточнювати цей результат. Саме контроль і верифікація визначають придатність генеративної системи для складних інтерактивних медіаструктур. Якщо користувач не має інструментів для виявлення аномалій, фіксації коректних елементів і локальної корекції помилок, то перевага швидкої генерації частково нівелюється витратами на повторну перевірку й ризиком руйнування вже узгодженої структури.

У роботі обґрунтовано доцільність переходу до когнітивно-адаптивного інтерфейсу, який поєднує візуалізацію сценарію, human-in-the-loop контроль, маркування аномалій і механізми локальної корекції. Такий інтерфейс має виконувати роль семантичного мосту між природномовним описом, структурованою моделлю даних і візуальним середовищем редагування. Він не замінює LLM, а задає умови, у яких результати генерації стають видимими, перевірюваними й керованими. Візуальна карта, панель семантичного контексту, progressive disclosure параметрів, індикатори невизначеності, теплові карти аномалій і підсвічування конфліктів правил створюють основу для зниження когнітивного навантаження користувача.

Окреме значення має механізм контрольних вузлів. Він дозволяє позначати перевірені елементи сценарію як незмінні для наступних ітерацій генерації. Завдяки цьому користувач отримує можливість переходити від глобальної регенерації всього сценарію до керованого редагування окремих фрагментів.

Локальна регенерація через семантичне відновлення дає змогу змінювати лише визначену область, шар або сутність, зберігаючи контекст і не руйнуючи вже узгоджені частини моделі. Саме цей перехід є ключовим для практичного застосування LLM у системах інтерактивного моделювання.

Аналітичне порівняння Chat UI та когнітивно-адаптивного інтерфейсу показало, що запропонований підхід має переваги за критеріями керованості, локальності редагування, прозорості логіки, підтримки балансу та придатності для складних сценаріїв. Скорочена KLM-модель засвідчила, що когнітивно-адаптивний інтерфейс може зменшувати насамперед витрати на когнітивну оцінку результату та очікування відповіді системи. Це досягається завдяки тому, що користувач не описує всю проблему текстом і не перевіряє весь сценарій після кожного уточнення, а працює з конкретною областю, конкретними обмеженнями і контрольованою локальною зміною.

Таким чином, запропонована концепція доводить, що ефективність генеративного ШІ у сценарному моделюванні залежить не лише від якості мовної моделі, а й від архітектури інтерфейсу, через який користувач взаємодіє з результатами генерації. Когнітивно-адаптивний інтерфейс дозволяє зберегти переваги LLM як швидкого генеративного інструмента, але водночас повертає користувачу контроль над структурою, балансом і логікою сценарію. Це створює основу для розробки нових редакторів інтерактивного контенту, у яких людина виступає не пасивним отримувачем відповіді ШІ, а активним куратором і верифікатором семантичної моделі.

Перспективи подальших досліджень пов’язані з емпіричною перевіркою запропонованої концепції. Наступним етапом має стати створення прототипу когнітивно-адаптивного інтерфейсу та його тестування на типових задачах сценарного моделювання. Доцільним є проведення порівняльних експериментів із класичним Chat UI для вимірювання часу редагування, кількості помилок, рівня когнітивного навантаження та стабільності результатів після кількох ітерацій корекції. Окремим напрямом є розробка агентних механізмів балансування сценаріїв, автоматизована перевірка правил, виявлення конфліктів між шарами моделі та інтеграція таких механізмів у human-in-the-loop цикл редагування.

Список літератури.

1. Togelius, J., Yannakakis, G. N., Stanley, K. O., & Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. IEEE Transactions on Computational Intelligence and AI in Games, 3(3), 172-186. https://doi.org/10.1109/TCIAIG.2011.2148116.

2. Yannakakis, G.N., & Togelius, J. (2011). Experience-driven procedural content generation. IEEE Transactions on Affective Computing, 2(3), 147–161. https://doi.org/10.1109/T-AFFC.2011.6.

3. Ministry of Defence. (2017). Defence wargaming handbook. https://www.gov.uk/government/publications/defence-wargaming-handbook.

4. Oliinyk, V., Biziuk, A., Deineko, Z., & Chelombitko, V. (2025). Formalization of text prompts to artificial intelligence systems. Eastern-European Journal of Enterprise Technologies, 5 (2(137)), 84-97. https://doi.org/10.15587/1729-4061.2025.335473.

5. Farrokhi Maleki, M., & Zhao, R. (2024). Procedural content generation in games: A survey with insights on emerging LLM integration. AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 20(1), 167-178. https://doi.org/10.1609/aiide.v20i1.31877.

6. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems, 43(2), 1-55. https://doi.org/10.1145/3703155.

7. Shneiderman, B. (2020). Human-centered artificial intelligence: Reliable, safe & trustworthy. International Journal of Human–Computer Interaction, 36(6), 495-504. https://doi.org/10.1080/10447318.2020.1741118.

8. Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., & Horvitz, E. (2019). Guidelines for human-AI interaction. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. (p. 1-13). Association for Computing Machinery. https://doi.org/10.1145/3290605.3300233.

9. Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Cognitive Science, 12(2), 257-285. https://doi.org/10.1207/s15516709cog1202_4.

10. Sweller, J., van Merriënboer, J. J. G., & Paas, F. G. W. C. (1998). Cognitive architecture and instructional design. Educational Psychology Review, 10(3), 251-296. https://doi.org/10.1023/A:1022193728205.

11. Larkin, J.H., & Simon, H.A. (1987). Why a diagram is (sometimes) worth ten thousand words. Cognitive Science, 11(1), 65-100. https://doi.org/10.1111/j.1551-6708.1987.tb00863.x.

12. Yannakakis, G. N., Liapis, A., & Alexopoulos, C. (2014). Mixed-initiative co-creativity. In Proceedings of the 9th International Conference on the Foundations of Digital Games. (p. 1-8). https://www.um.edu.mt/library/oar/handle/123456789/29459.

13. Moray, N., Rodriguez, D., & Clegg, B. (2000). Levels of automation in process control. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 44(1), 93-96. https://doi.org/10.1177/154193120004400125.

14. Parasuraman, R., Sheridan, T.B., & Wickens, C.D. (2000). A model for types and levels of human interaction with automation. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 30(3), 286-297. https://doi.org/10.1109/3468.844354.

15. Gruber, T.R. (1995). Toward principles for the design of ontologies used for knowledge sharing? International Journal of Human-Computer Studies, 43(5-6), 907-928. https://doi.org/10.1006/ijhc.1995.1081.

16. Endert, A. (2014). Semantic interaction for visual analytics: Toward coupling cognition and computation. IEEE Computer Graphics and Applications, 34(4), 8-15. https://doi.org/10.1109/MCG.2014.73.

17. Bisantz, A. M., Cao, D., Jenkins, M., Pennathur, P. R., Farry, M., Roth, E., Potter, S.S., & Pfautz, J. (2011). Comparing uncertainty visualizations for a dynamic decision-making task. Journal of Cognitive Engineering and Decision Making, 5(3), 277-293. https://doi.org/10.1177/1555343411415793.