

**Міністерство освіти і науки України
Державний університет
інформаційно-комунікаційних технологій**

Золотухіна О.А

МЕТОДИ ТА ЗАСОБИ АНТРОПОЦЕНТРИЧНИХ ОБЧИСЛЕНЬ ТА ТЕХНОЛОГІЙ

Навчальний посібник

Київ – 2025

УДК 004.5:004.8:004.9

Рецензенти:

Олександр ВОВНА

д.т.н, професор, професор кафедри
програмних систем і технологій

Київського національного університету імені Тараса Шевченка

Дмитро ІВАНОВ

д.т.н, доцент, професор кафедри
комп'ютерних наук та інформаційних технологій

Житомирського державного університету імені Івана Франка

Рекомендовано до видання

Вченою радою Навчально-наукового інституту інформаційних технологій
Державного університету інформаційно-комунікаційних технологій
(протокол № 8 від 2 грудня 2025 року)

004.5:004.8:004.9 (075.8) 3-81 Золотухіна О.А. Методи та засоби антропоцентричних обчислень та технологій. Навчальний посібник. К.:ДУІКТ, 2025. – 84 с.

Посібник присвячено вивченню теоретичних засад і прикладних аспектів антропоцентричних обчислень та технологій як сучасної парадигми проєктування комп'ютерних і інформаційних систем, орієнтованих на людину. Видання систематизує концептуальні основи антропоцентричного підходу, розглядає моделі взаємодії людини й інтелектуальних систем, а також підхід human-in-the-loop та етапи проєктування систем із включенням людини в контур прийняття рішень. Окрему увагу приділено застосуванню інтелектуальних обчислень і штучного інтелекту в антропоцентричних системах, типових ролях людини у взаємодії з алгоритмами та етичним аспектам використання таких технологій.

Посібник може бути використаний здобувачами третього (освітньо-наукового) рівня вищої освіти спеціальностей галузі знань F «Інформаційні технології», під час опанування дисциплін, пов'язаних із дослідженнями у сфері інтелектуальних, когнітивних та людиноорієнтованих технологій. Матеріали посібника також будуть корисними для науковців і розробників, які досліджують або створюють технології, орієнтовані на людину.

© Державний університет
інформаційно-комунікаційних технологій, 2025

ЗМІСТ

Передмова	5
Розділ 1. Концептуальні засади антропоцентричного підходу	8
1.1 Сутність та еволюція антропоцентричної парадигми.....	8
1.2. Людина як центральний елемент системи «людина–комп’ютер».....	12
Розділ 2. Концептуальні моделі антропоцентричних систем.....	16
2.1. Когнітивно-орієнтовані моделі.....	17
2.2. Контекстуально-орієнтовані моделі	23
2.3. Адаптивно-поведінкові моделі	28
Розділ 3. Підхід human-in-the-loop	39
3.1. Загальні положення підходу Human-in-the-Loop.....	39
3.2. Етапи проектування систем на основі підходу Human-in-the-Loop ...	40
3.2.1. Визначення ролі людини у системі.....	41
3.2.2 Аналіз потреб та поведінки користувачів	43
3.2.3. Проектування архітектури взаємодії	46
3.2.4. Розробкв прототипів і моделювання взаємодії.....	48
3.2.5. Інтеграційне тестування та налаштування.....	51
3.2.6. Валідація, адаптація та масштабування.....	54
Розділ 4. Застосування інтелектуальних обчислень та штучного інтелекту в антропоцентричних системах	57
4.1. Роль інтелектуальних обчислень в антропоцентричних системах.....	57
4.2. Моделі взаємодії людини та алгоритмів і систем штучного інтелекту	60
4.1.1. Людина як спостерігач	60

4.2.2. Людина як коректор рішень.....	62
4.2.3. Людина як співавтор рішень (human–AI collaboration).....	63
4.2.4. Людина як джерело знань і навчальних даних	65
4.2.5. Людина як джерело зворотного зв’язку та адаптації	66
4.3. Етичні аспекти.....	67
Література	71
Додаток А. Приклад чекліста для оцінки антропоцентричності системи .	77
Додаток Б. Приклади анкет для оцінювання взаємодії з користувачем	79
Б.1 System Usability Scale (SUS)	79
Б.2 User Experience Questionnaire (UEQ)	80

ПЕРЕДМОВА

Стрімкий розвиток комп'ютерних, інформаційних та інтелектуальних технологій зумовлює переосмислення ролі людини в сучасних соціо-технічних системах. Автоматизація, алгоритмічне прийняття рішень і системи штучного інтелекту дедалі глибше інтегруються в професійну діяльність, освіту, медицину, управління та повсякденне життя. У цих умовах стає очевидним, що ефективність і безпека технологій не можуть оцінюватися виключно за технічними показниками. Ключовим чинником стає здатність систем підтримувати людину, враховувати її когнітивні можливості, обмеження, контекст діяльності та цінності.

Антропоцентричні обчислення та технології формують відповідь на ці виклики, пропонуючи підхід, у межах якого людина розглядається не як пасивний користувач або джерело помилок, а як центральний елемент системи та активний учасник процесів прийняття рішень. Такий підхід передбачає інтеграцію інженерних, когнітивних, поведінкових і етичних міркувань на всіх етапах життєвого циклу системи – від аналізу потреб і проєктування архітектури взаємодії до валідації, адаптації та масштабування.

Метою цього навчального посібника є систематизація теоретичних засад і методичних підходів до антропоцентричних обчислень та технологій, а також формування у здобувачів освіти й фахівців цілісного уявлення про принципи проєктування комп'ютерних та інформаційних систем із урахуванням ролі людини на різних етапах їх життєвого циклу. Актуальність посібника зумовлена необхідністю переосмислення традиційних техноцентричних моделей автоматизації та переходу до підходів, що забезпечують прозорість, керованість, пояснюваність і етичну обґрунтованість використання інтелектуальних алгоритмів. Особливе

значення це має для систем, у яких помилки або некоректні рішення можуть мати суттєві соціальні, правові чи безпекові наслідки.

Структура посібника побудована з урахуванням логіки поступового формування знань – від загальних концептуальних положень до прикладних аспектів застосування антропоцентричних підходів. У матеріалі послідовно розглядаються основи антропоцентричної парадигми, роль людини в інтелектуальних системах, типові моделі взаємодії людини та комп'ютерних і інформаційних систем, а також місце інтелектуальних обчислень у процесах підтримки прийняття рішень. Значну увагу приділено аналізу моделей взаємодії людини та штучного інтелекту. Окремим змістовним акцентом є розгляд етичних питань, пов'язаних із застосуванням інтелектуальних технологій у антропоцентричних системах.

Теоретичні положення подано з урахуванням сучасних наукових концепцій у галузі інтелектуальних обчислень, людиноорієнтованого проєктування та когнітивних наук. Матеріал викладено з акцентом на причинно-наслідкові зв'язки між технічними рішеннями та їх впливом на людину. Приділено увагу чіткому розмежуванню ролей людини та алгоритмічних компонентів, а також обґрунтуванню меж автоматизації в антропоцентричних системах.

Посібник орієнтований насамперед на здобувачів третього (освітньо-наукового) рівня вищої освіти спеціальностей галузі знань F «Інформаційні технології», які вивчають дисципліни, пов'язані з антропоцентричними, людиноорієнтованими та інтелектуальними системами, зокрема, для вивчення дисципліни «Методи та засоби антропоцентричних обчислень та технологій» в рамках освітньої програми «Інженерія програмного забезпечення» спеціальності F2 «Інженерія програмного забезпечення». Водночас матеріали посібника можуть бути корисними здобувачам другого (магістерського) рівня, науково-педагогічним працівникам, аспірантам, а також фахівцям-практикам, які займаються проєктуванням, аналізом або

експлуатацією систем із використанням методів штучного інтелекту та інтелектуальних обчислень. Завдяки міждисциплінарному характеру викладу посібник може бути використаний також у суміжних галузях, де важливу роль відіграє взаємодія людини й автоматизованих систем.

РОЗДІЛ 1. КОНЦЕПТУАЛЬНІ ЗАСАДИ АНТРОПОЦЕНТРИЧНОГО ПІДХОДУ

1.1 Сутність та еволюція антропоцентричної парадигми

Сутність антропоцентричної парадигми полягає в розгляді людини як центрального елемента будь-якої системи – технологічної, соціальної, когнітивної чи інформаційної – і в орієнтації на її потреби, можливості, обмеження та цінності під час розроблення та застосування обчислювальних технологій. Вона поєднує інженерні підходи з когнітивними, поведінковими та етичними міркуваннями і протиставляє себе техноцентричним моделям, де основним пріоритетом, як правило, виступає ефективність алгоритму або інфраструктури [1].

Відмінність між антропоцентричним і техноцентричним підходами полягає в пріоритетах та точці фокусу системи, тобто у тому, на що орієнтоване проєктування та функціонування технологій (рис.1.1).

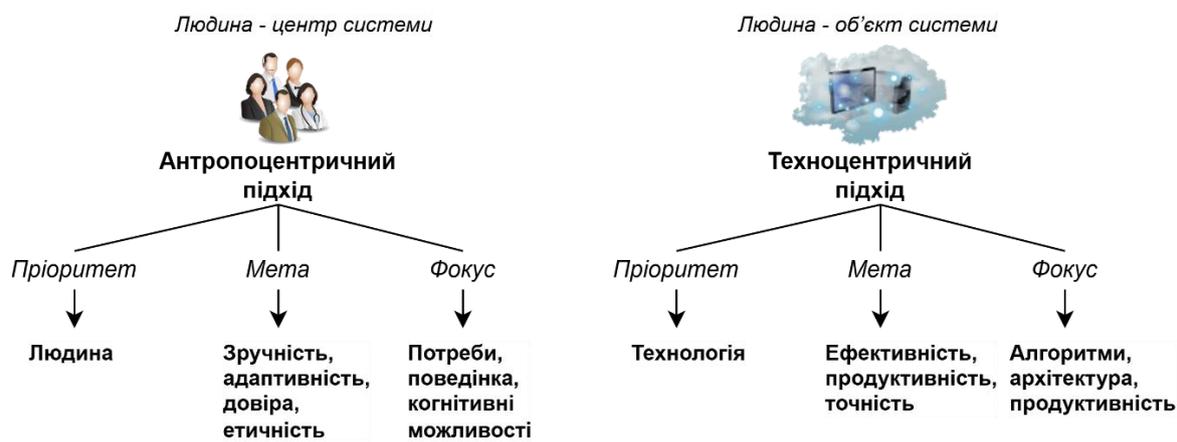


Рис. 1.1. Відмінність між антропоцентричним і техноцентричним підходами

Техноцентричний підхід фокусується на технології, ефективності, продуктивності або оптимізації алгоритмів. Метою такого підходу є

максимізація технічних характеристик системи, навіть якщо це ускладнює користування людиною. Користувач в подібних системах розглядається як об'єкт налаштування системи, а не її центр, менше враховуються когнітивні, емоційні чи соціальні аспекти.

Антропоцентричний підхід має фокус на людині, її потребах, здібностях, обмеженнях і цінностях, і має на меті створення систем, що пристосовуються до користувача, забезпечують зрозумілу, ефективну та безпечну взаємодію. Таким чином, він враховує когнітивні, емоційні та поведінкові аспекти користувача; адаптує інтерфейс і функціональність під індивідуальні та соціальні потреби; поєднує технічні рішення з етичними, соціальними та психологічними вимірами.

Визначення антропоцентричних обчислень (в англомовній науковій літературі – термін Human Centered Computing, НСС) вперше активно обговорювалось в 1997 р. на воркшопі з Human-Centered Systems, організатор Національний науковий фонд США (Flanagan J, Huang T, Jones P, Kasif S (1997) Human-centered systems: Information, interactivity, and intelligence. Tech. rep., NSF). Приклади визначень, які були дані за результатами воркшопу:

- НСС – «філософсько-гуманістична позиція щодо етики та естетики робочого місця»;
- система НСС – це «будь-яка система, яка покращує продуктивність людини»;
- система НСС – це «будь-яка система, яка відіграє будь-яку роль у посередництві взаємодії людей»;
- НСС – це «процес розробки програмного забезпечення, результатом якого є інтерфейси, дійсно зручні для користувача»;
- НСС – це «опис того, що робить хороший інструмент – комп'ютер виконує всю адаптацію»;
- НСС – це «міждисципліна, що розвивається, що потребує

інституціоналізації та спеціальних навчальних програм».

Інші визначення:

– НСС – це «наука про проектування обчислень і обчислювальних артефактів для підтримки людських зусиль» (Foley J (2008) HCC educational digital library. Tech. rep., <http://hcc.cc.gatech.edu>);

– НСС – це «бачення обчислювальних досліджень, яке поєднує технічні дослідження з широкими наслідками обчислювальної техніки в цілеспрямований спосіб. НСС охоплює інформатику та декілька інженерних дисциплін, когнітивну науку, економіку та соціальні науки» (Canny J (2001) Human-center computing. Tech. rep., UC Berkeley HCC Retreat);

– НСС – це «розробка, оцінка та розповсюдження технології, яка призначена для посилення та розширення людських можливостей до сприйняття, розуміння, міркування, вирішування та співпраці; проведення пізнавальної роботи; досягнення, підтримки та використання досвіду» (Hoffman R (2004) Human-centered computing principles for advanced decision architectures. Tech. rep., Army Research Laboratory).

Якщо розглядати ретроспективу НСС, то початкові кроки антропоцентричного мислення пов'язані з розвитком ергономіки й інженерної психології, що фокусувалися на зниженні людських помилок і підвищенні безпеки/ефективності інтерфейсів і систем управління. Цей період заклав методологічні підвалини – вимір людських можливостей, уваги та навантаження – які пізніше були інтегровані в НСІ (Human Computer Interaction). Становлення НСІ як окремої дисципліни супроводжувалося принципом *direct manipulation* [2] та працею Дональда Нормана з дизайну предметів повсякденного користування [1] – акцент на видимості, передбачуваності й зворотному зв'язку. Цей етап позначився переходом від адаптації людини до машини до проектування інтерфейсів, що відповідають людській ментальній моделі.

Розуміння взаємодії у ширшому соціальному контексті (групова робота, спільні практики, організаційні процеси) призвело до нової хвилі досліджень (Computer-Supported Cooperative Work, CSCW) і до уваги до того, як технології вбудовуються у соціальні практики. У цьому ключі почали враховуватися не лише індивідуальні когнітивні моделі, а й соціальні, культурні та організаційні фактори [3, 4]. Paul Dourish [3] і інші запропонували концепцію «embodied interaction» (втілена взаємодія), що підкреслює, як значення виникає в практичній діяльності, а не лише в абстрактних когнітивних процесах. Паралельно з'явилась ідея «третьої хвилі» HCI – увага до досвіду, емоцій, культури та повсякденності [5]. Ці підходи зблизили дизайн із філософією, соціологією й антропологією.

Останні десятиліття дали синтез наук: НСС об'єднує мультидисциплінарні методи для проєктування «з людиною в центрі», а в контексті штучного інтелекту (ШІ) розвиваються підходи пояснюваного та «відповідального» ШІ, що намагаються забезпечити прозорість, довіру і контроль [6]. Сучасні дослідження підкреслюють потребу у визначенні операційних принципів людиноорієнтованого штучного інтелекту (Human-centered AI, HCAI) та у стандартах реалізації людиноорієнтованих систем [7, 8, 9].

В той же час нова хвиля критики вказує на ризики незаперечного антропоцентризму: припущення про «бажаність» гібридизації людини й ШІ, ігнорування нерівностей, а також виклики щодо приватності та довіри [10]. Частина сучасних досліджень пропонує переосмислення: від антропоцентризму до більш комплексних перспектив (наприклад, пост-антропоцентричний дизайн [11], еко- та соціоорієнтовані підходи тощо [12, 13, 14]).

1.2. Людина як центральний елемент системи «людина–комп'ютер»

Коли ми говоримо про систему «людина–комп'ютер», важливо усвідомлювати, що фокусом є не просто інтерфейс чи технологія, а сама людина: її потреби, можливості, обмеження, контекст використання. Такі системи створюються «для людського використання», тому дизайн, розробка й оцінка мають враховувати когнітивні, фізичні, соціальні та емоційні аспекти користувача. З огляду на це, система, де людина є центральним елементом, повинна відповідати певним вимогам:

- людина задає цілі, завдання та отримує результати;
- система налаштовується під особливості людини: її знання, мотивацію, навички, контекст;
- взаємодія є двосторонньою: людина впливає на систему, система відповідає згідно з її потребами;
- ключовими якостями є корисність, ефективність, задоволеність, доступність з точки зору користувача системи.

Зважаючи на це, основні особливості побудови таких систем включають наступне:

1. Орієнтація на завдання та контекст людини.

Система повинна починатися з розуміння: що саме людина хоче зробити, в якому контексті, з яким рівнем знань та навичок. Це означає аналіз задач, які користувач вирішує, ситуацій використання, обмежень середовища.

2. Урахування когнітивних, сенсорних і моторних можливостей.

Людина має як сильні сторони, так і обмеження: обмежена пам'ять, увага, швидкість обробки інформації, моторні навички, сенсорні здібності. Тому система має бути спроектована так, щоб не навантажувати користувача зайвими складними кроками, щоб інформація подавалася у

зручній формі. Наприклад: чіткі візуальні сигнали, можливість контролю темпу роботи, адаптивність інтерфейсу під різні сенсорні режими (зір, слух), підтримка людей із обмеженими можливостями.

3. Інтерактивність та зворотній зв'язок.

Людина повинна отримувати адекватний зворотний зв'язок – як система реагує на її дії, яке її поточне становище, які вибори у неї є. Це дозволяє формувати коректну ментальну модель системи, відчувати контроль і впевненість. У класичній моделі взаємодії Donald A. Norman виділяв етапи:

- встановлення мети;
- формування наміру;
- вибір дій;
- виконання
- спостереження за результатом;
- інтерпретація;
- оцінка.

На практиці система повинна мати чіткі індикатори стану, зрозумілі повідомлення про помилки, підказки, адаптивну поведінку.

4. Адаптація та персоналізація.

Людина не є стандартною одиницею – існує велика варіативність: досвід, культурні особливості, мотивація, контекст. Системи мають бути здатні реагувати на ці відмінності, оскільки характеристики користувача (вік, досвід, уподобання тощо) суттєво впливають на взаємодію. Практично це може бути налаштування складності системи під досвід користувача, вибір модальностей (голос, текст, жести), можливість налаштування інтерфейсу, «умовна» допомога тощо.

5. Етичність, справедливість, доступність.

Оскільки людина у центрі, системи повинні враховувати етичні аспекти: приватність, безпеку, справедливість, доступність. На практиці це

може включати врахування особливостей людей із різними фізичними можливостями, з інших культур, зі специфічними потребами; тестування з реальними користувачами, що включають різні групи.

6. Інтеграція із системою як середовищем, а не лише інтерфейсом.

Коли людина є центральною частиною системи, важливим є весь контекст, а не тільки безпосередньо засоби взаємодії (клавіатура, екран і т.д.). Антропоцентричний підхід враховує середовище, завдання, соціальну взаємодію, організаційну культуру та інші аспекти. Таким чином, система розглядається не як ізольований інтерфейс конкретного користувача, а як частина великого соціо-технічного ландшафту: процеси роботи, навчання, підтримка, взаємодія людей між собою через систему.

7. Процес розробки за участі користувача.

Щоб забезпечити, що людина дійсно є у центрі, розробка системи має обов'язково включати користувача: дослідження потреб, прототипування, тестування із залученням кінцевих користувачів, ітераційне вдосконалення через використання інтерв'ю з користувачами, спостереження, сценаріїв, юзабіліті-тестування, А/В-тестів, збору якісного зворотного зв'язку. В HCI це реалізується з використанням дизайн-мислення, user-centred design (UCD).

З огляду на вищезазначене перед розробниками та організаціями постають додаткові виклики:

- потреба у досягненні балансу між автоматизацією/штучним інтелектом та контролем користувача: коли система автоматизує багато функцій, важливо, щоб людина мала механізми контролю, розуміла систему, довіряла їй;

- недооцінювання відмінностей в культурних, соціальних, індивідуальних та інших характеристиках користувачів: з одного боку в процесі розробки буває доволі складно і ресурсовитратно врахувати всі можливі підгрупи користувачів, що володіють схожими характеристиками,

з іншого боку – виділення занадто загальних підгруп цільової аудиторії може призвести до втрати специфічних сценаріїв при проєктуванні взаємодій;

- непередбачуваність дій користувача у складних системах;
- проблеми етики, приватності та довіри: антропоцентричний підхід передбачає не лише зручність користування, а й те, чи система дійсно підтримує інтереси людини, не експлуатує її і забезпечує справедливість, чи можуть довіряти користувачі рішенням систем, зокрема, при використанні ШІ-рішень.

Слід зауважити, що перевірка рівня антропоцентричності в реальних системах може бути доволі складною операцією. Для спрощення цього процесу в Додатку А наведено приклад чекліста, який можна використовувати в якості відправної точки при аналізі конкретної системи.

РОЗДІЛ 2. КОНЦЕПТУАЛЬНІ МОДЕЛІ АНТРОПОЦЕНТРИЧНИХ СИСТЕМ

Оскільки фундаментальною одиницею в НСС є не окремо людина, не окремо комп'ютер, система тощо, не окремо контекст та робоче середовище, а всі три одиниці одночасно, з цієї точки зору в рамках НСС при проєктуванні систем доцільно розглядати різні аспекти взаємодії:

- людина+людина;
- людина+програмне забезпечення;
- людина+апаратне забезпечення;
- людина+робоче середовище;
- комп'ютер+комп'ютер (наприклад, програмні агенти, якщо вони впливають на загальну продуктивність системи, призначеної для використання людиною).

Концептуальна модель будь-якої антропоцентричної системи повинна відображати, як технологічні рішення мають підтримувати когнітивні, поведінкові та соціальні аспекти людської діяльності, забезпечуючи гармонійне співіснування штучних і людських елементів системи. Інакше кажучи, це повинно бути узагальнене представлення структури, функцій і процесів взаємодії між людиною, комп'ютером та середовищем, у якому людина виступає центральним компонентом, що визначає цілі, контекст і критерії ефективності системи.

Далі наведено опис деяких моделей, які відображають сучасні підходи у НСІ, когнітивній ергономіці та системному дизайні, та приклади їх застосування. Слід зауважити, що обрана класифікація є відносно умовною, значна кількість дослідників в цій галузі може дотримуватись власного бачення щодо класифікації та визначати інші види моделей. Крім того, в сучасній розробці часто використовуються гібридні моделі, які

комбінують характеристики зазначених моделей або використовують інші, специфічні підходи.

2.1. Когнітивно-орієнтовані моделі

В основі когнітивно-орієнтованих моделей лежить розуміння того, як людина сприймає, запам'ятовує, приймає рішення та діє у взаємодії з системою. Такі моделі використовуються для побудови інтерфейсів, що відповідають мисленню користувача (mental congruence) і орієнтовані на когнітивні процеси (увага, пам'ять, мислення, навчання). Традиційне використання когнітивно-орієнтованих моделей – передбачення поведінки користувача, оцінки складності завдань, оптимізації робочих процесів.

Найбільш відомою з цього класу моделей є *модель GOMS* (Goals, Operators, Methods, Selection Rules), запропонована Кардом, Мораном і Ньюеллом в класичній праці [15]. Вона описує когнітивні процеси користувача під час взаємодії з комп'ютерною системою через структуру цілей, операцій, методів і правил вибору. Основна ідея полягає в тому, що користувач має певну мету (Goal), яку реалізує за допомогою базових дій або операцій (Operators), об'єднаних у методи (Methods); у ситуаціях, коли можливо кілька шляхів досягнення мети, застосовуються правила вибору (Selection rules). На відміну від техноцентричних моделей, які зосереджуються на продуктивності системи, GOMS спрямована на розуміння людської ефективності, часу виконання завдань, когнітивного навантаження та можливостей навчання користувача [16]. Модель GOMS відображає когнітивну природу діяльності людини та дозволяє розглядати користувача як активного агента, що приймає рішення, планує дії та оптимізує свої стратегії взаємодії з інтерфейсом. Подальші модифікації моделі, в тому числі такі як CPM-GOMS (Cognitive Perceptual Motor GOMS) і NGOMSL (Natural GOMS Language), розширили базову модель,

інтегрувавши часові характеристики виконання дій і принципи когнітивної архітектури людини. Так, CPM-GOMS дозволяє моделювати паралельні процеси сприйняття, мислення та моторики [17], що є особливо актуальним для сучасних систем з інтенсивною взаємодією користувача. CPM-GOMS дозволяє моделювати паралельне виконання дій, що є характерним для сучасних інтерфейсів AR/VR, «розумних» пристроїв або автопілотних систем. Для спеціалізованих систем, що використовують віртуальні середовища, існує модифікований метод оцінки цілей, операторів, методів та правил вибору (модель H-GOMS) [18], який дозволяє отримати кількісну оцінку системи взаємодії віртуальної руки у віртуальному середовищі.

Одним із різновидів моделі GOMS є KLM-GOMS (Keystroke-Level Model) [19], яка визначає для кожного елементарного оператора певний час виконання дії (таблиця 2.1). Застосування моделі передбачає розрахунок модельної оцінки, яку можна порівняти з реальним часом користувача під час тестування системи. Якщо фактичний час значно перевищує прогнозований, це свідчить про надмірне когнітивне або моторне навантаження, або ж неочевидність інтерфейсу. Таким чином, GOMS дає змогу кількісно оцінити ефективність дизайну і виявити етапи, де виникають затримки через неінтуїтивні елементи чи надмірні дії.

На практиці зазначені в таблиці 2.1 значення інтервалів часу можуть варіюватися в широких межах. Така змінюваність унеможливорює отримання абсолютних часових значень, проте застосування типових значень дозволяє здійснити об'єктивну порівняльну оцінку декількох реалізацій інтерфейсів. Складання послідовності дій K, H, P – завдання не складне, проте існують деякі труднощі з визначенням точок M, в яких користувач приймає несвідоме рішення про подальшу елементарну дію. Ядром моделі KLM-GOMS є правила, що дозволяють визначити, в яких моментах користувач затримається для ментальної підготовки до наступного кроку.

Таблиця 2.1 – Час виконання елементарних дій оператора в моделі KLM-GOMS

Оператор	Позначення	Середній час (с)	Коментар
Натискання клавiші або кнопки миші	К	0.2	базова моторна дія
Рух руки або миші до цілі	Р	1.1	залежить від відстані та розміру цілі
Ментальна підготовка (планування дії)	М	1.35	когнітивна пауза перед виконанням
Переміщення руки до миші або клавіатури	Н	0.4	якщо потрібно змінити пристрій введення
Час реакції комп'ютера	Р	-	визначається дослідним шляхом для кожного випадку індивідуально

Правила моделі KLM-GOMS:

Правило №1. Оператор М необхідно встановлювати перед усіма операторами К і Р, призначеними для вибору команд, але перед операторами Р, що вказують аргументи цих команд, оператор М ставити не слід.

Правило №2. Якщо оператор, наступний за оператором М, є повністю очікуваним з точки зору оператора, що передує М, то цей оператор М повинен бути знищений. При переміщенні ГПВ з наміром натиснути його кнопку після досягнення мети руху, відповідно до цього правила слід

видалити оператор M , встановлений за правилом 0: $P M K \rightarrow P K$.

Правило №3. У когнітивних одиницях слід видаляти всі оператори M за винятком першого. Когнітивною одиницею називається рядок виду $M K M K M K \dots$, безперервна послідовність введених символів, що утворюють в сукупності ім'я команди або її аргумент.

Правило №4. Якщо оператор K означає зайвий розділовий знак, який стоїть в кінці когнітивної одиниці (наприклад, роздільник команди, наступний відразу за роздільником аргументу цієї команди), то слід видалити оператор M , що стоїть перед ним.

Правило №5. Якщо оператор K є роздільником, що стоїть після константного рядку (назва команди або будь-яка інша послідовність символів, яка зберігає незмінний вид), то слід прибрати оператор M , що стоїть перед ним. Оскільки, в цьому випадку додавання роздільника стає частиною рядка і не вимагає спеціального оператора M . Однак, якщо оператор K є роздільником рядку аргументів або будь-якої іншої змінної рядку, то оператор M перед нею зберігається.

Правило №6. Частина оператора M , що перекривають оператор R , не враховуються.

Розглянемо для прикладу програму, призначену для перекладу десяткового числа в двійкове подання і навпаки. Інтерфейс програми представлений на рис.2.1. Користувач повинен встановити перемикач на відповідний варіант конвертації, після чого ввести число і натиснути клавішу Enter. Результат буде виведений в поле для пошуку, ніякі інші інструменти виведення не використовуються.

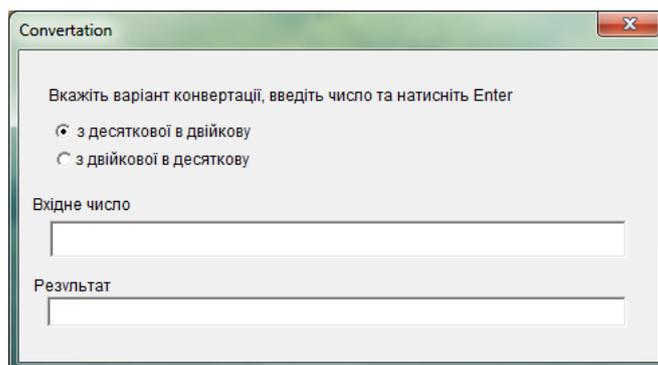


Рис.2.1. Інтерфейс програми для демонстрації моделі KLM-GOMS

Для спрощення завдання обмежимо вхідне число максимум чотирма розрядами, і домовимося, що необхідність переведення числа з десятичної системи в двійкову і навпаки виникають з однаковою ймовірністю. Складемо послідовність елементарних дій системи.

Переміщення руки до графічного пристрою введення (миша або тачпад): Н.

Переміщення курсору до необхідного варіанту конвертації: Н Р.

Натискання на відповідну опцію: Н Р К.

У половині випадків в програмі вже буде обраний потрібний варіант конвертації, що позбавить користувача необхідності переводити перемикач в потрібне положення. В даному прикладі розглядається протилежний варіант.

Користувач знову переміщує руки до клавіатури: Н Р К Н.

Вводить чотиризначне число: Н Р К Н К К К К.

Натискає Enter: Н Р К Н К К К К К.

Згідно з правилом №1 оператор М розставляються перед усіма операторами Р і К: Н М Р М К Н М К М К М К М К М К.

Замінімо послідовності Р М К на Р К, згідно з правилом №2: Н М Р К Н М К М К М К М К М К.

Правило №3 наказує видаляти оператор М, що стоїть в середині когнітивної одиниці: Н М Р К Н М К К К К М К.

Правило №5 наказує зберігати оператор М перед кінцевим К. В даному прикладі 4 і 6 правила не застосовуються. Підставивши замість символів операторів відповідні часові інтервали отримаємо: $H + M + P + K + H + M + K + K + K + K + M + K = 0.4 + 1.35 + 1.1 + 0.2 + 0.4 + 1.35 + 4*(0.2) + 1.35 + 0.2 = 7.15$ с.

Модель GOMS, хоч і є класичною, досі застосовується як аналітичний інструмент у людиноцентричному проєктуванні сучасних інтерфейсів [18, 20, 21, 22]. Але вона має як сфери ефективного використання, так і принципові обмеження, пов'язані з еволюцією взаємодії людини і технологій. У сучасних UI/UX-сценаріях GOMS частіше за все використовується для аналізу мікровзаємодій – дрібних кроків користувача при роботі з веб- або мобільними застосунками (наприклад, оформлення замовлення, надсилання повідомлення, зміна налаштувань). GOMS може також адаптуватися до мультимодальної взаємодії – наприклад, коли користувач одночасно використовує голосові команди, дотик та зоровий контроль. Один з практичних плюсів GOMS – можливість оцінювати дизайн на етапі проєктування, без потреби проводити користувацьке тестування. Це зручно у ранніх фазах UX-проєктування, коли треба швидко перевірити кілька варіантів сценаріїв і обрати найефективніший з погляду часу чи когнітивного навантаження.

Тим не менш, використання моделі GOMS має суттєві обмеження, які пов'язані з наступними факторами:

1. GOMS передбачає раціональну, послідовну поведінку користувача, яка **не враховує емоційні реакції, інтуїцію або спонтанні дії**. Реальні користувачі часто діють не оптимально, особливо в нових інтерфейсах чи під стресом.

2. GOMS, як детермінована модель, **не враховує динаміку середовища та контекстуальні підказки**, що особливо критично для систем, що мають змінний контекст, у якому кроки користувача не завжди

визначені наперед.

3. Побудова повної GOMS-моделі вимагає ретельного аналізу кожної альтернативної послідовності дій і правил вибору, що *стає громіздким у великих застосунках* з сотнями функцій. Через це GOMS частіше застосовується для локальних сценаріїв та мікрвзаємодій, а не для всієї системи.

4. Модель описує лише часові параметри та когнітивне навантаження, але *не оцінює естетичну, емоційну чи соціальну привабливість інтерфейсу* – компоненти, що безпосередньо визначають користувацький досвід.

2.2. Контекстуально-орієнтовані моделі

Контекстуально-орієнтовані моделі спрямовані на аналіз системної взаємодії, а не лише інтерфейсу. Людина розглядається як частина ширшої системи «людина–технологія–середовище», а поведінка користувача залежить від контексту: завдань, інструментів, соціальної взаємодії. Такі моделі дають змогу проектувати системи, що підтримують реальні робочі або життєві практики користувачів, і їх часто застосовуються в дизайні систем безпеки, промислових і навчальних середовищ.

Модель діяльності (Activity Theory) [23] базується на ідеї, що взаємодія людини з технологією відбувається в контексті її діяльності – з певною метою, мотивом і соціальними умовами. У центрі моделі не окремі дії користувача, а структура діяльності, де важливими є мотиви, інструменти, правила, спільнота і розподіл ролей. Модель допомагає виявити конфлікти між очікуваннями, правилами й інтерфейсом системи.

Методика застосування моделі діяльності в антропоцентричних системах включає наступні кроки:

1. Аналіз системи діяльності. Формується загальна структура діяльності: суб'єкт, об'єкт (мета), засоби (інструменти, інтерфейс), правила,

спільнота, розподіл ролей і очікуваний результат.

2. Збір емпіричних даних. Здійснюється спостереження за користувачами, проведення глибоких інтерв'ю, анкетування, аналіз робочих щоденників чи інші заходи, які дозволять дослідити реальний контекст виконання діяльності.

3. Аналіз суперечностей. Виявляються невідповідності між компонентами системи діяльності, наприклад, коли правила не узгоджуються з мотивами суб'єкта (користувача) або коли технічні засоби не підтримують досягнення мети.

4. Проектування змін. На основі виявлених суперечностей створюють оновлене дизайн-рішення, яке може включати зміну структури інтерфейсу, логіки навігації, системи мотивації, розподілу функцій тощо.

5. Оцінювання результатів. Перевіряється, чи усунуті суперечності, чи підвищилася узгодженість між діяльністю користувача та технічною системою. Для цього використовують аналітичні показники поведінки користувача, тестування (в тому числі, А/В-тестування), опитування користувачів тощо.

Розглянемо приклад практичного застосування моделі діяльності в контексті розробки системи дистанційного навчання для підтримки командної роботи студентів спеціальностей інформаційних технологій. Припустимо, що ми визначили в якості проблеми низький рівень взаємодії між студентами: попри групову форму навчання, більшість завдань виконується індивідуально. Тоді вміст основних кроків застосування моделі діяльності можна описати наступним чином.

1. Аналіз системи діяльності.

Суб'єкт: студент.

Об'єкт (мета): опанування навчального матеріалу та отримання оцінки.

Засоби: елементи інтерфейсу навчальної платформи (чат, спільний редактор, система завдань).

Правила: обов'язкове виконання частини завдань у групі.

Спільнота: студенти та викладач.

Розподіл ролей: лідер групи, учасники, викладач-координатор.

Результат: створення колективного навчального продукту.

2. Збір емпіричних даних для аналізу командної діяльності студентів.

Об'єкти збору даних:

1) активність користувачів у групових просторах:

- кількість створених і відредагованих спільних документів;
- кількість коментарів, повідомлень у чатах, постів на форумах;
- частота використання функцій спільного редагування.

2) розподіл ролей у команді:

– хто ініціює створення завдань, виконує технічні дії, подає результати;

– наскільки рівномірно розподіляються обов'язки між членами групи.

3) індивідуальна та колективна продуктивність:

- співвідношення індивідуально виконаних завдань до колективних;
- час виконання спільних проєктів;
- кількість корекцій або доопрацювань спільних результатів.

4) когнітивна та соціальна залученість:

– рівень участі у дискусіях, обмін аргументами, підтримка рішень інших членів команди;

– самооцінка студентами ефективності командної роботи (за визначеною шкалою).

Методи та технології збору даних можуть включати нижченаведені, але з урахуванням доступних технологій для конкретного випадку:

– автоматизований моніторинг активності (модуль аналітики LMS, результати логування активності користувачів: кількість входів до системи, час активності, використані функції, кількість переглядів, завантажень,

повідомлень, коментарів, часові інтервали взаємодії тощо);

- спостереження за виконанням групових завдань (запис екрану або спільного простору роботи; фіксація поведінкових патернів, рівня самостійності та ініціативи під час групових завдань; фіксація послідовності дій учасників: хто ініціює редагування, хто коментує, хто завершує роботу,);

- опитування студентів (для збору інформації можуть використовуватись засоби LMS або сторонні сервіси типу Google Forms, для кількісного вимірювання суб'єктивного досвіду користувачів доцільно застосувати адаптовані стандартизовані шкали: шкала оцінки зручності системи (System Usability Scale, SUS) [24], опитувальник досвіду користувача (User Experience Questionnaire, UEQ) [25], опитувальник якості командної роботи [26, 27, 28], шкала соціальної присутності (Social Presence Scale) [29, 30, 31]);

- аналіз комунікації (наприклад, через контент-аналіз повідомлень у чатах і форумах з класифікацією повідомлень за типами: ініціативні, інформаційні, кооперативні, оцінні, нейтральні тощо; результати можуть бути відображені у вигляді мережевої карти комунікацій між учасниками);

- глибинні інтерв'ю (проводяться зі студентами після завершення групових проєктів та мають на меті виявити суб'єктивні чинники низької взаємодії: невпевненість, конкуренція, нечіткий розподіл ролей; результати транскрибуються й кодуються за категоріями моделі діяльності (мотиви, цілі, інструменти, правила, спільнота, результат)).

Можливі кількісні критерії оцінювання результатів для виявлення суперечностей та подальшого проєктування змін:

- результати навчальної діяльності (середні чи абсолютні бали по завданням, по темам і т.п.);

- рівень взаємодії в команді – частка спільних дій, кількість коментарів і відповідей на повідомлення;

- баланс розподілу ролей – коефіцієнт рівномірності активності, для розрахунків може бути використаний індекс Джині [32];
- комунікативна щільність – середня кількість зв'язків між учасниками у мережевій моделі комунікацій;
- суб'єктивна задоволеність командною роботою – середній бал за шкалою UEQ/SUS;
- зміна когнітивної залученості – співвідношення між активністю в обговореннях і результативністю виконаних завдань.

3. Виявлення суперечностей (*розглядається лише одна із можливих гіпотез в рамках даного прикладу*).

Інтерфейс системи орієнтований переважно на індивідуальні дії: кожен студент має власний простір, що не сприяє спільній роботі. Внаслідок цього у студентів формується інший мотив – швидке виконання особистих завдань, а не досягнення колективної мети.

4. Проектування змін (*розглядаються лише деякі можливі пропозиції в рамках даного прикладу*).

Пропонується впровадження наступних рішень:

- спільні робочі дошки з відображенням внеску кожного учасника;
- додавання індикаторів колективного прогресу та відзнаки за командну активність;
- для викладачів – створення панелі моніторингу динаміки групової взаємодії.

5. Оцінювання результативності (*розглядаються лише деякі пропозиції щодо мети, критеріїв оцінювання та показників ефективності в рамках даного прикладу*).

Мета оцінювання – з'ясувати, наскільки оновлений інтерфейс сприяє досягненню навчальних і соціальних цілей користувачів. Приклади критеріїв оцінювання для даної задачі:

- рівень залученості користувачів – частота взаємодій у середовищі

системи (участь у спільних сесіях, використання інструментів комунікації);

- якість співпраці – узгодженість дій учасників групи, своєчасність виконання завдань, ступінь рівномірності внеску учасників;

- користувацьке сприйняття інтерфейсу – оцінка зручності, зрозумілості та задоволеності роботою з системою;

- досягнення навчальних результатів – порівняння динаміки виконання завдань і рівня успішності до та після впровадження змін.

Очікувані показники ефективності:

- збільшення кількості взаємодій між користувачами (зокрема у групових формах роботи);

- зростання частоти використання інструментів колективної діяльності;

- підвищення середніх оцінок за спільні проекти;

- позитивна динаміка у відповідях користувачів щодо зрозумілості та мотиваційного потенціалу системи.

Приклади опитувальників та шкал, які можуть бути використані для збору інформації про взаємодію системи з користувачами, наведено в додатку Б.

2.3. Адаптивно-поведінкові моделі

Основна ідея адаптивно-поведінкових моделей полягає у тому, що поведінка користувача є динамічною і змінюється під впливом отриманого зворотного зв'язку, навчання, досвіду та умов навколишнього середовища. Адаптивно-поведінкові моделі розглядають користувача не як пасивний об'єкт, а як активного суб'єкта, чия поведінка постійно модифікується взаємодією з системою та контекстом діяльності.

Концептуально ці моделі можуть інтегрувати кілька ключових компонентів:

1. Стани користувача – емоційні, когнітивні та фізіологічні аспекти, які впливають на вибір дій у системі.

2. Ситуаційні умови – зовнішні фактори середовища або контексту, що визначають поведінку користувача (наприклад, доступність інструментів, обмеження часу, соціальна взаємодія).

3. Правила адаптації – механізми, за допомогою яких система або сам користувач змінює стратегії поведінки у відповідь на успіх або невдачу попередніх дій.

4. Мотиваційні чинники – інтерес, цілі та внутрішні стимули користувача, що визначають пріоритетність вибору певних дій.

Практична цінність адаптивно-поведінкових моделей полягає у їхній здатності передбачати варіанти поведінки користувачів та пропонувати адаптації інтерфейсу або функціональних елементів системи для підвищення ефективності діяльності. Такі моделі дозволяють, зокрема, прогнозувати ймовірність того, що користувач обере певну послідовність дій, відреагує на інформаційні підказки або змінить стратегію у відповідь на нові умови.

Наприклад, для задачі персоналізації навчального контенту залежно від рівня підготовки, когнітивного навантаження та мотивації здобувача освіти при роботі в адаптивних навчальних системах та інтелектуальних навчальних середовищах модель компоненти моделі можуть визначатись наступним чином:

- стани користувача: рівень розуміння матеріалу, втома, залученість;
- ситуаційні умови: тип пристрою, час навчання, формат занять (самостійно / у групі);
- правила адаптації: зміна складності завдань, темпу подачі матеріалу, типу пояснень;

– мотиваційні чинники: навчальні цілі, оцінювання, внутрішня мотивація до опанування теми.

Очікуваний ефект від застосування такої моделі може полягати у підвищенні ефективності навчання та зменшенні когнітивного перевантаження.

При адаптації інтерфейсу до поточної поведінки та контексту використання користувача в інтерактивних інформаційних системах та користувацьких інтерфейсах очікуваний ефект може полягати, наприклад, в покращенні зручності використання та зниженні кількості помилок. Для цього адаптивно-поведінкова модель може включати:

- стани користувача: розгубленість, впевненість, досвід взаємодії;
- ситуаційні умови: середовище використання, доступність функцій, час виконання операцій;
- правила адаптації: приховування або спрощення елементів інтерфейсу, контекстні підказки;
- мотиваційні чинники: прагнення швидко досягти результату, мінімізація зусиль.

В загальному випадку адаптивно-поведінкову модель взаємодії користувача з інформаційною системою доцільно формалізувати як динамічну систему, стан якої змінюється в часі під впливом внутрішніх і зовнішніх чинників.

Нехай у дискретні моменти часу $t = 1, 2, \dots, T$ поведінка користувача описується сукупністю змінних, що відображають його стан, контекст діяльності та мотиваційні характеристики.

Стан користувача в момент часу t може бути представлений у вигляді вектора:

$$S_t = \langle s_t^1, s_t^2, \dots, s_t^n \rangle,$$

де компоненти s_t^i відповідають емоційним, когнітивним або фізіологічним аспектам (наприклад, рівень уваги, напруження,

залученість). Частина цих змінних є безпосередньо спостережуваною, тоді як інші доцільно розглядати як латентні та оцінювати опосередковано. До безпосередньо спостережуваних та вимірюваних величин відносяться ті параметри, які можуть бути зафіксовані без інтерпретаційних припущень шляхом реєстрації подій, сенсорних вимірювань або логування взаємодії. Ці величини є об'єктивними, можуть бути виміряні повторно та не залежать від інтерпретації дослідника. Приклади таких величин наведені на рис.2.2.

Фізіологічні стани	Поведінкові стани	Операційні стани
<ul style="list-style-type: none"> • Рівень фізіологічної активації (визначається за частотою серцевих скорочень, електропровідністю шкіри). • Рівень рухової активності (вимірюється акселерометрами, кількістю рухів). • Стан зорової активності (частота кліпання, фіксації погляду). 	<ul style="list-style-type: none"> • Стан активної взаємодії (користувач регулярно виконує дії, переходи між елементами). • Стан бездіяльності (відсутність дій протягом визначеного інтервалу часу). • Стан інтенсивної діяльності (висока частота дій, короткі інтервали між ними). 	<ul style="list-style-type: none"> • Стан виконання завдання (користувач перебуває в межах сценарію розв'язання задачі). • Стан помилки (зафіксовано помилкову дію або некоректне завершення кроку).

Рис.2.2. Приклади безпосередньо вимірюваних параметрів станів користувача в адаптивно-поведінкової моделі

Латентні величини не піддаються прямому вимірюванню, оскільки вони описують внутрішні стани або суб'єктивні характеристики користувача. Їх значення оцінюються на основі моделей, індикаторів або узагальнення спостережуваних даних. Опосередковане визначення латентних величин зазвичай ґрунтується на комбінації кількох спостережуваних індикаторів. Наприклад когнітивне навантаження може оцінюватися за часом виконання, кількістю помилок та частотою пауз;

рівень залученості – за тривалістю сесії, регулярністю взаємодії та різноманітням дій; фрустрація – за зростанням кількості повторних спроб, різкими змінами поведінки або припиненням взаємодії (рис.2.3).

Когнітивні стани	Емоційні стани	Мотиваційні стани
<ul style="list-style-type: none"> • Рівень уваги (оцінюється за стабільністю дій, кількістю відволікань, швидкістю реакцій). • Когнітивне навантаження (визначається за поєднанням часу виконання, кількості помилок і пауз). • Рівень засвоєння інформації (оцінюється за результатами контрольних дій або тестів). 	<ul style="list-style-type: none"> • Фрустрація (інтерпретується за повторюваними помилками, різкими змінами поведінки). • Зацікавленість (оцінюється за тривалістю взаємодії та добровільним продовженням роботи). • Стрес (визначається на основі поєднання фізіологічних і поведінкових індикаторів). 	<ul style="list-style-type: none"> • Рівень залученості (узагальнений показник активності, регулярності та різноманітності дій). • Орієнтація на досягнення результату (оцінюється за послідовністю цілеспрямованих дій). • Намір продовжувати взаємодію (виводиться з патернів повернення до системи).

Рис.2.3. Приклади латентних параметрів станів користувача в адаптивно-поведінкової моделі

У формальній моделі такі стани зазвичай вводяться як латентні змінні, параметри яких оцінюються за допомогою статистичних, імовірнісних або навчальних методів.

Ситуаційні умови задаються контекстним вектором:

$$C_t = \langle c_t^1, c_t^2, \dots, c_t^m \rangle,$$

який включає параметри середовища взаємодії, обмеження задачі та технічні умови (час, пристрій, доступні ресурси).

Контекстний вектор C_t описує зовнішні щодо користувача умови, у яких відбувається взаємодія з системою в момент часу t . На відміну від вектора стану користувача S_t , який відображає внутрішні характеристики, C_t :

- не є властивістю користувача як такої;
- може змінюватися незалежно від поведінки користувача;

– суттєво впливає на інтерпретацію дій та вибір адаптивних стратегій.

Ситуаційні умови, формалізовані через контекстний вектор C_t , є необхідним компонентом адаптивно-поведінкових моделей, оскільки саме вони задають рамки, у межах яких поведінка користувача набуває конкретного змісту та інтерпретації. Доцільно розглядати C_t як композицію кількох підконтекстів, кожен з яких відповідає окремій групі ситуаційних умов. Приклади компонентів:

– часовий контекст (одиниці вимірювання: безперервні – секунди, хвилини; дискретні – ранок, день, вечір тощо): c_t^1 час доби; c_t^2 тривалість поточної сесії; c_t^3 залишковий час на виконання завдання;

– технічний контекст (типи даних – категоріальні, номінальні): c_t^4 тип пристрою (настільний комп'ютер, мобільний пристрій); c_t^5 спосіб введення (клавіатура, сенсорний екран); c_t^6 характеристики екрана (розмір, роздільна здатність);

– контекст задачі (типи даних – дискретні, порядкові): c_t^7 тип завдання (навчальне, пошукове, аналітичне); c_t^8 рівень складності завдання; c_t^9 етап виконання завдання;

– контекст доступних ресурсів (типи даних – булеві, чисельні): c_t^{10} : доступність підказок або довідкових матеріалів; c_t^{11} можливість повернення до попередніх кроків; c_t^{12} кількість дозволених спроб;

– соціальний та організаційний контекст (типи даних – категоріальні, бінарні): c_t^{13} індивідуальна або групова взаємодія; c_t^{14} наявність зовнішнього контролю або оцінювання; c_t^{15} регламентованість процесу.

Мотиваційні чинники в адаптивно-поведінкових моделях доцільно формалізувати через функцію корисності, яка агрегує вплив стану користувача, ситуаційних умов і допустимих дій. Такий підхід дозволяє явно задавати пріоритети, виконувати кількісне порівняння альтернатив і

реалізовувати адаптацію як задачу оптимального вибору дії в динамічному середовищі. Функція корисності формалізує переваги користувача або системи щодо можливих дій у конкретному стані:

$$U_t = U(S_t, C_t, A_t),$$

де $A_t = \{a_1, a_2, \dots, a_k\}$ множина можливих дій користувача або системи. Значення функції корисності відображає пріоритетність вибору дій у конкретному стані. U_t представляє собою числову оцінку доцільності вибору дії a_i . Мотивація в цій постановці не є окремою змінною, а проявляється через відносні значення корисності для різних дій. На практиці функцію корисності доцільно подавати як зважену суму окремих компонентів.

Розглянемо до прикладу адаптивну навчальну систему. Множину можливих дій системи визначимо вектором

$$A_t = \{a_1, a_2, a_3\},$$

де a_1 – подати спрощене пояснення навчального матеріалу; a_2 – подати стандартний матеріал; a_3 – запропонувати ускладнене завдання.

Вектор стану та контексту представимо як набір латентних величин:

$$S_t = \langle \hat{Z}, \hat{N} \rangle,$$

$$C_t = \langle T \rangle,$$

де $\hat{Z} \in [0, 1]$ – рівень засвоєння матеріалу; $\hat{N} \in [0, 1]$ – когнітивне навантаження, T – часовий контекст, $T \in [0, 1]$, 1- жорстке обмеження.

Введемо три мотиваційні критерії.

Критерій 1 – досягнення навчальної цілі:

$$u_1(a) = \begin{cases} 1 - \hat{Z}, & a = a_1 \\ 0.5, & a = a_2 \\ \hat{Z}, & a = a_3 \end{cases}.$$

Критерій 2 – мінімізація когнітивного навантаження:

$$u_2(a) = \begin{cases} 1 - \hat{N}, a = a_1 \\ 0.5, a = a_2 \\ 1 - 2\hat{N}, a = a_3 \end{cases}.$$

Критерій 3 – урахування часових обмежень:

$$u_2(a) = \begin{cases} 1 - T, a = a_1 \\ 0.5, a = a_2 \\ T, a = a_3 \end{cases}.$$

Нехай у момент часу t маємо наступний стан та контекст:

- $\hat{Z} = 0.4$, низький рівень засвоєння,
- $\hat{N} = 0.7$, високе навантаження,
- $T = 0.8$, жорсткі часові обмеження.

Визначимо функцію корисності через зважену адитивну модель:

$$U_t(a) = \sum_{j=1}^3 w_j \cdot u_j(S_t, C_t, a),$$

де w_j – вагові коефіцієнти кожного з критеріїв.

Визначимо значення вагових коефіцієнтів $w_1 = 0.5, w_2 = 0.3, w_3 =$

0.2. Тоді, величини корисності приймуть значення:

$$U_t(a_1) = 0.5(1 - 0.4) + 0.3(1 - 0.7) + 0.2(1 - 0.8) = 0.5(0.6) + \\ 0.3(0.3) + 0.2(0.2) = 0.43,$$

$$U_t(a_2) = 0.5(0.5) + 0.3(0.5) + 0.2(0.5) = 0.5,$$

$$U_t(a_3) = 0.5(0.4) + 0.3(1 - 1.4) + 0.2(0.8) = 0.2 - 0.12 + 0.16 = 0.24,$$

$$U_t(a_2) > U_t(a_1) > U_t(a_3).$$

Отже, оптимальною для даних параметрів є дія a_2 – подання стандартного матеріалу. Це означає, що хоча рівень засвоєння низький, високе когнітивне навантаження та жорсткі часові обмеження зміщують мотиваційний баланс у бік компромісного сценарію.

Адаптація системи реалізується через правило переходу:

$$S_{t+1} = f(S_t, C_t, a_t, r_t),$$

де a_t – застосована адаптивна дія системи в момент часу t ; r_t зворотний зв'язок, що відображає успішність або неуспішність попередньої

взаємодії; функція f описує механізм еволюції стану користувача.

Найпростіший варіант інтерпретації функції f – явна формула оновлення стану:

$$S_{t+1} = S_t + \Delta(S_t, C_t, a_t, r_t),$$

де Δ – правило зміни стану. Наприклад: рівень засвоєння зростає, якщо дія була успішною; когнітивне навантаження зменшується після спрощення матеріалу. Такий підхід добре підходить для методичних або пояснювальних моделей, але потребує явних гіпотез про причинно-наслідкові зв'язки.

Функція f може реалізовуватись як набір умовних правил виду:

Якщо (умови), то (оновлення стану).

Формально:

$$f = \{(умова_i \rightarrow перехід_i)\}.$$

Цей підхід є доволі легко інтерпретованим людиною, він широко поширений у НСІ та експертних системах та не потребує великих обсягів даних.

У більш загальному випадку f задає ймовірнісний перехід:

$$P(S_{t+1} | (S_t, C_t, a_t, r_t)).$$

Тобто стан у момент $t + 1$ не визначений однозначно, а описується розподілом імовірностей. Отже, одна й та сама дія може мати різний ефект, а внутрішній стан користувача змінюється з невизначеністю. Такий підхід характерний для марковських моделей, байєсівських мереж або підкріплювального навчання.

Функція f може бути моделлю, параметри якої навчаються:

$$S_{t+1} = f_{\theta}(S_t, C_t, a_t, r_t),$$

де θ – параметри моделі, яка може бути задана як лінійна або нелінійна регресія, нейронна мережа, рекурентна модель, байєсівський оцінювач стану тощо. У цьому випадку f не задається явно, вона

апроксимує реальну динаміку поведінки.

Величина r_t відображає оцінку результату взаємодії після застосування дії a_t . Вона може мати різну природу залежно від системи.

Дискретний зворотний зв'язок може бути визначений як

$$r_t \in \{-1, 0, +1\}.$$

Значення $+1$ визначає успіх (правильне виконання, позитивний результат); 0 – нейтральний результат; -1 – невдача або помилка. Такий підхід часто застосовується в навчальних системах, тренувальних симуляціях, ігрових середовищах.

Неперервний числовий зворотний зв'язок

$$r_t \in \mathbb{R} \text{ або } r_t \in [0,1]$$

дозволяє тонку адаптацію та широко використовується в оптимізаційних моделях. Прикладами такого типу зв'язку можуть бути частка правильних відповідей, нормалізований показник ефективності, зважена оцінка успішності дії.

Векторний зворотний зв'язок

$$r_t = \langle r_t^1, r_t^2, \dots, r_t^k \rangle$$

дозволяє багатокритеріальну оцінку та використовується разом із функцією корисності.

У деяких системах r_t не спостерігається напряму, а оцінюється:

$$r_t = g(\text{поведінкові індикатори}).$$

В цьому випадку величина r_t - це оцінка якості результату, а не прямий сигнал. Приклади індикаторів: припинення взаємодії; повторювані помилки; різке зниження активності.

Розглянемо застосування наведеної формальної моделі в адаптивній електронній навчальній системі, призначеній для індивідуалізованого опанування навчального матеріалу.

У цьому випадку вектор стану користувача може бути визначений як:

$$S_t = \langle \text{рівень засвоєння, когнітивне навантаження, залученість} \rangle.$$

Рівень засвоєння оцінюється на основі результатів тестових завдань, когнітивне навантаження – за непрямими поведінковими ознаками (час виконання, кількість помилок), а залученість – через частоту та регулярність взаємодії.

Контекстний вектор включає параметри навчальної ситуації:

$$C_t = \langle \text{тип контенту, час навчання, пристрій} \rangle.$$

Ці дані фіксуються системою автоматично та використовуються для вибору відповідного режиму подання матеріалу.

Мотиваційні чинники задаються у вигляді цільової функції, що відображає прагнення користувача до досягнення навчального результату за мінімальних витрат часу та зусиль. Наприклад, система може максимізувати ймовірність успішного виконання наступного завдання за умови збереження прийняттого рівня когнітивного навантаження.

Правила адаптації визначають, яким чином система змінює навчальний сценарій. Якщо виявляється зниження рівня засвоєння та зростання когнітивного навантаження, функція переходу переводить систему в стан, у якому пропонуються спрощені пояснення або додаткові приклади. За умови стабільного успіху та високої залученості система, навпаки, підвищує складність матеріалу або темп навчання.

Таким чином, адаптивна навчальна система реалізує формальну модель як замкнений цикл взаємодії, у якому стан користувача, контекст діяльності та правила адаптації безперервно уточнюються на основі накопиченого зворотного зв'язку. Це забезпечує перехід від статичного навчального процесу до динамічного, орієнтованого на індивідуальні особливості користувача.

РОЗДІЛ 3. ПІДХІД HUMAN-IN-THE-LOOP

3.1. Загальні положення підходу Human-in-the-Loop

Human-in-the-Loop (HITL) [35] – це загальний підхід до проєктування складних систем (штучний інтелект, автоматизоване прийняття рішень, робототехніка, інтерфейси людина-машина), де людина не просто кінцевий споживач результатів, а активний елемент циклу обробки, який взаємодіє з системою, впливає на її параметри та результат.

Основні тенденції у цій галузі зосереджуються на кількох ключових аспектах:

- 1) інтеграція когнітивних моделей людини в алгоритми: як враховувати обмеження уваги, пам'яті, ухвалення рішень людиною; як моделювати людську поведінку та помилки для підвищення надійності систем; як адаптувати автоматичні алгоритми до людських патернів взаємодії;
- 2) спільне навчання систем та людей (дозволяє системам «вчитися» разом з користувачем і зменшувати кількість помилок): бек-енд навчання моделей з урахуванням зворотного зв'язку від людини; пояснювані AI, де модель не лише видає результат, але й формулює пояснення для людини; активне навчання з людиною в циклі, коли система запитує людину саме в тих випадках, де автоматична впевненість низька;
- 3) дизайн інтерфейсів та людино-машинна взаємодія: візуальне подання інформації; інтерактивні протоколи для коригування дій системи людиною; оцінювання ухвалення рішень людиною у реальному часі; розробка feedback loops, які дозволяють людині зрозуміти не лише що відбувається, а й чому;
- 4) етичні, правові та безпекові аспекти: відповідальність за рішення, прийняті з участю автоматизованих систем; обмеження використання HITL у критичних застосуваннях (охорона

здоров'я, транспорт, оборона); збереження приватності та безпеки персональних даних.

3.2. Етапи проєктування систем на основі підходу Human-in-the-Loop

Узагальнений алгоритм проєктування системи, яка ґрунтується на підході HITL можна представити схемою на рис.2.4.

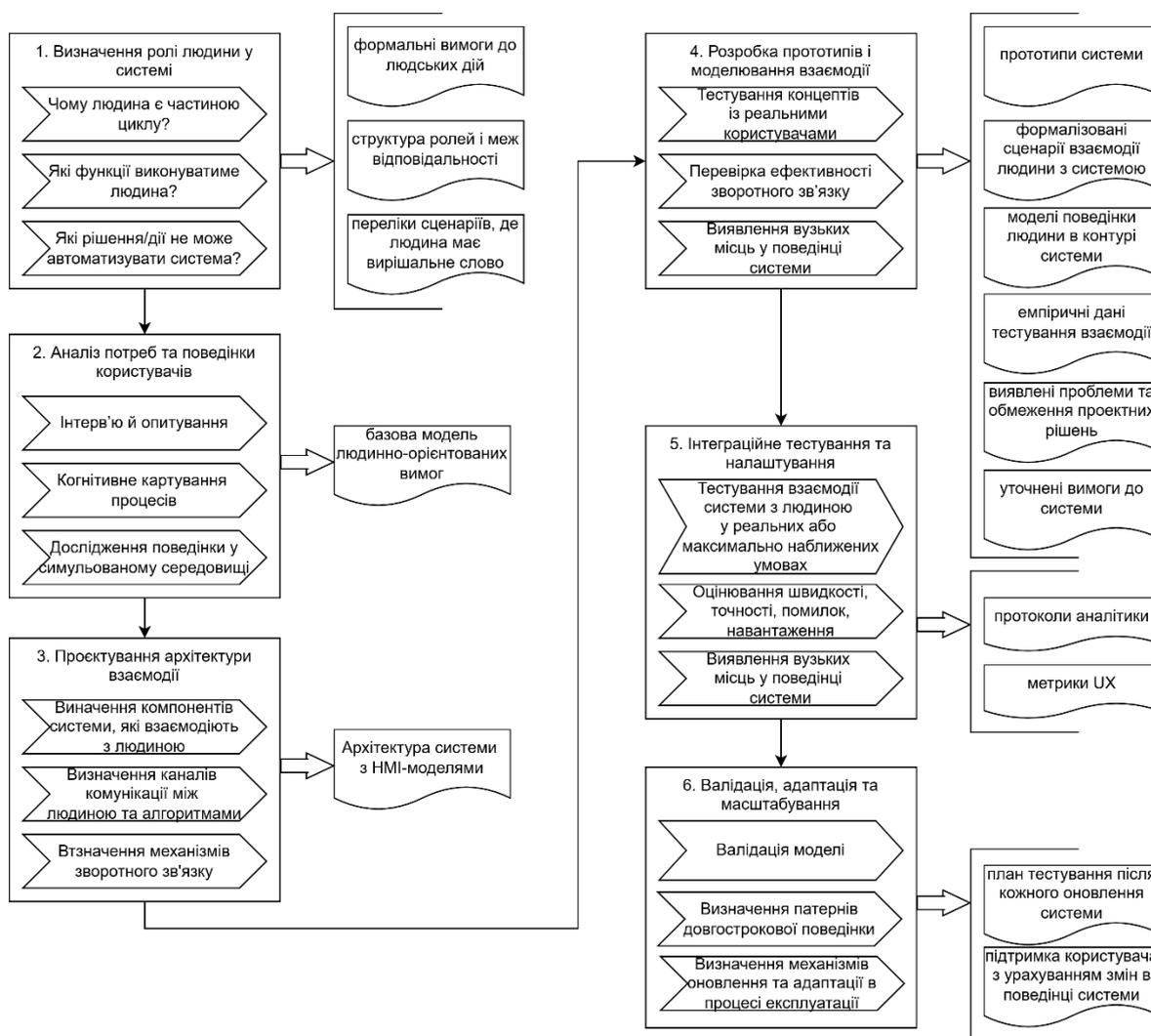


Рис. 2.4. Етапи проєктування системи на основі підходу HITL

3.2.1. Визначення ролі людини у системі

Етап визначення ролі людини у системі є базовим концептуальним етапом проектування комп'ютерних та інформаційних систем із залученням людини до циклу функціонування. Саме на цьому етапі формується розуміння того, яким чином людський фактор інтегрується в систему, а також встановлюються межі автоматизації та відповідальності між людиною і технічними компонентами. На відміну від традиційних автоматизованих систем, де людина розглядається переважно як зовнішній користувач, у human-in-the-loop підході людина є структурним елементом системи, що безпосередньо впливає на її поведінку, результати роботи та безпеку. Етап визначення ролі людини у системі є методологічно визначальним для всього процесу проектування human-in-the-loop-систем. Помилки або нечіткість на цьому етапі призводять до системних проблем, які неможливо повністю усунути на пізніших стадіях розробки, незалежно від якості реалізації алгоритмів чи інтерфейсів. Метою етапу є формалізоване визначення функціональної ролі людини в межах системи, зокрема:

- обґрунтування необхідності участі людини у циклі обробки інформації або прийняття рішень;
- встановлення рівня та характеру цієї участі;
- запобігання неконтрольованому перекладанню критичних функцій або відповідальності між людиною і автоматизованими модулями.

Необхідність цього етапу зумовлена кількома принциповими чинниками:

1. Обмеженість автоматизованих алгоритмів.

Навіть сучасні алгоритмічні та інтелектуальні системи мають обмеження, пов'язані з неповнотою даних, неоднозначністю ситуацій, відсутністю контексту або етичними аспектами прийняття рішень. Людина

вводиться в систему саме в тих точках, де автоматизація є недостатньою або небезпечною.

2. Наявність людської відповідальності.

У багатьох прикладних доменах (медицина, транспорт, управління, безпека) остаточна відповідальність за рішення не може бути повністю делегована машині. Це потребує чіткого визначення, які рішення є рекомендаційними, а які – остаточними.

3. Запобігання конфліктам між людиною і системою.

Без чіткого визначення ролей система може або надмірно обмежувати дії людини, або, навпаки, перекладати на неї надмірну кількість рутинних чи критичних операцій, що призводить до зростання помилок і когнітивного навантаження.

На цьому етапі послідовно аналізуються такі аспекти.

1. Обґрунтування включення людини до контуру системи.

Першочергово визначається, чому система не може бути повністю автономною. Аналізуються:

- типи завдань, що потребують людського судження;
- ситуації невизначеності або виняткові сценарії;
- етичні, правові або безпекові обмеження автоматизації.

Це дозволяє уникнути формального або декларативного використання підходу human-in-the-loop.

2. Визначення функцій, які виконує людина.

Конкретизуються функції людини у системі, наприклад:

- контроль та нагляд за автоматизованими процесами;
- підтвердження або відхилення рішень системи;
- коригування параметрів роботи алгоритмів;
- прийняття рішень у критичних або нестандартних ситуаціях.

Важливо, що ці функції формулюються не в термінах інтерфейсу, а в термінах системної ролі.

3. Визначення меж автоматизації.

На цьому кроці встановлюється, які дії система може виконувати автономно, а які – лише за участю людини. Формується розподіл:

- повністю автоматизованих процесів;
- напівавтоматизованих процесів із контролем людини;
- процесів, де людина є основним суб'єктом прийняття рішень.

Такий розподіл є критично важливим для подальшого проектування архітектури та алгоритмів.

У результаті виконання етапу отримують:

- формалізовану структуру ролей та відповідальності, яка визначає взаємодію людини й технічних компонентів;
- чітко сформульовані вимоги до людських дій, зокрема часові, когнітивні та процедурні обмеження;
- перелік сценаріїв обов'язкового втручання людини, включно з критичними та аварійними ситуаціями.

3.2.2 Аналіз потреб та поведінки користувачів

На етапі аналізу потреб та поведінки користувачів система починає розглядатися не абстрактно, а в контексті реальних можливостей, обмежень і поведінкових характеристик людини, яка з нею взаємодіє. У підході human-in-the-loop цей етап має принципове значення, оскільки помилки в розумінні потреб або поведінки людини безпосередньо трансформуються в системні помилки, які не можуть бути компенсовані лише технічними засобами. Метою етапу є формування обґрунтованої моделі користувача, що описує:

- функціональні потреби людини у взаємодії з системою;
- когнітивні, фізичні та часові обмеження;
- типові поведінкові патерни у штатних та нестандартних ситуаціях;

– очікування людини щодо реакцій, пояснень і рівня контролю з боку системи.

Ця модель слугує основою для подальшого проектування архітектури взаємодії, інтерфейсів та алгоритмів адаптації. Необхідність аналізу потреб і поведінки користувачів зумовлена низкою фундаментальних чинників. По-перше, людина не є детермінованим елементом системи. Її реакції змінюються залежно від досвіду, навантаження, контексту та рівня довіри до системи. Без урахування цієї варіативності система може демонструвати формальну коректність, але практичну неефективність. По-друге, людські помилки мають системний характер. Вони часто зумовлені не неухважністю, а невідповідністю між логікою системи та ментальною моделлю користувача. Аналіз поведінки дозволяє виявити ці розбіжності на ранньому етапі. По-третє, в НІТЛ-системах людина часто працює в умовах обмеженого часу, інформаційного перевантаження або стресу, що принципово відрізняє такі системи від класичних інформаційних сервісів.

Етап аналізу потреб та поведінки користувачів реалізується як поєднання якісних та кількісних методів аналізу.

1. Ідентифікація цільових груп користувачів.

Спочатку визначаються:

- категорії користувачів;
- рівні їхньої підготовки та досвіду;
- ролі у взаємодії з системою (оператор, контролер, експерт, спостерігач).

Це дозволяє уникнути усередненого образу користувача та врахувати різні сценарії використання.

2. Аналіз завдань і цілей користувачів

Досліджуються:

- типові завдання, які користувач виконує за допомогою системи;
- частота та критичність цих завдань;

- наслідки помилок або затримок у виконанні.

У контексті human-in-the-loop важливо не лише що робить користувач, а у який момент і за яких умов система потребує його участі.

3. Дослідження когнітивних та фізичних обмежень.

На цьому кроці аналізуються:

- обсяг інформації, який людина здатна ефективно обробляти;
- часові характеристики реакції;
- втомлюваність, стресові фактори, здатність до багатозадачності.

Ці параметри безпосередньо впливають на допустимі режими взаємодії та рівні автоматизації.

4. Вивчення поведінкових патернів.

Особлива увага приділяється:

- типовим діям користувача у стандартних ситуаціях;
- поведінці у нестандартних або критичних умовах;
- схильності до довіри або недовіри до автоматизованих рішень.

Аналіз цих патернів дозволяє передбачити, як людина реально взаємодіятиме із системою, а не як це передбачено проектною документацією.

5. Формування людино-орієнтованих вимог.

На завершальному кроці отримані дані узагальнюються у вигляді:

- функціональних вимог до взаємодії;
- нефункціональних вимог (зручність, зрозумілість, навантаження);
- обмежень, які система не повинна перевищувати.

Ці вимоги формуються з позиції людини як елемента системи, а не лише як користувача інтерфейсу.

Результатом етапу є формалізована модель користувача, яка включає:

- опис ролей і контекстів використання;
- поведінкові та когнітивні характеристики;
- обґрунтовані людино-орієнтовані вимоги до системи.

3.2.3. Проєктування архітектури взаємодії

Етап проєктування архітектури взаємодії є системоутворювальним інженерним етапом, на якому концептуальні уявлення про роль людини та результати аналізу її потреб трансформуються у формалізовану структуру взаємодії між людиною, алгоритмами та технічними компонентами системи. У системах human-in-the-loop архітектура взаємодії визначає не лише технічну організацію компонентів, але й логіку розподілу ініціативи, контролю та відповідальності між людиною і автоматизованими модулями. Метою цього етапу є побудова узгодженої архітектурної моделі взаємодії, яка:

- забезпечує коректне включення людини в контур функціонування системи;
- визначає способи обміну інформацією між людиною та алгоритмічними компонентами;
- гарантує керованість, зрозумілість і безпеку спільної роботи людини й системи.

Фактично, на цьому етапі формується відповідь на запитання: як саме людина і система взаємодіють у процесі досягнення цілей, а не лише хто і що робить. Необхідність окремого етапу проєктування архітектури взаємодії зумовлена специфікою НІТЛ-систем. По-перше, людина і алгоритм оперують різними типами інформації. Алгоритми працюють з формалізованими даними та моделями, тоді як людина сприймає інформацію через інтерпретацію, контекст і попередній досвід. Архітектура взаємодії має забезпечити коректне узгодження цих різномірних інформаційних представлень. По-друге, участь людини не є постійною. Вона може бути епізодичною, умовною або критичною лише в окремих сценаріях. Без чіткої архітектури взаємодії система або перевантажує людину, або, навпаки, позбавляє її можливості втручання у важливі моменти. По-третє, у НІТЛ-системах принципово важливим є управління

ініціативою: коли рішення ініціює система, коли – людина, а коли ініціатива передається динамічно. Це неможливо забезпечити без явного архітектурного проектування.

Проектування архітектури взаємодії виконується як послідовний перехід від абстрактних ролей до конкретних структурних і функціональних рішень:

1. Формування структурної моделі взаємодії.

На першому кроці визначається:

- які компоненти системи безпосередньо взаємодіють з людиною;
- які модулі є повністю автоматизованими;
- які компоненти виконують роль посередників між людиною та алгоритмами (інтерфейсні, пояснювальні, контрольні модулі).

Людина на цьому етапі розглядається як логічний компонент архітектури, а не як зовнішній користувач. Це дозволяє формально описати її участь у загальному контурі управління.

2. Проектування каналів комунікації.

Визначаються канали обміну інформацією між людиною та системою:

- напрямки передавання інформації (від системи до людини та навпаки);
- типи сигналів і повідомлень;
- допустима частота, обсяг та форма подання інформації.

Особлива увага приділяється вибору модальностей введення та виведення (візуальна, аудіальна, тактильна), з урахуванням когнітивних обмежень та контексту використання системи. Архітектурні рішення на цьому кроці безпосередньо впливають на когнітивне навантаження та швидкість реакції людини.

3. Визначення механізмів зворотного зв'язку:

- способів інформування людини про стан системи;

- механізмів пояснення рішень алгоритмів;
- каналів підтвердження, корекції або відміни дій системи.

4. Визначення рівнів автоматизації.

На цьому кроці формалізується розподіл функцій між людиною і системою за рівнями автоматизації:

- повністю автоматизовані процеси;
- процеси з контролем або підтвердженням з боку людини;
- процеси, де людина є основним суб'єктом прийняття рішень.

Важливо, що ці рівні визначаються в контексті конкретних сценаріїв, а не для системи загалом, що дозволяє забезпечити гнучкість і адаптивність.

Результатом етапу проєктування архітектури взаємодії є:

- повністю описана архітектурна модель системи з урахуванням участі людини;
- формалізовані канали та протоколи взаємодії;
- визначені рівні автоматизації для різних сценаріїв;
- базові моделі людино-машинної взаємодії (НМІ), що слугують основою для прототипування.

3.2.4. Розробка прототипів і моделювання взаємодії

Етап розробки прототипів і моделювання взаємодії є перехідним етапом між архітектурним проєктуванням та інженерною реалізацією системи. На цьому етапі абстрактні архітектурні рішення, моделі ролей людини та вимоги до взаємодії переводяться у експериментально перевірювану форму, що дозволяє оцінити реальну працездатність human-in-the-loop-підходу до впровадження системи. У контексті НІТЛ саме на цьому етапі уперше перевіряється, наскільки коректно людина інтегрована в контур системи не концептуально, а операційно. На відміну від суто технічних систем, де прототипування спрямоване передусім на перевірку функціональності, у НІТЛ-системах основний фокус зосереджений на

взаємодії, зрозумілості, керованості та поведінкових ефектах. Метою етапу є емпірична перевірка та уточнення спроектованої взаємодії між людиною і системою шляхом створення прототипів та проведення моделювання реальних сценаріїв використання. Зокрема, цей етап спрямований на:

- тестування життєздатності концептів взаємодії з участю людини;
- перевірку ефективності каналів зворотного зв'язку;
- виявлення невідповідностей між проектною логікою системи та реальною поведінкою людини;
- отримання даних, необхідних для коригування архітектурних і алгоритмічних рішень.

Необхідність окремого етапу прототипування та моделювання взаємодії зумовлена специфікою людського фактору. По-перше, поведінка людини не може бути повністю виведена з аналітичних моделей. Навіть коректно побудовані моделі користувача не здатні передбачити всі варіанти реакцій, помилок або стратегій взаємодії. Прототипування дозволяє перейти від припущень до спостережуваних фактів. По-друге, у НІТЛ-системах існує ризик прихованих системних помилок, які не проявляються на рівні алгоритмів, але виникають у взаємодії (неоднозначні повідомлення, надмірна кількість підтверджень, несинхронність дій людини і системи). По-третє, саме на цьому етапі можна виявити непропорційний розподіл навантаження між людиною і системою, коли формально роль людини визначена коректно, але на практиці вона або перевантажена, або фактично усунена з процесу прийняття рішень.

Етап реалізується як ітеративний процес, що поєднує створення прототипів, моделювання взаємодії та аналіз отриманих результатів.

1. Розробка прототипів взаємодії

На першому кроці створюються:

- функціональні або нефункціональні прототипи інтерфейсу;
- експериментальні реалізації окремих сценаріїв взаємодії;

- прототипи різного рівня деталізації залежно від цілей дослідження.

Прототипи не обов'язково відображають повну функціональність системи. Їх основне призначення – відтворити ключові точки взаємодії людини і системи, у яких відбувається прийняття рішень, втручання або корекція.

2. Формалізація та відтворення сценаріїв взаємодії.

Реалізуються та відпрацьовуються:

- типові сценарії використання;
- критичні та нестандартні сценарії;
- сценарії обов'язкового втручання людини.

Ці сценарії використовуються як основа для експериментального тестування та дозволяють оцінити, чи відповідає фактична взаємодія спроектованим ролям і рівням автоматизації.

3. Моделювання поведінки людини в контурі системи

На цьому кроці застосовуються:

- експерименти з реальними користувачами;
- симулятори поведінки;
- псевдокористувачі або агентні моделі.

Моделювання дозволяє дослідити:

- часові характеристики реакцій людини;
- ймовірність помилок;
- варіативність поведінки в однакових умовах;
- вплив зворотного зв'язку на рішення людини.

4. Збір та аналіз емпіричних даних.

У процесі тестування накопичуються:

- кількісні показники (швидкість, точність, кількість помилок);
- якісні спостереження щодо зрозумілості та зручності взаємодії;
- суб'єктивні оцінки користувачів.

Ці дані є об'єктивною основою для подальших проектних рішень, на відміну від суто експертних припущень.

З урахуванням наведених вище процесів, результатами етапу є:

- функціональні та нефункціональні прототипи системи;
- формалізовані та перевірені сценарії взаємодії людини з системою;
- уточнені моделі поведінки людини в контурі системи;
- емпіричні дані щодо ефективності, надійності та зрозумілості взаємодії;
- виявлені проблеми та обмеження проектних рішень;
- уточнені функціональні та нефункціональні вимоги до системи.

3.2.5. Інтеграційне тестування та налаштування

Етап інтеграційного тестування та налаштування є критичним етапом переходу від експериментально перевіреної взаємодії до стабільної, придатної до експлуатації системи. На цьому етапі відбувається поєднання всіх компонентів системи – алгоритмічних, програмних, інтерфейсних та людського фактору – в єдиний функціональний контур. У контексті human-in-the-loop інтеграційне тестування принципово відрізняється від класичного інженерного тестування, оскільки об'єктом перевірки є не лише коректність взаємодії між технічними модулями, але й стійкість, передбачуваність та безпека системи за участю людини. Метою етапу є перевірка цілісної роботи системи в умовах, максимально наближених до реальної експлуатації, а також налаштування параметрів взаємодії між людиною і системою таким чином, щоб:

- забезпечити узгодженість дій усіх компонентів;
- мінімізувати негативний вплив людських помилок;
- підтвердити досягнення заданих показників ефективності, надійності та безпеки;

– виявити та усунути інтеграційні конфлікти, які не проявляються на рівні окремих модулів.

Необхідність інтеграційного тестування у НІТЛ-системах зумовлена низкою фундаментальних чинників. По-перше, людина є динамічним і адаптивним елементом системи. Її поведінка змінюється залежно від контексту, досвіду, довіри до системи та накопичених помилок. Це означає, що навіть коректно протестовані прототипи можуть демонструвати іншу поведінку після інтеграції з повноцінними алгоритмічними модулями. По-друге, взаємодія між компонентами системи створює нові властивості, які неможливо передбачити шляхом ізольованого тестування. Зокрема, можуть виникати затримки, каскадні ефекти помилок або конфлікти управління ініціативою між людиною і системою. По-третє, саме на цьому етапі перевіряється, чи відповідає реальна експлуатаційна поведінка системи етичним, правовим і безпековим вимогам, визначеним на попередніх етапах проектування.

Інтеграційне тестування та налаштування реалізується як систематичний процес перевірки, аналізу та коригування:

1. Інтеграція компонентів у єдиний контур.

На першому кроці здійснюється:

- поєднання алгоритмічних модулів, інтерфейсів та механізмів зворотного зв'язку;
- підключення людини як активного елемента контуру управління;
- перевірка базової працездатності системи у штатному режимі.

Людина на цьому етапі розглядається не як тестовий користувач, а як повноцінний компонент системи, поведінка якого впливає на загальний результат.

2. Тестування сценаріїв у реалістичних умовах.

Проводиться тестування:

- типових сценаріїв експлуатації;

- сценаріїв з підвищеним навантаженням;
- критичних і аварійних сценаріїв.

Метою є оцінка того, як система поводить себе в умовах часових обмежень, неповної інформації або помилок з боку людини, а також чи здатна вона коректно підтримувати або компенсувати ці ситуації.

3. Аналіз узгодженості дій людини і системи.

На цьому кроці аналізуються:

- моменти втрати або конфлікту ініціативи;
- випадки надмірного або недостатнього втручання людини;
- ефективність механізмів пояснення та зворотного зв'язку.

Особлива увага приділяється ситуаціям, коли людина і система приймають суперечливі рішення, та способам розв'язання таких конфліктів.

4. Налаштування параметрів взаємодії.

На основі результатів тестування виконуються:

- коригування порогів спрацювання автоматизованих рішень;
- оптимізація частоти та форми зворотного зв'язку;
- адаптація рівнів автоматизації для різних сценаріїв;
- уточнення процедур втручання людини.

Налаштування спрямоване на досягнення балансу між автономністю системи та ефективною участю людини.

Результатами інтеграційного тестування та налаштування є:

- підтверджена працездатність системи як єдиного цілого;
- скориговані параметри взаємодії людини і системи;
- виявлені та усунуті інтеграційні конфлікти;
- оновлені сценарії та регламенти взаємодії;
- емпірично обґрунтовані показники ефективності та безпеки;
- готовність системи до валідації та дослідної експлуатації.

3.2.6. Валідація, адаптація та масштабування

Етап валідації, адаптації та масштабування є завершальним етапом життєвого циклу проєктування та впровадження системи, у межах якого перевіряється відповідність системи початковим цілям, вимогам і обмеженням, а також її здатність до стійкого функціонування в довгостроковій перспективі та в ширшому контексті застосування. У системах human-in-the-loop об'єктом валідації є не лише технічна коректність і продуктивність системи, але й сталість, передбачуваність та прийнятність взаємодії людини з системою в часі. Людина в такій системі не є статичним елементом, а навпаки – змінює свою поведінку, рівень довіри, стратегії взаємодії та очікування, що потребує спеціального підходу до завершальної оцінки.

Метою етапу є комплексне підтвердження готовності системи до повноцінної експлуатації та подальшого розвитку, зокрема:

- валідація відповідності системи функціональним, нефункціональним і людино-орієнтованим вимогам;
- оцінка довготривалої взаємодії людини і системи;
- забезпечення можливості адаптації системи до змін у контексті використання;
- підготовка системи до масштабування без втрати якості взаємодії та безпеки.

Необхідність цього етапу зумовлена фундаментальними особливостями НІТЛ-систем. По-перше, коректна робота системи в тестових або пілотних умовах не гарантує її ефективності в реальній експлуатації. У реальному середовищі з'являються нові сценарії, нетипові користувачі, тривалі навантаження та організаційні фактори, які не можуть бути повністю відтворені на попередніх етапах. По-друге, людина навчається та адаптується до системи, змінюючи свою поведінку. Без урахування цього ефекту система може поступово втрачати ефективність

або, навпаки, створювати приховані ризики надмірної автоматизації та зниження пильності людини. По-третє, масштабування НІТЛ-системи (за кількістю користувачів, сценаріїв або сфер застосування) без окремої валідації може призвести до неконтрольованого зростання складності взаємодії, що негативно впливає на безпеку та керованість.

Етап валідації, адаптації та масштабування реалізується як багаторівневий процес:

1. Валідація відповідності вимогам і цілям.

На першому кроці здійснюється:

- перевірка досягнення початково визначених цілей системи;
- валідація функціональних і нефункціональних показників;
- підтвердження відповідності системи людино-орієнтованим вимогам.

Особливу увагу приділяють тому, чи збережено баланс між автономністю системи та роллю людини, визначений на ранніх етапах проектування.

2. Оцінка довготривалої взаємодії людини і системи.

Далі аналізується:

- стабільність поведінки системи в часі;
- зміни в стратегіях взаємодії користувачів;
- еволюція рівня довіри до автоматизованих рішень;
- ризики звикання, надмірної залежності або втрати контролю.

Цей аспект є критично важливим для систем, що використовуються на постійній основі або в критичних доменах.

3. Адаптація системи до змін контексту.

На цьому етапі визначаються та впроваджуються механізми:

- адаптації інтерфейсів і зворотного зв'язку;
- коригування рівнів автоматизації;
- оновлення алгоритмічних моделей з урахуванням нових даних;

- навчання або перенавчання користувачів.

Адаптація розглядається не як разова дія, а як безперервний процес підтримки відповідності системи реальним умовам використання.

4. Оцінка готовності до масштабування.

Завершальним кроком є аналіз:

- можливості збільшення кількості користувачів;
- розширення функціональності або сфер застосування;
- перенесення системи в інші організаційні або технічні контексти.

При цьому ключовим критерієм є не лише технічна масштабованість, а здатність зберегти якість людино-машинної взаємодії без зростання ризиків і когнітивного навантаження.

Результатами етапу валідації, адаптації та масштабування є:

- підтверджена відповідність системи початковим цілям і вимогам;
- оцінка довгострокової ефективності та безпеки взаємодії;
- визначені механізми адаптації системи до змін;
- обґрунтовані обмеження та умови масштабування;
- рекомендації щодо подальшого розвитку та супроводу системи.

РОЗДІЛ 4. ЗАСТОСУВАННЯ ІНТЕЛЕКТУАЛЬНИХ ОБЧИСЛЕНЬ ТА ШТУЧНОГО ІНТЕЛЕКТУ В АНТРОПОЦЕНТРИЧНИХ СИСТЕМАХ

4.1. Роль інтелектуальних обчислень в антропоцентричних системах

На відміну від техноцентричних підходів, де алгоритмічна оптимальність є домінантною, антропоцентричні системи розглядають штучний інтелект як засіб підтримки, підсилення та розширення людських когнітивних можливостей, а не як автономного замітника людини. У цьому контексті інтелектуальні обчислення виконують роль інструментального та медіаторного шару між складністю реального середовища та обмеженими когнітивними ресурсами людини.

Інтелектуальні обчислення в антропоцентричних системах виконують насамперед функцію зменшення когнітивного навантаження на людину та підтримки прийняття рішень у складних і невизначених умовах. Сучасні комп'ютерні та інформаційні системи оперують значними обсягами різномірних даних, обробка яких виходить за межі можливостей людини без технічної підтримки. Інтелектуальні алгоритми здійснюють попередній аналіз, узагальнення та структурування інформації, перетворюючи її на форми, придатні для людського сприйняття й інтерпретації. Наприклад, у системах підтримки медичних рішень алгоритми машинного навчання агрегують результати аналізу зображень, лабораторних показників і анамнезу, подаючи лікарю узгоджену картину можливих діагнозів. Подібну роль інтелектуальні обчислення відіграють у фінансово-аналітичних системах, де вони формують зведені індикатори ризиків на основі великих масивів ринкових даних, а також у системах управління транспортними потоками, де складні динамічні процеси

зводяться до зрозумілих рекомендацій для оператора.

Окрім зменшення інформаційної перевантаженості, інтелектуальні обчислення забезпечують підтримку прийняття рішень у ситуаціях невизначеності, що є характерною ознакою антропоцентричних систем. У таких умовах алгоритми не замінюють людину, а пропонують можливі сценарії, оцінки наслідків і рівні ризику, залишаючи остаточне рішення за людиною. Наприклад, у системах реагування на надзвичайні ситуації штучний інтелект може моделювати поширення пожежі або повені за різними сценаріями, тоді як відповідальна особа обирає стратегію дій з урахуванням соціальних і організаційних чинників. Аналогічно, в системах стратегічного планування алгоритми аналізують альтернативні варіанти розвитку подій, але саме людина визначає пріоритети та приймає рішення з огляду на довгострокові наслідки. У промислових системах управління інтелектуальні обчислення допомагають оцінювати компроміси між ефективністю, безпекою та витратами, не усуваючи людину з процесу вибору.

Оскільки людина залишається відповідальною за рішення, результати роботи алгоритмів мають бути інтерпретованими та пояснюваними. Інтелектуальні обчислення в цьому контексті доповнюються механізмами пояснення, які дозволяють відстежувати причини рекомендацій або прогнозів. Наприклад, у системах кредитного скорингу пояснювані моделі дають змогу фінансовим аналітикам зрозуміти, які чинники вплинули на оцінку ризику. У системах технічної діагностики пояснюваність дозволяє інженерам простежити логіку виявлення несправностей, а в освітніх адаптивних платформах – пояснити користувачеві, чому система пропонує саме такий навчальний маршрут. Таким чином, інтелектуальні обчислення стають засобом діалогу, а не «чорним ящиком».

Ще одним суттєвим аспектом є адаптація системи до людини, а не людини до системи. Антропоцентричні системи передбачають врахування

індивідуальних особливостей користувача, його досвіду, стилю прийняття рішень і контексту діяльності. Інтелектуальні обчислення забезпечують таку адаптацію через моделювання користувача та динамічну зміну поведінки системи. Наприклад, у рекомендаційних системах інформаційних ресурсів алгоритми поступово підлаштовуються під інтереси та рівень підготовки користувача. У системах підтримки диспетчерів авіаційного або залізничного транспорту інтерфейс і рівень деталізації інформації можуть змінюватися залежно від досвіду оператора. В освітніх середовищах інтелектуальні обчислення дозволяють адаптувати складність завдань до індивідуальних можливостей здобувача освіти, не порушуючи загальних цілей навчання.

Роль інтелектуальних обчислень простежується також на різних етапах життєвого циклу антропоцентричних систем. На етапі проєктування вони використовуються для моделювання сценаріїв взаємодії людини та системи, що дає змогу заздалегідь оцінити наслідки рішень і виявити потенційні ризики. Наприклад, у проєктуванні систем управління складними технічними об'єктами симуляційні моделі дозволяють дослідити поведінку системи в критичних режимах без реальної загрози. Під час експлуатації інтелектуальні обчислення забезпечують адаптивність і підтримують різні моделі взаємодії, зокрема *human-in-the-loop* та *human-on-the-loop*, як це має місце в системах моніторингу інфраструктури або кібербезпеки. На етапі еволюційного розвитку вони дають змогу враховувати нові дані, зміну умов середовища та людський зворотний зв'язок, що характерно, наприклад, для персоналізованих інформаційних систем або довготривалих проєктів у сфері «розумних міст».

Водночас антропоцентричний підхід вимагає чіткого усвідомлення обмежень і відповідальності, пов'язаних із використанням інтелектуальних обчислень. Незважаючи на зростання автономності алгоритмів, вони не можуть повністю замінити людське судження в ситуаціях, що потребують

етичної оцінки, соціальної відповідальності або врахування неформалізованих чинників. Наприклад, у правових інформаційних системах алгоритми можуть аналізувати судову практику, але остаточне рішення залишається за суддею. У медичних системах підтримки рішень рекомендації штучного інтелекту не скасовують клінічної відповідальності лікаря, а в управлінні персоналом аналітичні інструменти не можуть замінити людське розуміння соціальної динаміки колективу. Таким чином, інтелектуальні обчислення в антропоцентричних системах повинні використовуватися в межах чітко визначених ролей і обмежень, забезпечуючи підтримку людини без втрати її центральної позиції в системі.

4.2. Моделі взаємодії людини та алгоритмів і систем штучного інтелекту

Моделі взаємодії людини та алгоритмів штучного інтелекту, наведені в цьому пункті, не є взаємовиключними й на практиці часто комбінуються в межах однієї системи. Вибір конкретної моделі або їх поєднання визначається рівнем автоматизації, критичністю завдань, вимогами до відповідальності та можливостями людини ефективно взаємодіяти з інтелектуальними компонентами системи.

4.1.1. Людина як спостерігач

У цій моделі система штучного інтелекту функціонує в автоматичному режимі, а роль людини полягає в моніторингу процесів та результатів без безпосереднього втручання в алгоритмічне прийняття рішень (модель human-on-the-loop). Людина відстежує коректність роботи системи, рівень її надійності та відповідність результатів заданим критеріям. Застосування такої моделі є доцільним у випадках стабільного

та добре формалізованого середовища, низької або середньої критичності наслідків помилок, високого рівня довіри до алгоритмічних моделей. Обґрунтуванням використання цієї моделі є зменшення когнітивного навантаження на людину та підвищення продуктивності системи, водночас зберігається можливість переходу до ручного керування у разі виявлення аномалій.

Центральним елементом моделі є алгоритмічне ядро, яке самостійно обробляє вхідні дані, виконує інференцію або оптимізацію, формує рішення або дії в середовищі. До цього компонента можуть належати моделі машинного або глибокого навчання, оптимізаційні або евристичні алгоритми, комбіновані гібридні підходи. Ключовою ознакою є відсутність залежності від негайного людського підтвердження кожного рішення.

Система моніторингу та спостереження в цій моделі забезпечує прозоре відображення стану системи для людини-спостерігача. Вона включає показники продуктивності (точність, швидкодія, стабільність), індикатори довіри та надійності, виявлення відхилень і аномалій. Моніторинг виконує не керувальну, а інформаційну функцію, що принципово відрізняє *human-on-the-loop* від *human-in-the-loop*.

Компонент оцінювання відповідності та обмежень в моделі здійснює автоматичну перевірку відповідності результатів формальним критеріям, дотримання заздалегідь визначених обмежень, узгодженості поведінки з нормативними або технічними правилами. Людина у цій частині не втручається в обчислення, але аналізує агреговані оцінки та сигнали ризику. Цей компонент є ключовим для забезпечення керованої автономії.

Інтерфейс спостереження та пояснення в моделі орієнтований не на керування, а на огляд стану системи; інтерпретацію результатів; пояснення причин поведінки алгоритму (за можливості). Особливістю є знижене когнітивне навантаження, що досягається шляхом агрегування інформації; візуального кодування ризиків; порогових сигналів замість постійної уваги.

Хоча людина не є частиною циклу прийняття рішень, модель передбачає можливість переходу до ручного або напівавтоматичного режиму; зупинку системи; зміну режимів роботи у разі виявлення критичних відхилень. Таким чином, людина залишається остаточним гарантом безпеки, але не операційним агентом.

4.2.2. Людина як коректор рішень

Ця модель передбачає обов'язкову участь людини в циклі прийняття рішень (human-in-the-loop). Алгоритм штучного інтелекту формує попередні результати або рекомендації, які підлягають перевірці, уточненню чи коригуванню людиною перед їх остаточним застосуванням. Модель широко використовується у критичних системах (медицина, безпека, фінанси), за наявності високої невизначеності або неповних даних, у випадках, де помилка має значні соціальні або юридичні наслідки. Її обґрунтування полягає в поєднанні обчислювальної ефективності алгоритмів із контекстуальним, етичним та експертним судженням людини. Така взаємодія підвищує надійність системи, але водночас потребує ретельного проєктування інтерфейсів і процедур верифікації.

Алгоритмічний компонент генерації рішень моделі відповідає за аналіз вхідних даних; формування попередніх рішень, прогнозів або рекомендацій; оцінювання альтернатив (за наявності). Алгоритмічний компонент не є остаточним суб'єктом прийняття рішень. Його функція полягає у зменшенні складності задачі для людини та підготовці структурованих варіантів дій.

Для ефективною корекції людиною результати алгоритму мають бути інтерпретованими; контекстуалізованими; супроводженими поясненнями причин отриманого рішення. Це забезпечує компонент представлення та пояснення результатів, він виконує критичну роль у забезпеченні усвідомленого втручання, без якого модель human-in-the-loop втрачає

антропоцентричний характер.

Людина в цій моделі оцінює коректність і доцільність алгоритмічного результату; враховує контекст, який не був формалізований у моделі; застосовує експертне, етичне або нормативне судження. Людина виступає остаточним суб'єктом відповідальності за прийняте рішення, що є принципово важливим у критичних системах. Механізм корекції та підтвердження реалізує підтвердження рішення без змін, часткове або повне коригування результату, відхилення запропонованого рішення з вимогою перерахунку. Механізм корекції формалізує втручання людини та забезпечує відтворюваність і трасованість прийнятих рішень.

Результати корекції людиною можуть використовуватися для донавчання моделей; уточнювати параметри алгоритмів; оновлювати правила або обмеження. Такий механізм зворотного зв'язку та навчання забезпечує еволюційний розвиток системи на основі людського досвіду.

Модель «людина як коректор рішень» є доцільною, коли автоматизовані рішення мають значні наслідки, існує неповна формалізація предметної області, необхідно забезпечити юридичну та етичну відповідальність. Її застосування дозволяє поєднати обчислювальну ефективність штучного інтелекту з контекстуальною компетентністю людини, що є базовим принципом антропоцентричних систем.

4.2.3. Людина як співавтор рішень (human–AI collaboration)

У цій моделі людина та штучний інтелект виступають рівноправними агентами, які спільно формують рішення. Алгоритм не лише пропонує варіанти, а й адаптується до стилю мислення, пріоритетів і дій людини, тоді як людина активно враховує рекомендації системи. Модель human–AI collaboration передбачає партнерську, симетричну взаємодію людини та алгоритмічного агента, у межах якої рішення формується спільно, і жодна зі сторін не є виключно домінуючою. Людина й штучний інтелект

виступають когнітивно доповнювальними агентами, що поєднують формалізовані обчислювальні можливості з людською інтерпретацією, інтуїцією та контекстуальним мисленням. Ця модель характерна для творчих і слабо формалізованих завдань, стратегічного планування та проєктування, складних аналітичних процесів.

Алгоритмічний компонент у цій моделі генерує гіпотези, варіанти рішень або сценарії; аналізує альтернативи та їх наслідки; адаптується до стилю взаємодії людини. На відміну від *human-in-the-loop*, алгоритм не є лише джерелом пропозицій, а виступає активним учасником колективного мислення.

Роль людини полягає не в перевірці, а в спільному конструюванні рішення, що особливо важливо для слабо формалізованих і відкритих задач. Людина формує стратегічні цілі, інтерпретує результати з урахуванням ширшого контексту, може змінювати критерії, обмеження та саму постановку задачі.

Центральним компонентом моделі зазвичай є інтерактивний когнітивний простір, у якому фіксуються проміжні гіпотези, зберігається історія взаємодії, забезпечується узгодження дій між агентами. Цей простір забезпечує координацію та прозорість спільної діяльності.

У моделі передбачено двосторонню адаптацію: алгоритм адаптується до рішень і уподобань людини; людина коригує власні дії, спираючись на аналітичні можливості системи. Цей механізм реалізує принцип коеволюції людини та системи.

Інтерфейс в цій моделі підтримує діалогову взаємодію, забезпечує гнучку зміну рівня деталізації, дозволяє досліджувати альтернативи та «що-якщо» сценарії. Інтерфейс є не лише засобом відображення, а інструментом спільного мислення.

4.2.4. Людина як джерело знань і навчальних даних

У межах цієї моделі людина виконує функцію носія експертних знань, які використовуються для побудови, налаштування або донавчання моделей штучного інтелекту. Це може реалізовуватися через ручну розмітку даних, формалізацію правил, визначення обмежень або прикладів коректної поведінки системи. Людина виступає первинним носієм предметних знань, які не можуть бути повністю або коректно отримані автоматизовано. Модель є ключовою на етапах початкового навчання моделей, у доменах з обмеженою кількістю даних, за потреби забезпечення інтерпретованості результатів. Її обґрунтування пов'язане з неможливістю повної автоматизації процесу набуття знань у складних предметних областях, де значна частина знань є неявною або контекстуальною.

Людина в цій моделі володіє неявними, контекстуальними або процедурними знаннями, інтерпретує предметну область, формує уявлення про допустимі та недопустимі стани системи. Людина не взаємодіє з результатами виконання системи, а передає знання до моменту або незалежно від експлуатації. Оскільки людські знання є неформалізованими за своєю природою, в моделі обов'язково необхідний компонент, який забезпечує перетворення знань у формальні представлення, структурування понять, правил і зв'язків, узгодження термінології та інтерпретацій. Формалізація може здійснюватися у вигляді правил і обмежень, онтологій, прикладів і контрприкладів, анотованих даних.

Якщо модель містить компонент створення навчальних даних, людина бере участь у розмітці даних, визначенні класів, міток і атрибутів, оцінюванні якості та репрезентативності наборів даних. Цей компонент є ключовим у доменах, де дані є обмеженими, автоматичне маркування є неможливим, а помилки мають високі наслідки.

Алгоритмічний компонент навчання використовує надані знання та дані, формує початкові або уточнені моделі, забезпечує узгодження між

знаннями та статистичними закономірностями. Важливою особливістю є залежність якості моделі від якості людського внеску.

Механізм валідації та узгодження знань забезпечує перевірку узгодженості знань між різними експертами, виявлення суперечностей, уточнення або перегляд формалізованих уявлень. Людина може повертатися до етапу формалізації для уточнення знань.

4.2.5. Людина як джерело зворотного зв'язку та адаптації

Ця модель фокусується на використанні поведінкових і явних реакцій людини для безперервного вдосконалення роботи системи. Зворотний зв'язок може бути прямим (оцінки, коментарі) або непрямим (патерни використання, корекції рішень). Модель «людина як джерело зворотного зв'язку та адаптації» передбачає, що людина не бере безпосередньої участі в оперативному прийнятті рішень, але систематично або подієво надає інформацію про якість, доцільність або відповідність результатів роботи інтелектуальної системи. Цей зворотний зв'язок використовується для адаптації, донавчання та еволюційного вдосконалення алгоритмічних компонентів. Застосовується переважно в адаптивних інтерфейсах, рекомендаційних та персоналізованих системах, системах, що функціонують у динамічному середовищі.

Алгоритмічний компонент моделі функціонує автономно в режимі виконання задач, генерує результати, рекомендації або дії, містить механізми оновлення параметрів на основі зворотного зв'язку. На відміну від *human-in-the-loop*, рішення не зупиняється в очікуванні людини, а коригується *постфактум*.

Людина в цій моделі оцінює якість або корисність результатів, фіксує помилки, невідповідності або небажані ефекти, надає явний або неявний зворотний зв'язок. Людина не змінює конкретне рішення, але впливає на майбутню поведінку системи. Зворотний зв'язок може бути явним (оцінки,

коментарі, мітки правильності) або неявним (поведінкові патерни, частота використання, відмова від рекомендацій). Оскільки людський зворотний зв'язок часто є шумним, неповним, контекстно залежним, система потребує окремого механізму його агрегації, нормалізації, перетворення на навчальний сигнал.

На основі інтерпретованого зворотного зв'язку оновлюються параметри моделей, коригуються ваги, пороги або правила, уточнюються профілі користувачів. Цей компонент забезпечує довготривалу антропоцентричну адаптацію системи.

4.3. Етичні аспекти

Застосування інтелектуальних обчислень та штучного інтелекту в антропоцентричних системах неминуче супроводжується комплексом етичних питань, оскільки такі системи безпосередньо впливають на діяльність, рішення та відповідальність людини. На відміну від повністю автоматизованих технічних систем, антропоцентричні системи передбачають взаємодію між алгоритмічними компонентами та людським суб'єктом, що зумовлює необхідність розгляду не лише технічної ефективності, а й моральних, соціальних і правових наслідків їх використання.

Однією з ключових етичних проблем є питання відповідальності за прийняті рішення. В антропоцентричних системах рішення часто формуються в результаті спільної діяльності людини та алгоритму, що ускладнює визначення суб'єкта відповідальності у разі помилки або негативних наслідків. Якщо рекомендація сформована системою штучного інтелекту, але остаточне рішення ухвалює людина, виникає етична напруга між формальною відповідальністю людини та фактичним впливом алгоритму на її вибір. Особливо гостро ця проблема проявляється в

медичних, юридичних і безпекових системах, де помилки можуть мати серйозні наслідки для життя, здоров'я або прав інших осіб. З етичної точки зору неприпустимим є перекладання відповідальності на алгоритм, оскільки він не є моральним агентом, а отже не може нести етичної чи правової відповідальності.

Тісно пов'язаним із відповідальністю є питання прозорості та пояснюваності рішень, що формуються інтелектуальними обчисленнями. В антропоцентричних системах людина має не лише виконувати роль користувача, а й усвідомлювати підстави, на яких система пропонує певні рекомендації або прогнози. Використання непрозорих моделей, результати яких неможливо інтерпретувати, створює етичну проблему позбавлення людини можливості критично оцінювати рішення системи. Це може призводити до сліпої довіри до алгоритмів або, навпаки, до їх необґрунтованого відторгнення. З етичної позиції пояснюваність є необхідною умовою збереження автономії людини як суб'єкта прийняття рішень і запобігання маніпулятивному впливу з боку системи.

Важливим етичним аспектом є збереження людської автономії та недопущення надмірної алгоритмічної залежності. Антропоцентричні системи створюються з метою підтримки людини, однак за відсутності належних обмежень вони можуть поступово підміняти людське судження, формуючи ситуацію так званого «когнітивного аутсорсингу». У такому випадку людина починає систематично покладатися на рекомендації системи, навіть у ситуаціях, де необхідне критичне мислення або врахування контекстуальних чинників. Етична проблема полягає в тому, що втрата навичок самостійного аналізу та прийняття рішень може мати довгострокові негативні наслідки для професійної компетентності та відповідальності людини.

Окрему групу етичних питань становлять проблеми упередженості та дискримінації, закладені в алгоритмічних моделях. Інтелектуальні

обчислення ґрунтуються на даних, які відображають реальні соціальні практики, а отже можуть містити приховані упередження. В антропоцентричних системах це створює особливо серйозну етичну загрозу, оскільки алгоритмічні рекомендації можуть впливати на людські рішення, посилюючи або легітимізуючи дискримінаційні практики. Наприклад, у системах відбору персоналу, кредитного скорингу або оцінювання ризиків алгоритмічні упередження можуть непрямо формувати рішення людини, навіть якщо вона формально зберігає контроль. Етичним обов'язком розробників і користувачів таких систем є виявлення, аналіз і мінімізація упередженостей, а також усвідомлення обмеженості алгоритмічних висновків.

Не менш значущим є питання захисту персональних даних і приватності, оскільки антропоцентричні системи часто працюють з чутливою інформацією про людину, її поведінку, уподобання та професійну діяльність. Інтелектуальні обчислення потребують великих обсягів даних для навчання й адаптації, що підвищує ризики несанкціонованого доступу, вторинного використання даних або втрати контролю над ними. З етичної точки зору неприйнятним є використання персональних даних без усвідомленої згоди людини або в цілях, які виходять за межі первісно задекларованого призначення системи. У контексті антропоцентричних систем особливо важливо забезпечити баланс між адаптивністю системи та збереженням приватності користувача.

Ще одним суттєвим етичним виміром є справедливий розподіл повноважень між людиною та системою. Антропоцентричний підхід передбачає, що людина зберігає можливість втручання, корекції або скасування рішень, запропонованих системою. Якщо архітектура системи або організаційні регламенти фактично позбавляють людину такого права, антропоцентричність стає декларативною. Етична проблема полягає в тому, що формальна присутність людини в контурі управління не гарантує її

реального впливу на процеси, особливо в умовах високої швидкості або складності алгоритмічних рішень.

У ширшому соціальному контексті етичні питання застосування інтелектуальних обчислень в антропоцентричних системах пов'язані також із довірою до технологій. Довіра формується не лише через технічну надійність, а й через відповідність системи етичним очікуванням суспільства. Якщо система сприймається як несправедлива, непрозора або така, що підриває людську гідність, це знижує готовність людей взаємодіяти з нею та приймати її рекомендації. В антропоцентричних системах довіра є критично важливою, оскільки ефективність таких систем значною мірою залежить від активної участі людини.

Таким чином, етичні питання при застосуванні інтелектуальних обчислень та штучного інтелекту в антропоцентричних системах охоплюють відповідальність, прозорість, автономію, недискримінацію, захист даних і реальний людський контроль. Їх розв'язання вимагає не лише технічних рішень, а й міждисциплінарного підходу, що поєднує інженерні, правові та етичні міркування, з чітким усвідомленням того, що людина залишається центральним елементом системи, а не допоміжним компонентом алгоритмічного процесу.

ЛІТЕРАТУРА

1. Norman, Donald A. The design of everyday things. 2013. 369 p. <https://dl.icdst.org/pdfs/files4/4bb8d08a9b309df7d86e62ec4056ceef.pdf>
2. Shneiderman, “Direct Manipulation: A Step Beyond Programming Languages” in Computer, vol. 16, no. 8, pp. 57-69, Aug. 1983, doi: 10.1109/MC.1983.1654471. <https://scispace.com/pdf/direct-manipulation-a-step-beyond-programming-languages-266253emfp.pdf>
3. Dourish, Paul. (2001). Where the Action Is: The Foundations of Embodied Interaction.
4. Human Work Interaction Design: Designing for Human Work. Edited by T. Clemmensen, P. Campos, R. Omgreen, A. Mark Pejtersen, and W. Wong. The first IFIP TC 13.6 WG Conference: Designing for Human Work, February 13-15, 2006, Madeira, Portugal. <https://doi.org/10.1007/978-0-387-36792-7>
5. Susanne Bødker. 2006. When second wave HCI meets third wave challenges. In Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles (NordiCHI '06). Association for Computing Machinery, New York, NY, USA, 1–8. <https://doi.org/10.1145/1182475.1182476>
6. Jaimes, Alejandro & Gatica-Perez, Daniel & Sebe, Nicu. (2007). Human-Centered Computing: Toward a Human Revolution. Computer. 40. 30 - 34. 10.1109/MC.2007.169.
7. Schmager, S., Pappas, I. O., & Vassilakopoulou, P. (2025). Understanding Human-Centred AI: a review of its defining elements and a research agenda. Behaviour & Information Technology, 44(15), 3771–3810. <https://doi.org/10.1080/0144929X.2024.2448719>
8. Ozmen Garibay, O., Winslow, B., Andolina, S., Antona, M., Bodenschatz, A., Coursaris, C., ... Xu, W. (2023). Six Human-Centered Artificial Intelligence Grand Challenges. International Journal of Human–Computer

Interaction, 39(3), 391–437. <https://doi.org/10.1080/10447318.2022.2153320>

9. Nakao, Y., Strappelli, L., Stumpf, S., Naseer, A., Regoli, D., & Gamba, G. D. (2022). Towards Responsible AI: A Design Space Exploration of Human-Centered Artificial Intelligence User Interfaces to Investigate Fairness. *International Journal of Human–Computer Interaction*, 39(9), 1762–1788. <https://doi.org/10.1080/10447318.2022.2067936>

10. Ryan, M. We’re only human after all: a critique of human-centred AI. *AI & Soc* 40, 1303–1319 (2025). <https://doi.org/10.1007/s00146-024-01976-2>

11. Lee-Smith, M., Ross, T., Wilson, G. T., Tso, F. P., Cavazzi, S., & Morley, J. G. (2024). The data hungry home: A post-anthropocentric and generative design framework. *International Journal of Design*, 18(2), 79-97. <https://doi.org/10.57698/v18i2.05>

12. Umbrello S. The Ecological Turn in Design: Adopting a Posthumanist Ethics to Inform Value Sensitive Design. *Philosophies*. 2021; 6(2):29. <https://doi.org/10.3390/philosophies6020029>

13. García-Acosta, Gabriel & Romeva, C.R.. (2010). From anthropocentric design to ecospheric design: Questioning design epicentre. 11th International Design Conference, DESIGN 2010. 29-38. https://www.researchgate.net/publication/289724464_From_anthropocentric_design_to_ecospheric_design_Questioning_design_epicentre

14. Avsec S, Jagiełło-Kowalczyk M, Żabicka A, Gil-Mastalerczyk J, Gawlak A. Human-Centered Systems Thinking in Technology-Enhanced Sustainable and Inclusive Architectural Design. *Sustainability*. 2024; 16(22):9802. <https://doi.org/10.3390/su16229802>

15. Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

16. John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3(4), 320–351.

17. Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world task performance. *Human-Computer Interaction*, 8(3), 237–309.
18. Zhou, Xiaozhou & Teng, Fei & Du, Xiaoxi & Li, Jiarui & Jin, Minxin & Xue, Chengqi. (2022). H-GOMS: a model for evaluating a virtual-hand interaction system in virtual environments. *Virtual Reality*. 27. 10.1007/s10055-022-00674-y.
19. Card, S. K., Moran, T. P., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7), 396–410.
20. Lee, Ahreum & Song, Kiburm & Ryu, Hokyoung & Kim, Jieun & Kwon, Gyu Hyun. (2015). Fingerstroke time estimates for touchscreen-based mobile gaming interaction. *Human movement science*. 44. 211-224. 10.1016/j.humov.2015.09.003.
21. Kim, Min Ji & Eune, Juhyun. (2025). "안전디딤돌 앱의 사용자 경험 개선: GOMS 모델 적용" "Enhancing User Experience of the Safety Stepping Stone App: Applying the GOMS Model".
22. Li, Wentao & Li, Zhizhong. (2024). Mental Workload Prediction Method Based on GOMS. 10.54941/ahfe1004861.
23. Hashim, Nor Hazlina; Jones, Michael (2007). Activity theory: a framework for qualitative analysis. University of Wollongong. Conference contribution. <https://hdl.handle.net/10779/uow.27796473.v1>
24. Brooke, John. (1995). SUS: A quick and dirty usability scale. *Usability Eval. Ind.*. 189.
25. Schrepp, Martin. (2015). User Experience Questionnaire Handbook. 10.13140/RG.2.1.2815.0245.
26. A teamwork questionnaire. Te Hononga Akoranga COMET. URL: <https://cometauckland.org.nz/assets/files/Teamwork-Questionnaire.pdf>

27. Schrepp, Martin. (2015). User Experience Questionnaire Handbook. 10.13140/RG.2.1.2815.0245.
28. Team Effectiveness Questionnaire. University of Colorado. URL: https://www.cu.edu/sites/default/files/Team_effectiveness_questionnaire.pdf
29. Kreijns, Karel & Kirschner, Paul & Jochems, Wim & Buuren, Hans. (2011). Measuring perceived social presence in distributed learning groups. *Education and Information Technologies*. 16. 365-381. 10.1007/s10639-010-9135-7.
30. Kreijns, Karel & Xu, Kate & Weidlich, Joshua. (2021). Social Presence: Conceptualization and Measurement. *Educational Psychology Review*. 34. 10.1007/s10648-021-09623-8.
31. Frank Biocca, Chad Harms. Networked Minds Social Presence Inventory. (Scales only, Version 1.2). Measures of co-presence, social presence, subjective symmetry, and intersubjective symmetry. URL: https://web-archive.southampton.ac.uk/cogprints.org/6742/1/2002_netminds_scales.pdf
32. Farris, Frank. (2010). The Gini Index and Measures of Inequality. *American Mathematical Monthly*. 117. 851-864. 10.4169/000298910X523344.
33. Díaz, Alberto & Gervás, Pablo. (2003). Dynamic User Modeling in a System for Personalization of Web Contents. 281-290. 10.1007/978-3-540-25945-9_28.
34. Azzam, Fatima & Kayed, Mohammed. (2022). A Model for Generating a User Dynamic Profile on Social Media. *Journal of King Saud University - Computer and Information Sciences*. 34. 10.1016/j.jksuci.2022.08.036.
35. Varga, Balint, Flemisch, Frank and Hohmann, Sören. "Human in the loop" at - *Automatisierungstechnik*, vol. 72, no. 12, 2024, pp. 1109-1111. <https://doi.org/10.1515/auto-2024-0148>
36. International Journal of Human-Computer Interaction. <https://www.tandfonline.com/toc/hihc20/current>
37. Android UI design guides <https://developer.android.com/>

[design/ui/mobile/guides/foundations/accessibility?authuser=0](https://developer.apple.com/design/ui/mobile/guides/foundations/accessibility?authuser=0)

38. Human Interface Guidelines (Apple platform)
<https://developer.apple.com/design/human-interface-guidelines?authuser=0>

39. Flat-Design Best Practices. Nielsen Norman Group. <https://www.nngroup.com/articles/flat-design-best-practices/>

40. Material Design <https://m3.material.io/?authuser=0>

41. Настанови з доступності вебвмісту (WCAG)
<https://www.w3.org/Translations/WCAG21-ua/>

42. Національна стратегія зі створення безбар'єрного простору в Україні до 2030 року. <https://zakon.rada.gov.ua/laws/show/366-2021-%D1%80#n10>

43. Jain, Nilakshi. (2023). UI Design : Key to Captivate User Understanding. 173 p.

44. Tripathy, Abhijit & Nanda, Sai Soumyak & Gupta, Khusbu & Mishra, Priyanka & Monisha, Chippada. (2022). Demystifying UI/UX: A client's guide on understanding User Interfaces and User Experience.

45. Stull E. UX Fundamentals for Non-UX Professionals: User Experience Principles for Managers, Writers, Designers, and Developers. Apress, 2018. 331p.

46. Tomlin Craig W. UX Optimization: Combining Behavioral UX and Usability Testing Data to Optimize Websites. Apress, 2018. 198 p.

47. Canziba Elvis. Hands-On UX Design for Developers. Packt Publishing, 2018. 350p.

48. Lull, D. (2017). UX Psychology Basics. In: Discussions in User Experience . Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3267-5_4

49. Золотухіна О.А., Ільїн О.Ю., Горячев Т.В. Автоматизована обробка спеціалізованих запитів користувачів. Наукові праці ДонНТУ,

Серія “Інформатика, кібернетика та обчислювальна техніка”, №1, 2024. С.44-50.

50. Methods of Biometric Authentication for Person Identification/ Daria Polunina, Oksana Zolotukhina, Olena Nehodenko, and Iryna Yarosh. 2nd International Congress of Electrical and Computer Engineering (ICECENG'23), November 22 to 25, 2023, Bandirma, Turkey. P.327-339. https://doi.org/10.1007/978-3-031-52760-9_23.

51. Бажан Ю.П., Золотухіна О.А. Ідентифікація поведінкових патернів користувачів у UI/UX-дизайні на основі аналізу користувацьких логів. Зв'язок. 2025. №4. С.67-75. <https://doi.org/10.31673/2412-9070.2025.040867>

52. Золотухіна О.А., Худік Б.О., Сорокін Д.В. Людиноцентричний аудит і концептуальний реінжиніринг українських цифрових сервісів. Наука і техніка сьогодні. Серія «Техніка», 2025. №13 (54). С. 2090-2110. 10.52058/2786-6025-2025-13(54)-2090-2110

Додаток А
ПРИКЛАД ЧЕКЛІСТА ДЛЯ
ОЦІНКИ АНТРОПОЦЕНТРИЧНОСТІ СИСТЕМИ

1. Орієнтація на користувача та завдання

Чи визначені основні категорії користувачів системи?

Чи проведено аналіз завдань і контекстів використання?

Чи відображаються реальні потреби користувача у функціональних можливостях системи?

2. Когнітивні та фізичні можливості

Чи враховано обмеження пам'яті та уваги користувача?

Чи підтримується зрозуміле подання інформації (структура, візуальні та сенсорні сигнали)?

Чи адаптовано інтерфейс під різні сенсорні та моторні можливості користувачів?

3. Інтерактивність та зворотний зв'язок

Чи надає система чіткий зворотний зв'язок на дії користувача?

Чи відображає система стан виконання завдань і помилок?

Чи можна контролювати темп взаємодії та коригувати дії без шкоди для результату?

4. Персоналізація та адаптація

Чи підтримується налаштування інтерфейсу та функціоналу під різні групи користувачів?

Чи система адаптується до досвіду, навичок або контексту конкретного користувача?

Чи є механізми допомоги, підказок та навчання користувача?

5. Етика, доступність та інклюзивність

Чи враховано різні групи користувачів, включно з людьми з обмеженими можливостями?

Чи забезпечується захист приватності та безпека даних?

Чи система не дискримінує за культурними, віковими чи соціальними ознаками?

6. Інтеграція із середовищем і соціальний контекст

Чи враховано середовище, в якому використовується система?

Чи підтримується взаємодія між користувачами через систему (якщо релевантно)?

Чи система сприяє продуктивності та добробуту користувача?

7. Процес розробки

Чи користувачі брали участь у тестуванні та прототипуванні системи?

Чи ітеративно вдосконалюється дизайн на основі зворотного зв'язку користувачів?

Чи є документація, що відображає рішення, прийняті з урахуванням потреб користувача?

ДОДАТОК Б
ПРИКЛАДИ АНКЕТ ДЛЯ ОЦІНЮВАННЯ ВЗАЄМОДІЇ З
КОРИСТУВАЧЕМ

Б.1 System Usability Scale (SUS)

© Digital Equipment Corporation, 1986

(методика надається в авторському перекладі на основі джерела

https://digital.ahrq.gov/sites/default/files/docs/survey/systemusabilityscale%2528sus%2529_comp%255B1%255D.pdf)

1. Я думаю, що хотів би часто користуватися цією системою.

--	--	--	--	--

1 2 3 4 5

2. Я вважаю систему надмірно складною.

--	--	--	--	--

1 2 3 4 5

3. Я вважаю, що система проста у використанні.

--	--	--	--	--

1 2 3 4 5

4. Я думаю, що мені знадобиться підтримка технічного спеціаліста, щоб користуватися цією системою.

--	--	--	--	--

1 2 3 4 5

5. Я вважаю, що різні функції в цій системі добре інтегровані.

--	--	--	--	--

1 2 3 4 5

6. Я вважаю, що в цій системі забагато невідповідностей.

--	--	--	--	--

1 2 3 4 5

7. Я гадаю, що більшість людей дуже швидко навчаться користуватися цією системою.

--	--	--	--	--

1 2 3 4 5

8. Я вважаю систему дуже громіздкою у використанні.

--	--	--	--	--

1 2 3 4 5

9. Я почувався дуже впевнено, користуючись системою.

--	--	--	--	--

1 2 3 4 5

10. Мені потрібно було багато чого вивчити, перш ніж я зможу почати працювати з цією системою.

--	--	--	--	--

1 2 3 4 5

Використання SUS

Шкала SU зазвичай використовується після того, як респондент мав можливість скористатися оцінюваною системою, але перед будь-яким дебрифінгом чи обговоренням. Респондентів слід попросити записати свою безпосередню відповідь на кожен пункт, а не довго думати про них. Усі пункти слід перевірити. Якщо респондент вважає, що не може відповісти на певний пункт, він повинен позначити центральну точку шкали.

Оцінювання SUS

SUS дає одне число, яке представляє складений показник загальної зручності використання досліджуваної системи. Важливо, що бали за окремі пункти не є значущими самі по собі. Щоб розрахувати бал SUS, спочатку підсумуйте внески кожного пункту в бали. Внесок кожного пункту в бали коливатиметься від 0 до 4. Для пунктів 1, 3, 5, 7 та 9 внесок у бали дорівнює позиції шкали мінус 1. Для пунктів 2, 4, 6, 8 та 10 внесок дорівнює 5 мінус позиція шкали. Помножте суму балів на 2,5, щоб отримати загальне значення SU. Бали SUS мають діапазон від 0 до 100.

Б.2 User Experience Questionnaire (UEQ)

(методика надається в авторському перекладі та обробці на основі джерела <https://www.ueq-online.org/>)

Для оцінки продукту користувачам пропонується заповнити нижченаведену анкету. Опитувальний лист складається з пар протилежних за значенням визначень, які можна використовувати щодо продукту. Кола між парними визначеннями означають ступінь градації з-поміж них. Користувачі можуть висловити згоду з визначенням, відзначивши коло, яке найточніше відображає їх враження про продукт.

Електронне навчальне видання

Золотухіна Оксана Анатоліївна

**МЕТОДИ ТА ЗАСОБИ
АНТРОПОЦЕНТРИЧНИХ ОБЧИСЛЕНЬ
ТА ТЕХНОЛОГІЙ**

Навчальний посібник