

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

Агентно-базований підхід до моделювання колективної роботи
Текстова частина до курсової роботи
за спеціальністю «Комп’ютерні науки» - 122

Керівник курсової роботи

доцент

Гороховський С.С.

(Підпис)

“___” _____ 2021 року

Виконала студентка КН-МП1

Радзієвська О.В.

“___” _____ 2021 року

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Викладач кафедри інформатики,
доцент _____ Гороховський С. С.
(підпис)

„_____” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Радзієвській Олександрі Володимирівні

факультету інформатики 1 курсу магістерської програми

**ТЕМА: Агентно-базований підхід до моделювання колективної
роботи**

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

Огляд теоретичного матеріалу і здійснення дослідження

Висновки

Список літератури

Додатки (за необхідністю)

Список посилань

Дата видачі „_____” _____ 2020 р.

Керівник _____
(підпис)

Завдання отримав _____
(підпис)

**Тема: Агентно-базований підхід до моделювання колективної
роботи**

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	23.12.2020	
2.	Огляд літератури за темою роботи.	12.01.2021	
3.	Проведення досліджень	28.03.2021	
4.	Аналіз отриманих результатів з керівником	02.04.2021	
5.	Написання практичної частини	14.04.2021	
6.	Написання теоретичної частини	28.04.2021	
7.	Корегування роботи	01.05.2021	
8.	Створення презентації та написання доповіді.	05.05.2021	
9.	Остаточне оформлення пояснювальної роботи та слайдів.	07.05.2021	
10.	Захист курсової роботи	Травень 2021	

Студент _____
Керівник _____
“ ”

Зміст

Анотація	5
Вступ	6
Розділ 1. Мультиагентні системи. Особливості, проблеми й перспективи	8
1.1 Основні характеристики мультиагентних систем	8
2.1 Відмінність агентів від інших парадигм	10
1.2 Області застосувань агентних рішень	12
1.3 Особливості побудови агентів та основні підходи до розробки таких агентів.....	14
1.3 Основні проблеми, пов'язані з проектуванням мультиагентних систем	17
1.4 Висновки.....	19
Розділ 2. Організація колективної роботи.....	20
2.1 Важливість соціальних навиків в організації колективної роботи.....	20
3.2 Перспективи дослідження впливу соціальних навиків на роботу в команді	22
1.3 Висновки.....	23
Розділ 3. Опис створеної системи для моделювання колективної роботи	24
3.1 Загальний опис системи	24
3.2 Аналіз результатів	26
3.3 Опис інтерфейсу	29
3.4 Обґрунтування вибору технологій.....	31

3.4 Перспективи та недоліки	32
3.5 Висновки.....	33
Висновки	34
Джерела	35

Анотація

У даній роботі розглядається використання агентно-базованого підходу для моделювання соціальних взаємодій в команді при розв'язанні певної задачі. Зокрема увага звертається на випадки, коли не всі агенти однаковий рівень зацікавленості у досягненні цілі. Результатом роботи є система, що дозволяє експериментувати з різними параметрами агентів і досліджувати як це впливає на загальний успіх команди.

Вступ

Світ швидко розвивається, комп'ютери заміняють людину в усіх сферах, які не потребують творчого підходу, вміння продукувати нові ідеї ціниться як ніколи і разом з тим зростає необхідність навичок роботи в команді. Очевидно, що якщо над масштабним завданням працює група осіб, у перспективі це матиме значно більший шанс на успіх, адже кожен учасник вкладає в цей проєкт свій власний досвід та знання. Крім того відбувається навчання учасників одне від одного, що має позитивні наслідки в довготривалій перспективі. Але звичайно, робота команди не завжди може мати лише позитивні сторони, адже кожна людина індивідуальна і часто в членів команди можуть не співпадати точки зору на певну річ, вони можуть мати різні цілі або різну степінь вмотивованості. Коли людину наймають на роботу в місце в яке в усталеній команді, рекрутери інтуїтивно намагаються зрозуміти чи зможе ця людина максимально проявити себе саме в цій команді і на такому завданні. Було б добре мати змогу дослідити як різний темперамент впливає на роботу в команді та наскільки робота в команді має переваги над індивідуальною роботою. Один з варіантів дослідження - використати моделювання з допомогою агентно-базованого підходу, що може допомогти краще зрозуміти природу процесів.

Агентно-базований підхід сформувався на перетині багатьох галузей таких як штучного інтелекту, соціології, теорії ігор та інших. Його ключова перевага в тому, що він підходить для розв'язання задач, для яких не існує чіткого алгоритму. Агентно-базований підхід є дуже природньою абстракцією соціальних взаємовідносин, адже дозволяє представляти об'єкти реального світу (в даному випадку членів команди) як агентів з певними характеристиками, діями, що вони можуть виконувати та реакцією на зміну

середовища де вони перебувають. Цей метод широко використовується для моделювання процесів та явищ, та надає змогу краще розуміти глибинні процеси, зокрема в соціальних взаємовідносинах.

У даній роботі буде обговорено важливість соціальних навичок та переваги командної роботи в порівнянні з індивідуальною, продемонстрована робота ключових принципів агентно-базованого підходу на прикладі моделі, агенти якої разом проходять тест, намагаючись досягти максимального відсотку правильних відповідей. Також буде представлено інтерфейс, з допомогою якого можна досліджувати, як значення параметрів агентів впливають на загальний результат, та як агенти з різним рівнем інтелекту та мотивації взаємодіють між собою.

Перший розділ даної роботи присвячений загальному огляду агентно-базованого підходу, сфер використання, складнощів та відмінностей від інших підходів.

Другий розділ присвячений обговоренню важливості соціальних навиків у сучасному світі.

У третьому розділі буде надано опис розробленої системи, обговорено недоліки та перспективи для покращення.

Розділ 1. Мультиагентні системи. Особливості, проблеми й перспективи

1.1 Основні характеристики мультиагентних систем

Мультиагентна система (Multi-agent system) - це система, що складається з декількох агентів - автономних незалежних компонентів, здатних взаємодіяти між собою для досягнення певних цілей. Варто зауважити, що під взаємодією мається на увазі не простий обмін інформацією, а щось на зразок імітації взаємодії людей в реальному світі, в якій мають місце переговори, компроміси і домовленості. Простими словами агентам в системі не задається чіткий алгоритм дій, який вони мають виконати. Натомість їм вказується ціль, яку вони мають досягнути, і вони самостійно обирають шляхи її досягнення [1].

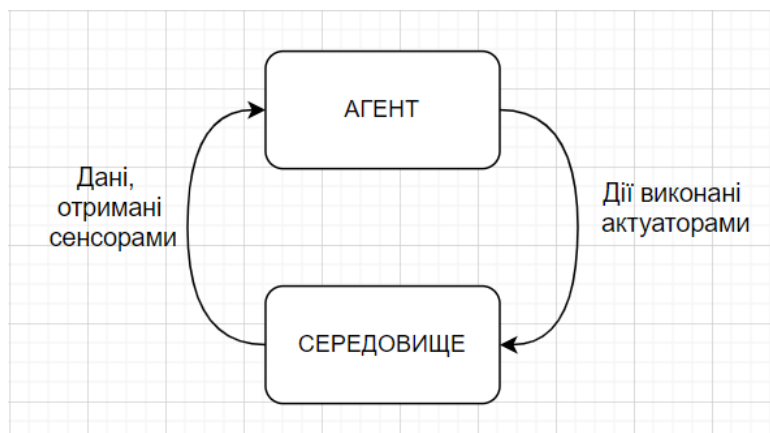


Рис. 1.1 Спрощена схема агента

Джеффри Вулдрідж та Нік Дженнінгс в 1995 сформулювали список особливостей, яким мають відповідати інтелектуальні агенти [2].

- **Reactivity** - здатність агента сприймати навколишнє середовище і своєчасно підлаштовувати свою поведінку у відповідності до поставлених цілей;

- **Proactiveness** - здатність агента демонструвати свою спрямованість на досягнення мети, беручи ініціативу в прийнятті рішень у відповідності до поставлених цілей;
- **Social ability** - здатність агента взаємодіяти в іншими агентами (і можливо людьми) у відповідності до поставлених цілей.

У наступних розділах буде більш детально розглянуто ці пункти та окреслено складнощі, пов'язані з ними.

Кожен агент має певну ціль і намагається її досягнути, приймаючи рішення, що базуються на його власних поведінкових налаштуваннях, налаштуваннях навколишнього середовища, інформації від інших агентів та історії попередніх рішень. Оскільки цілі інших агентів в системі можуть бути абсолютно протилежні, то для досягнення найкращого для всіх результату їм доводиться використовувати згадані методи соціальної взаємодії: співпраця, узгодження рішень та укладення домовленостей.

Мультиагентні системи можна поділити на два класи - homogeneous та heterogeneous. У першому випадку всі агенти мають однаковий набір поведінкових патернів. Прикладом може бути задача про дилему ув'язненого, де всі агенти мають однакову стратегію перемовин. У другому - агенти мають різні характеристики та поведінку. Як приклад можна навести модель біржі, де частина агентів виступає продавцями, а частина покупцями, які очевидно матимуть різні характеристики та цілі. [3]

Широке використання агентної парадигми почалось в середині 1990-х (хоча перші дослідження мультиагентних систем почалось в 1980), коли в ній вбачались великі перспективи в дослідженні великих масивів даних в розподілених системах, таких як Інтернет. Хоча агенти мають великий потенціал в цій сфері, їх можливості не обмежуються лише цим. Парадигма агентів є інтуїтивно зрозумілою метафорою для побудови систем які можна

назвати *artificial social systems* [1]. За допомогою агентно-базованого підходу можна моделювати різноманітні біологічні та екологічні системи, поведінку груп людей тощо. Більш конкретні приклади будуть наведені в пункті «Області застосування агентних рішень».

2.1 Відмінність агентів від інших парадигм

Агентна парадигма поєднала в собі багато різних галузей і підходів, і якщо не враховувати всіх деталей, може здатися що вона нічим не відрізняється, наприклад, від об'єктно-орієнтованої парадигми. Далі буде розглянуто підходи, які мають багато схожих рис з агентами, щоб на їх прикладах відслідкувати важливі відмінності.

Визначення агентів як сутностей, що мають певні властивості і поведінку дуже нагадує визначення об'єктів в об'єктно-орієнтованій парадигмі. Тим не менш, агенти мають декілька ключових відмінностей від об'єктів.

1. Об'єкт завжди виконує те, що йому сказали виконати, викликавши один з його методів. На противагу цьому, агент має змогу самостійно вирішувати чи буде він виконувати певну дію. Якщо ми говоримо про об'єктно-орієнтовану парадигму, відповідальність за рішення лежить на тому, хто викликає метод, у мультиагентній, відповідальність за рішення на агенті, якого попросили щось зробити. Вулдрідж описує цю різницю слоганом *"Objects do it for free; agents do it because they want to"*.
2. Класичним об'єктам абсолютно не притаманні якості, наявні в інтелектуальних агентах - reactivity, proactiveness та social ability - які відповідають за гнучкість агента в прийнятті рішень.

3. Кожен агент вимагає щонайменше один власний потік, тоді як у класичному випадку в системі написаній в об'єктно-орієнтованій парадигмі наявний один основний потік. [1]

Згадка про потоки виконання, насправду на думку про паралельні/розподілені системи (concurrent/distributed systems.). Дійсно, за визначенням мультиагентна система є підкласом паралельної, але має дві важливі ключові риси, які її відрізняють.

1. У паралельній/розподіленій системі правила синхронізації і координації структур чітко програмуються на етапі розробки, тоді як для агентної системи потрібен механізм, який би допомагав агентам синхронізуватися і координувати свої дії в процесі виконання.
2. У класичній паралельній/розподіленій системі всі компоненти мають спільну ціль - забезпечити коректне функціонування системи. У той же час, агенти можуть мати різні персональні цілі. [1]

Деякі агентні системи за своєю функціональністю схожі до експертних систем - системи на основі штучного інтелекту, що накопичує дані з певної області і може надавати ефективні рішення певного спектру задач. Та вони також не є ідентичними, адже експертна система отримує вхідну інформацію не з навколишнього середовища за допомогою сенсорів, а її задає користувач. Так само на відміну від агентів, які впливають на навколишнє середовище за допомогою актуаторів, експертні системи лише виводять результати своєї роботи користувачу. Інша відмінність - експертним системам, як і об'єктам, не притаманні гнучкість у прийнятті рішень і можливість "розумної" комунікації з іншими експертними системами. [1]

Отже підсумовуючи, перелічимо важливі риси мультиагентних систем, що відрізняють їх від інших програмних парадигм:

- кожен агент має власну ціль, до якої він прагне. Ця ціль може відрізнятись від цілей інших агентів системи;
- алгоритм дій агентів не програмується наперед - вони здатні автономно приймати рішення базуючись на власних налаштуваннях та умовах навколишнього середовища;
- агенти можуть обмінюватись повідомленнями, взаємодіяти між собою та домовлятись про певні рішення;
- агенти мають сенсори, якими отримують інформацію про навколишнє середовище, та здатні впливати на це середовище з допомогою актуаторів;
- жоден агент в системі не має уявлення про всю систему;
- система - децентралізована, тобто немає агента, що керує всією системою;
- кожен агент має щонайменше один власний потік виконання.

1.2 Області застосувань агентних рішень

Оскільки агенти можуть самостійно реагувати на зміни навколишнього середовища і приймати рішення про подальші дії, то з появою мультиагентних систем стало можливим розв'язувати задачі, які раніше були виключно прерогативою людини. Наприклад використання автономної системи на космічних кораблях, яка у випадку якихось проблем самостійно визначає проблему, знаходить можливі шляхи розв'язку проблеми та вирішує її. Хоча така задача здається досить тривіальною для людини, для розв'язання задач

такого типу неможливо написати чіткий алгоритм, який би міг зрозуміти комп'ютер.

Наведемо інші приклади систем, які можуть використовувати агенти, що самостійно приймають рішення :

- Моделювання фармакологічних рішень на ранніх стадіях розробки лікарських засобів.
- Діагностика раку молочної залози (більше про дослідження можна прочитати в [4]).

Інша цікава область, в якій мультиагентні системи мають значні перспективи - моделювання соціуму та соціальних процесів. Таке моделювання є важливим методом дослідження для прогнозування певних процесів та їх наслідків, перевірка гіпотез, розуміння суспільних явищ. Зрозуміло, що такі дослідження та експерименти проводились і без використання агентів, але в такому випадку неможливо змоделювати поведінку значної кількості груп з великою кількістю учасників в кожній. Наведемо декілька задач з моделювання соціальних процесів, для моделювання яких може бути використана мультиагентна система.

- Прогнозування рівня безробіття.
- Аналіз поточного трафіку для прогнозування місць, де в майбутньому можуть виникати затори.
- Прогнозування поширення епідемій.
- Моделювання споживчої поведінки.
- Моделювання роботи в команді, для виявлення недоліків в плануванні та розробці нових підходів для управління проектом.
- Моделювання дій при ліквідації екологічної катастрофи. (Dunin-Ke, Verbrugge, Slizak).

Значний поштовх в цьому у моделюванні соціальних процесів дала модель Sugarscape, побудована на основі принципів, які в 1996 році описали Джошуа Епштейном і Робертом Акстеллом в книжці *Growing Artificial Societies*. Вона використовувалась для симуляції та дослідження різних соціальних явищ таких як народжуваність, смертність, міграції, забруднення навколишнього середовища, поширення хвороб, культурні проблеми тощо.

На сьогодні існує окрема дисципліна *Agent-based social simulation*, яка займається моделюванням соціальних явищ з використанням мультиагентних систем.

1.3 Особливості побудови агентів та основні підходи до розробки таких агентів

Для початку окреслимо загальний алгоритм роботи мультиагентної системи.

Мультиагентна система складається з середовища і власне агентів. Кожен агент є частиною середовища всіх інших агентів. Стандартна взаємодія між агентом і середовищем відбувається наступним чином. Середовище і агент ініціалізується певним початковим станом. Агент обирає дію, яку він хоче виконати. Середовище відповідним чином змінюється під впливом дії агента. Агент знову обирає дію відповідно до нового стану середовища і так по колу. Про цю взаємодію можна думати як про те, що в теорії ігор називається “game against nature”. Умова виграшу агента може відрізнитися залежно від специфіки завдання. В achievement завданнях, агент “виграє”, якщо приведе середовище в один з цільових станів. При завданні типу maintenance агенту треба не допустити жодного з “програшних” станів.

Бувають також змішані варіанти (наприклад “агент переможе, якщо досягне стану a, при цьому уникаючи станів b, c, d”).

Далі надамо базове визначення понять та подамо кроки моделі у більш формальному вигляді.

Набір всіх можливих послідовностей дій позначимо як R . Тоді R_A підмножина R , де останню дію зробив агент, а R_E це підмножина R , де остання зміни залишилась за середовищем.

У більшості практичних задач, множина можливих станів навколишнього середовища неперервна, але загально кажучи, будь-яку неперервну множину станів можна подати як певну скінченну множину з наперед заданою точністю. Отже, припускаємо що ми маємо середовище зі скінченною множиною станів.

Формально можемо задати середовище трійкою

$$Env = \langle E, e_0, T \rangle, \text{ де}$$

E - скінченна множина станів середовища,

e_0 - початковий стан,

T - функція переходів між станами

Кожен агент також має скінченну кількість дій. Крім того, якщо мова йде про state-based агентів, то в кожного з них є певний внутрішній стан (множину внутрішніх станів агента позначимо як I), якого він набув в результаті попередніх взаємодій з середовищем (існує інший варіант моделювання агентів, в якому стан не зберігається всередині агента, натомість середовище є “історично залежним” (history dependent) тобто зберігає всю інформацію про попередні зміни середовища і дії агентів). Початковий стан агента, який він отримав при створенні позначимо як i_0 . Формально задамо агент як функцію, яка на вхід отримує середовище E і внутрішній стан I і повертає певну дію A .

Ag: E×I -> Ac

Якщо розглядати процес прийняття рішення про наступну дію агентом детальніше, то можна поділити його на два кроки: сприйняття середовища і безпосередньо виконання самої дії.

Функція see відповідає за сприйняття середовища E за допомогою сенсорів та повертає непорожній набір персерт (позначимо як Per).

see: E->Per

Після отримання інформації з середовища агент змінює свій внутрішній стан. Для цього використовується функцію next, що на вхід приймає поточний внутрішній стан I і інформацію з середовища Per і повертає оновлений внутрішній стан.

next: I×Per ->I

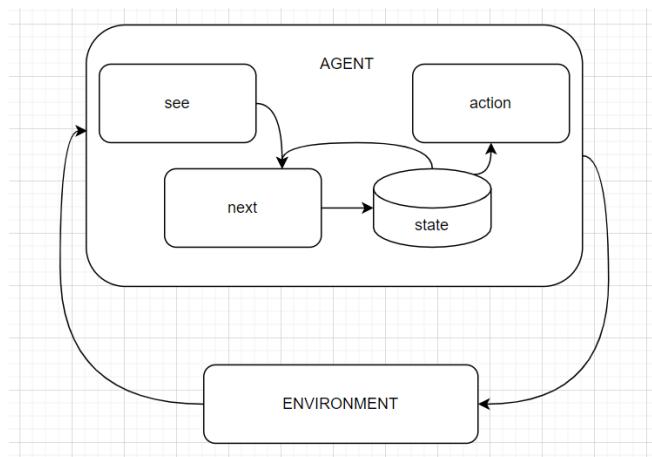


Рис. 1.2 Схема агента, що зберігає свій стан

Насамкінець функція action приймає поточний внутрішній стан і повертає дію, яку має виконати агент.

action: I->Ac

Важливим є питання, як дати зрозуміти агенту, що від нього очікується. Раніше вже згадувалось, що у випадку агентів не достатньо запрограмувати його дії явно, адже тоді він не реагуватиме на зміни

навколишнього середовища. Одним з рішень є вивести певну функцію Utility, що на вхід приймає середовище, а на вихід повертає певну оцінку - значення якості. І тоді задача агента зводиться до максимізації цієї функції.

Utility: env -> real value [1]

Умовою завершення роботи моделі може стати зведення середовища в один з кінцевих станів (виграшний чи програшний), або якась додаткова умова, наприклад, таймаут.

1.3 Основні проблеми, пов'язані з проектуванням мультиагентних систем

У попередньому розділі було перелічено риси інтелектуальних агентів - Reactivity, Proactiveness та Social ability. Далі буде обговорено складнощі в реалізації цих особливостей.

Proactiveness - орієнтованість на результат - риса, яка притаманна фактично всім програмам, адже кожна з них має прописані кроки, які мають бути виконані для досягнення поставленої цілі. Але коли виконується якась процедура програми, передбачається, що за час виконання середовище залишиться незмінним. Та на практиці в задачах, для яких використовуються агенти, в більшості випадків середовище буде постійно змінюватись.

Кожен агент має певний набір дій, які він може виконувати. Та в умовах динамічного середовища не всі з них можуть бути виконані (чи мати якийсь ефект) в конкретний момент часу. Наприклад, дія “перелізти через огорожу” може бути здійснена лише якщо поруч є огорожа, через яку потрібно перелізти, і якщо вона не вища ніж та, яку здатен подолати агент. Тому агент має не просто сліпо слідувати цілі, а й аналізувати середовище на наявність змін і підлаштовувати свої дії під ці зміни, тобто відповідати вимозі Reactivity. Написати програму, яка буде реагувати на всі зміни навколишнього

середовища теж не важко. Більш складне питання, як знайти баланс між Proactiveness та Reactivity. Адже, якщо агент просто робитиме дії для досягнення своєї цілі, незважаючи на зміни середовища, його дії будуть не завжди ефективними. У той же час, якщо агент буде постійно підлаштовуватись під мінімальні зміни середовища, то витрачатиме на це всі ресурси замість того щоб сконцентруватись на діях для досягнення поставленої задачі. [1]

На рахунок Social ability вже було згадано, що комунікація між агентами передбачає не просто обмін повідомленнями (що є тривіальною задачею), а соціальну взаємодію. Задля досягнення своєї цілі, агенту якимось чином потрібно переконати інших агентів, які можуть мати протилежні цілі, виконати певні дії в його інтересах.

Отже, при побудові агентів у першу чергу потрібно відповісти на декілька ключових запитань:

- як виділити ті риси об'єктів реального світу, які важливі при проектування відповідного агента?
- як знайти баланс між Proactiveness та Reactivity?
- як в конкретний момент вибрати дію, щоб вона була найкращим варіантом з можливих?
- як організувати комунікацію між різними видами агентів з, імовірно, різними цілями?
- як діяти, якщо невідомо які цілі переслідує інший агент?
- як автономні агенти мають координувати свої дії, якщо вони сліднують до однакових цілей?

1.4 Висновки

У даному розділі було розглянуто особливості агентно-базованого підходу та відмінності від інших методів, що дозволяють йому розв'язувати завдання, які не можуть бути розв'язані іншими способами в силу відсутності чіткої постановки задачі. Обговорено сфери використання цього підходу та описані основні поняття необхідні для побудови системи з використанням агентної парадигми.

Розділ 2. Організація колективної роботи

2.1 Важливість соціальних навиків в організації колективної роботи

У сучасному світі все більше процесів автоматизується і людина замінюється комп'ютером всюди, де це можливо і ефективно. Штучний інтелект вже перевищує можливості людини в окремих конкретних задачах. А враховуючи, що використання комп'ютера знищує ризики пов'язані з людським фактором, він має величезний потенціал у сфері медицини, судової справи, науки та промисловості.

Тим не менш, штучний інтелект все ще програє у випадках, де немає чітко окресленої проблеми і кроків вирішення задачі. У футуристичних сценаріях, у яких автори пророкують зникнення робочих місць через автоматизацію, шанс зберегти робочі місця обіцяється лише людям, чия робота пов'язана з емоціями та соціальними навичками.

Та не варто забувати, що людський інтелект все ж має значні обмеження, а складність задач, що постають, зростає. Шотландський психіатр Іоан Дірі в своїй роботі [5] вказує що хоча рівень інтелекту людини приблизно на 50% залежить від генетики, а інші 50% від подальшого розвитку, значно підвищити свої загальні розумові здібності дорослій людині досить складно. Природнім способом подолання цих обмежень є об'єднання людей в групи для виконання якогось завдання. Різний життєвий досвід, соціальний стан та підхід до виконання завдань дають можливість проведення якісних дискусій, для породження нових ідей. [6] У той же час, з переходом від одноосібного виконання завдання до командного, виникає потреба у врегулюванні конфліктів та знаходженні компромісів. Цікаво, що за дослідженнями, загальний успіх команди не так сильно корелює з індивідуальними

показниками інтелектуальних здібностей членів команди, як з соціальними навичками учасників [7]. Крім того спільна робота над задачею надає можливість учасникам ділитися досвідом та навчатись одне від одного, що має позитивні наслідки в довгостроковій перспективі. [8]

Враховуючи сказане вище, не дивно, що саме соціальні навички очолюють рейтинги важливих професійних рис 21 століття. Як приклад наведено рейтинг з дослідження Прінстонського Educational Testing Service (рис. 2.1) [9].

<i>Descriptive Statistics for Top 10 Competencies in the O*NET Skill Domain</i>			
Skill	M	SD	% important
Active listening	70	9	33
Speaking	69	11	37
Critical thinking	68	8	28
Reading comprehension	67	11	28
Judgment & decision making	61	9	8
Complex problem solving	60	9	6
Writing	60	12	12
Monitoring	60	8	6
Social perceptiveness	58	11	10
Time management	58	8	1

Note. $N = 536$. % important = percentage of occupations rating competencies as either very important or extremely important.

Рис. 2.1 Рейтинг топ 10 професійних навичок 21 століття [9]

Варто згадати, що у дослідженні PISA (Programme for International Student Assessment) за 2015 рік зазначалось зростання зарплати на 20% для посад [10], які потребують високого рівня соціальних навичок, хоча невисокого рівня, наприклад, технічних знань, що свідчить про ріст інтересу роботодавців до якостей наведених у рейтингу.

Отже, у світі високої конкуренції, важливим фактором успіху проекту є наявність команди, члени якої вміють комунікувати, надавати критичну

оцінку дій, знаходити компроміси у вирішенні проблем та продукувати ідеї та рішення, на рівень вищі ніж це можливо для окремої людини. Крім того, всередині команди мають бути правильно вибудовані внутрішні процеси, щоб дати членам команди змогу проявити свої вищезазначені якості.

3.2 Перспективи дослідження впливу соціальних навиків на роботу в команді

При прийомі на роботу (особливо в ІТ сфері) soft-skills інтерв'ю є не менш важливим етапом ніж перевірка технічних знань та умінь кандидата. На цих інтерв'ю оцінюється характер людини, її здатність працювати в команді, а також наскільки вона поділяє цінності компанії. Це важливий етап, особливо для продуктових компаній, які хвилюються за емоційний стан своїх співробітників та злагодженість команд. Часто компанія шукає людину у вже сформовану команду на певний проект, і в такому випадку оцінюється наскільки претендент має відповідні цінності й характер та чи зможе він максимально ефективно проявити себе.

Цікавим предметом дослідження є аналіз того наскільки риси характеру людини впливають на її ефективність роботи в команді. Моделювання з допомогою агентів може допомогти оцінити як люди з різними типами темпераменту співпрацюють між собою, які способи досягнення цілей обирають та виділити основні риси, необхідні для якісної комунікації.

Зупинимось конкретніше на компетенції Complex Problem Solving, що означає вміння вирішувати складні завдання. Раніше було отримано висновок, що у більшості випадків для вирішення таких проблем вже недостатньо участі однієї людини, тому на цьому моменті варто ввести поняття Collaborative Problem Solving. PISA визначає це поняття наступним чином: “Collaborative problem-solving competency is the capacity of an individual to effectively engage

in a process whereby two or more agents attempt to solve a problem by sharing the understanding and effort required to come to a solution and pooling their knowledge, skills and efforts to reach that solution.” [11]

У практичній частині цієї роботи описано систему, з допомогою якої можна змодельовати це явище. Групі агентів буде надано завдання, яке вони матимуть розв’язати самостійно, а потім спробують покращити результат шляхом дискусії. Також буде розглянуто випадок, який ілюструватиме як змінюється шанси на успішне вирішення завдання, коли частина агентів не зацікавлена в пошуку рішення.

1.3 Висновки

У цьому розділі було розглянуто важливість соціальних навиків як ключових професійних навиків 21 століття. Також було обговорено перспективи використання агентно-базованого підходу для їх дослідження.

Розділ 3. Опис створеної системи для моделювання колективної роботи

3.1 Загальний опис системи

У рамках дослідження способів моделювання колективної роботи, було розроблено систему, що моделює процес вирішення тесту командою, що складається з представників з різним типом темпераменту.

Перед командою агентів ставилось завдання подібне до “Wilderness Survival: A Consensus-Seeking Task”, розробленого Дональдом Т. Сімсоном. Це одна з класичних задач, що використовується на тренінгах для покращення взаємодії між членами колективу. Сенс задачі в тому, що кожному учаснику дається аркуш з описом критичних ситуацій, що можуть трапитись, якщо людина заблукала в лісі та трьома варіантами відповідей до кожного питання. Кожен учасник має самостійно відповісти на запитання, після чого учасники об'єднуються в команди і мають представити спільні відповіді. Очікується, що в результаті команди мають бути кращими ніж результати окремої людини. Варто зауважити, що для вирішення цієї задачі не обов'язково володіти специфічними знаннями, особливість завдань в тому, що до правильної відповіді можна дійти спираючись лише на логіку і життєвий досвід. Детальніше з умовою завдання можна ознайомитись за посиланням [11].

Для того щоб представити людину як агента в моделі, було використано наступні характеристики.

talkativeness (tl) - ступінь того наскільки агент готовий представити свої відповіді команді на певному кроці

agreeableness (ag) - ступінь того наскільки агент готовий сприймати і погоджуватись на ідеї інших членів команди

knowledge_sharing (ks) - ступінь того який відсоток спірних відповідей з тесту агент готовий обговорити за один крок

mimicry (mm) - риса яка відповідає за те, чи притаманно агенту погодитись на варіант на який погодилась більшість. Якщо це значення вказано, то означає відсоток узгодженості, який потрібен агенту щоб змінити свою думку. Наприклад, агент змінить думку на користь варіанту іншого агента, якщо за нього голосують $> 70\%$ інших агентів.

critical_thinking (cr) - визначає здатність агента критично мислити, а отже приймати правильні рішення. [6]

При запуску програми кожен агент самостійно “відповідає” на питання. Відсоток правильних відповідей агента корелює з рівнем його **critical_thinking** з похибкою 15% у більший чи менший бік. Після запуску моделі кожному агенту по черзі надається право ходу. Агент може або показати свій список відповідей іншим агентам, або відмовитись показувати його на поточному кроці взагалі. Якщо агент вирішив показати свій список всі інші агенти вирішують чи готові вони щось змінювати в своїх відповідях на поточному кроці. Якщо так, кожна пара агентів вирішує скільки питань вони готові обговорити. Питання обираються випадковим чином з тих, по яких у агентів ще немає згоди. Агент може змінити свою відповідь на ту, що пропонується, або залишити свою. Деякі агенти можуть повністю погодитись на відовіді що показує інший агент, не порівнюючи питання по одному, якщо бачать, що цей варіант підтримує значна частина інших агентів. Зупинка відбувається, якщо всі агенти досягли згоди (не обов’язково при цьому буде знайдено 100% правильних відповідей), або якщо останні 3 кола (кожен агент мав змогу запропонувати свої відповіді тричі) не було ніяких змін. Для кращого розуміння алгоритму дій агентів при прийнятті рішення можна скористатися схемою зображеною на рисунку 3.1.

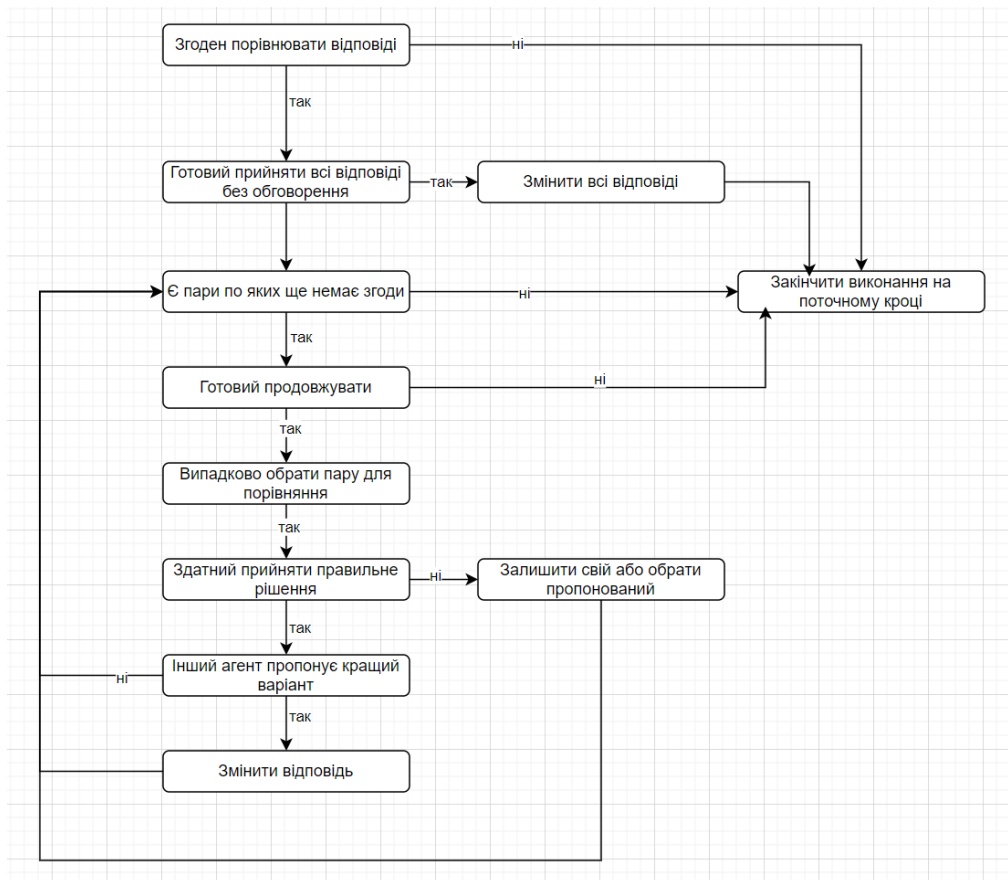


Рис. 3.1 Схема алгоритму прийняття агентом рішення про обмін

3.2 Аналіз результатів

Для загального аналізу впливу значень параметрів агента на загальний результат було проведено набір тестів. Кожному з параметрів *talkativeness*, *agreeableness*, *knowledge_sharing* та *critical_thinking* по черзі присвоювались значення з набору [0.1, 0.5, 0.9], а для *mimicry* з набору [None, 0.5, 0.9]. Для кожної комбінації параметрів здійснювалось по 50 запусків моделі. Умовами зупинки були досягнення компромісу, відсутність змін протягом 3 циклів, або якщо кількість циклів перевищувала 50. З повними результатами можна

ознайомитись у файлі results.csv практичної частини або за посиланням на репозиторій [13].

	A	B	C	D	E	F	G	H	I
1	talkativeness	agreeableness	knowledge_sharing	mimicry	critical_thinking	equal	loop	timeout	percent
22	0.1	0.1	0.9	0.1	0.9	0.1	3	47	0
23	0.1	0.1	0.9	0.5	0.1	0.1	0	35	15
24	0.1	0.1	0.9	0.5	0.5	0.5	0	41	9
25	0.1	0.1	0.9	0.5	0.9	0.9	0	50	0
26	0.1	0.1	0.9	0.9	0.1	0.1	0	30	20
27	0.1	0.1	0.9	0.9	0.9	0.5	0	43	7
28	0.1	0.1	0.9	0.9	0.9	0.9	1	49	0
29	0.1	0.5	0.1	0.1	0.1	0.1	0	12	38
30	0.1	0.5	0.1	0.1	0.5	0.5	7	18	25
31	0.1	0.5	0.1	0.1	0.9	0.9	42	8	0
32	0.1	0.5	0.1	0.5	0.1	0.1	0	12	38
33	0.1	0.5	0.1	0.5	0.5	0.5	12	25	13
34	0.1	0.5	0.1	0.5	0.9	0.9	40	10	0
35	0.1	0.5	0.1	0.9	0.1	0.1	0	16	34
36	0.1	0.5	0.1	0.9	0.5	0.5	9	17	24
37	0.1	0.5	0.1	0.9	0.9	0.9	39	11	0

Рис. 3.2 Приклад результатів роботи моделі

Розглянемо кожну рису агента окремо, без врахування контексту у вигляді значення інших параметрів і порахуємо середнє значення імовірності кожного з можливих результатів. У результаті отримуємо дані зображені в таблиці на рисунку 3.3.

equal					
	talkativeness	agreeableness	knowledge_sharing	mimicry	critical_thinking
0.1/None	44.27%	42.08%	66.42%	72.26%	58.49%
0.5	83.08%	83.84%	73.80%	73.63%	77.63%
0.9	92.19%	93.62%	79.32%	73.65%	83.42%
loop					
	talkativeness	agreeableness	knowledge_sharing	mimicry	critical_thinking
0.1/None	43.01%	33.57%	17.26%	15.72%	14.83%
0.5	4.12%	9.70%	15.48%	17.17%	16.28%
0.9	0.57%	4.44%	14.96%	14.81%	16.58%
timeout					
	talkativeness	agreeableness	knowledge_sharing	mimicry	critical_thinking
0.1/None	12.72%	24.35%	16.32%	12.02%	26.67%
0.5	12.80%	6.47%	10.71%	9.20%	6.08%
0.9	7.24%	1.94%	5.73%	11.54%	0.00%

Рис. 3.3 Імовірність результатів за певних значень параметрів

Відповідно до поданих розрахунків, важливими показниками в даному алгоритмі виявилися *talkativeness* та *agreeableness*, тобто бажання агентів ділитися своєю думкою та сприймати варіанти інших агентів. Високі значення хоч одного з цих параметрів гарантує 90% імовірності досягнення компромісу, що майже вдвічі більше за умови низьких значень цього параметру. Тим не менш компроміс не завжди означає знаходження 100% правильних відповідей. За точність результатів відповідає *critical_thinking*, хоча безпосередньо на шанс знаходження порозуміння в команді він майже не впливає. Значення *knowledge_sharing* здебільшого впливає на швидкість знаходження порозуміння, зокрема за низьких значень інших параметрів може призвести до таймауту. Параметр *mimicry* взагалі майже ніяк не впливає на загальний шанс отримати успішний результат, виграш від його наявності складає близько відсотка.

Також таблиця підтверджує і інтуїтивне припущення, що високі значення показників *talkativeness* та *agreeableness* зводять до мінімуму шанс на “зациклення” - стану, коли агенти перестають обговорювати і змінювати свої відповіді. До таймауту з великою долею імовірності призведуть низькі показники *critical_thinking* агентів.

Оскільки в реальному житті значення рис членів команди буде різним, проаналізуємо як взаємодіють між собою різні комбінації. Розглянемо команду, частина членів якої не зацікавлена в обговоренні. Представимо відповідних агентів за допомогою наступного набору параметрів.

agent_type_1: *talkativeness*: 0.7, *agreeableness*: 0.7,
knowledge_sharing: 0.7, *mimicry*: None, *critical_thinking*: 0.5
agent_type_2: *talkativeness*: 0.1, *agreeableness*: 0.1,
knowledge_sharing: 0.1, *mimicry*: None, *critical_thinking*: 0.5

Протестуємо яка мінімальна кількість активних агентів необхідна, щоб команда з 15 членів досягала компромісу в більшості випадків. Для тестування запусимо модель 100 раз. Результати зображені на рисунку 3.4 (перша таблиця). З них видно, що за таких значень параметрів, щоб досягти компромісу принаймні половина агентів групи має мати бажання приймати

first_type_num	second_type_num	equal	loop	timeout
0	15	0	50	0
1	14	0	34	16
2	13	4	35	11
3	12	11	39	0
4	11	22	28	0
5	10	24	26	0
6	9	23	27	0
7	8	27	23	0
8	7	32	18	0
9	6	32	18	0
10	5	37	13	0
11	4	34	16	0
12	3	46	4	0
13	2	41	9	0
14	1	33	17	0
15	0	50	0	0

first_type_num	second_type_num	equal	loop	timeout
0	15	0	100	0
1	14	1	69	30
2	13	24	76	0
3	12	41	59	0
4	11	47	53	0
5	10	55	45	0
6	9	63	37	0
7	8	69	31	0
8	7	74	26	0
9	6	87	13	0
10	5	79	21	0
11	4	80	20	0
12	3	89	11	0
13	2	89	11	0
14	1	80	20	0
15	0	100	0	0

Рис. 3.4 Приклади результатів при запуску моделі з агентами з високим (first_type) та низьким (second_type) рівнем knowledge_sharing

участь в обговоренні. При збільшенні значень параметрів першого агента на 20%, успіх досягається якщо третина команди проактивна (рис. 3.4 – друга таблиця).

3.3 Опис інтерфейсу

Для надання можливості експериментувати з різними налаштуваннями в ручному режимі було створено графічний інтерфейс (рис. 3.5).

Користувачеві пропонуються кнопки для додавання агентів з наперед заданими типами характеру. Умовно вони названі відповідно до класифікації Гіппократа: сангвініки, холерики, флегматики, меланхоліки. Ініціалізація

параметрів відбувалась орієнтуючись на характерні риси характеру представників, наприклад сангвінікам притаманна висока реактивність та урівноваженість, тоді як флегматикам низька реактивність та пасивність. Звісно складно описати реальні особистості за допомогою декількох параметрів, тому така класифікація і значення більш ілюстративна. Але в користувача є можливість міняти значення для кожної категорії (рис. 3.6). Поле `critical_thinking` в даному випадку не обов'язкове

для заповнення, адже рівень інтелектуальних здібностей людини не залежить від характеру. Якщо поле залишити порожнім, `critical_thinking` буде розставлено випадковим чином. Для більшої реалістичності значення генеруються за нормальним розподілом у межах від 0 до 1

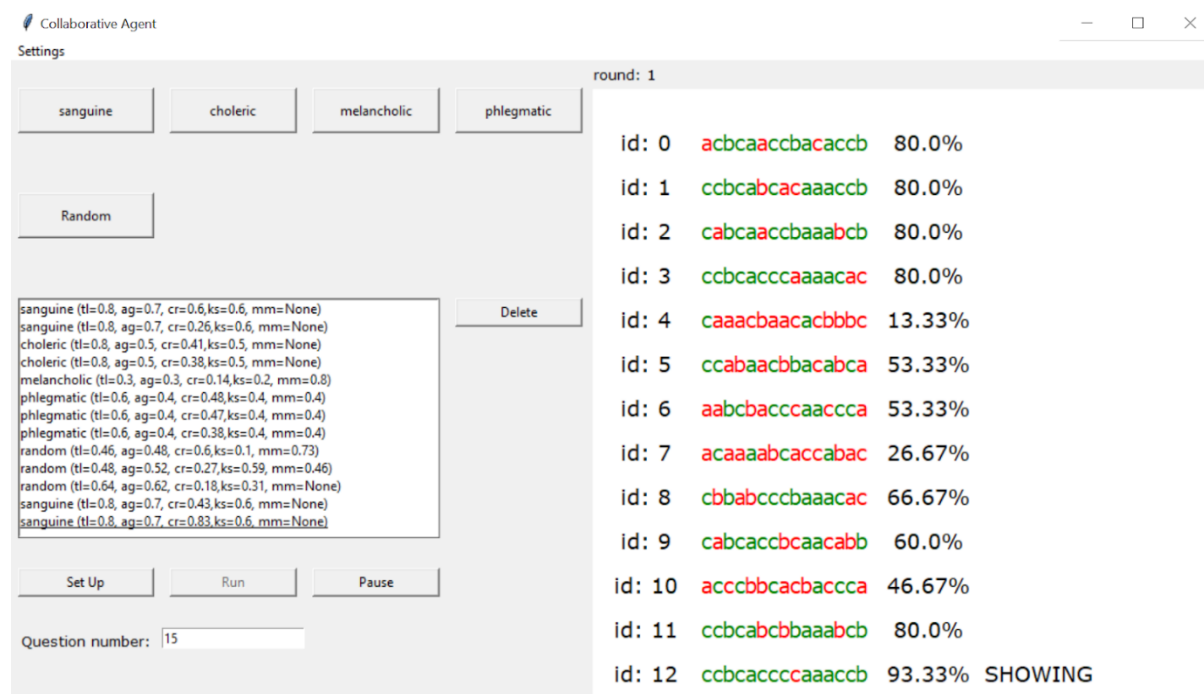
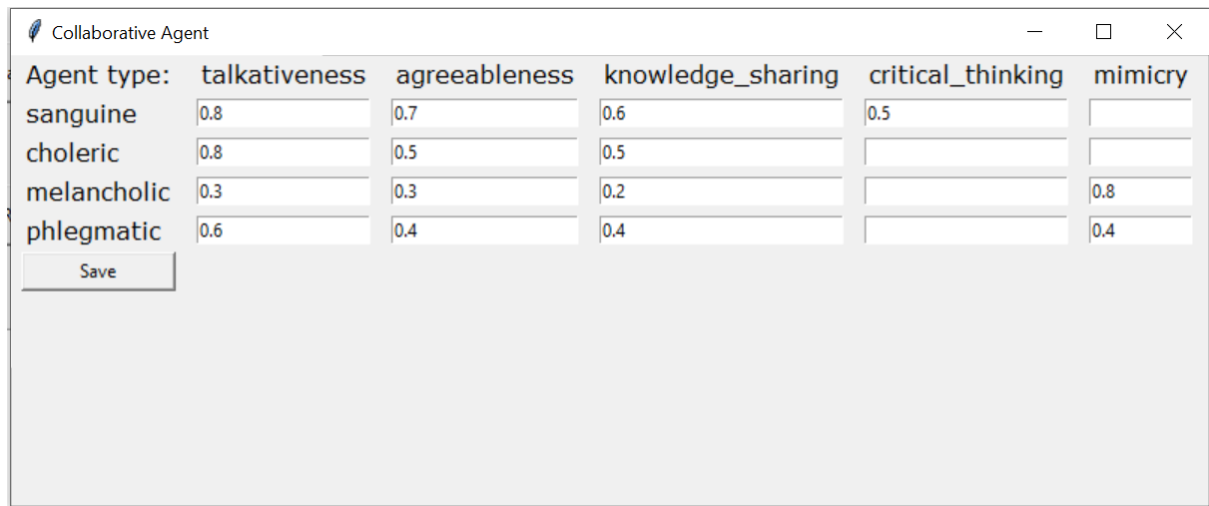


Рис. 3.5 Графічний інтерфейс програми

При генерації агента кнопкою `Random` випадковим чином розподіляються всі риси, а не лише `critical_thinking`.

Також на інтерфейсі присутні кнопки SetUp для генерування агентів з початковими відповідями, Run для запуску “обговорення” результатів, та Pause для зупинки “обговорення”. На демонстративній зоні зображено поточні відповіді агента з підсвічуванням правильних і неправильних відповідним кольором, а також відсоток правильних відповідей (агент не знає наскільки його варіант правильний, він може орієнтуватись лише на відповіді інших агентів).



Agent type:	talkativeness	agreeableness	knowledge_sharing	critical_thinking	mimicry
sanguine	0.8	0.7	0.6	0.5	
choleric	0.8	0.5	0.5		
melancholic	0.3	0.3	0.2		0.8
phlegmatic	0.6	0.4	0.4		0.4

Save

Рис. 3.6 Вікно редагування параметрів

3.4 Обґрунтування вибору технологій

На противагу класичним інструментам, спеціально призначеним виключно для моделювання різноманітних процесів (наприклад NetLogo), при виконанні цієї задачі було обрано мову Python та фреймворк Mesa. Мотивацією для цього було бажання дослідити альтернативні способи вирішення таких задач.

Фреймворк Mesa розповсюджується під Apache2 ліцензією і має базовий функціонал для реалізації агентів, що складається з окремих модулів, що згруповані в три категорії.

1. **Modeling** надає класи моделі та агента, об'єкт `schedule`, що дозволяє визначати послідовність, з якою будуть діяти агенти і простір в якому вони будуть взаємодіяти.

2. **Analysis**, що дозволяє збирати інформацію в процесі виконання, а також запускати модель декілька раз з різними параметрами, щоб дослідити їх вплив.

3. **Visualization** для візуалізації роботи моделі в браузері з використанням JavaScript. [12]

Функціонал деяких з цих модулів можна перевизначити, або взагалі використати для аналітики чи візуалізації інші інструменти, що наявні в Python. Зокрема в даній роботі для візуалізації було обрано бібліотеку Tkinter - стандартну бібліотеку для побудови графічного інтерфейсу в Python.

3.4 Перспективи та недоліки

Дана модель демонструє базові принципи взаємодії групи агентів при розв'язанні колективної задачі. Система дає можливість маніпулювати агентами з різними налаштуваннями, але запропонована базова класифікація потребує доробки, адже поточні параметри для різних типів агентів були виставлені спираючись на відчуття, а не на підтверджені соціологічні дослідження. А отже є радше демонстративним прикладом.

Щодо роботи алгоритму, цікавим є факт, що наразі, якщо на якомусь кроці виявляється що жоден агент не голосує за правильну відповідь до якогось питання, то ця відповідь вже не може бути досягнута, агенти

обговорюватимуть і шукатимуть компроміс між неправильними варіантами, а отже 100% точність стає неможлива. Необхідний глибший аналіз того наскільки така поведінка агентів відповідає поведінці людей в реальному житті. Можливо варто додати можливість агентам змінювати свою думку на протилежну до тої, що вони обговорюють.

Також можна додати нові риси агентам, наприклад параметр лідерських якостей та красномовства, та зробити щоб значення `knowledge_sharing` (бажання продовжувати обговорення) змінювалось по мірі тривалості дискусії, а не визначалось наперед.

3.5 Висновки

У даному заключному розділі було описано модель, створену з використанням агентно-базованого підходу. Проаналізовано вплив параметрів на успіх роботи команди. Представлено інтерфейс для проведення експериментів, що можуть допомогти краще зрозуміти, як агенти з різними показниками взаємодіють між собою. Також було описано недоліки системи та шляхи її вдосконалення.

Висновки

У рамках даної роботи досліджувалась важливість соціальних навиків людей у командній роботі за допомогою моделювання з використанням агентно-базованого підходу. Було підтверджено тезу, що успіх команди більшою мірою залежить від вміння комунікувати, ніж від інтелектуальних здібностей окремих членів.

Було створено систему, що дозволяє експериментувати, включаючи власні типи агентів, що відрізняються за характером, інтелектуальними здібностями та мотивацією, та досліджувати як склад команди впливає на результат.

Джерела

1. Wooldridge, M. (2001). An introduction to multi-agent systems. Chichester, UK: John Wiley & Sons.
2. Wooldridge, M. and Jennings, N.R. (1995) Intelligent Agents Theory and Practice. The Knowledge Engineering Review, 10, 115-152.
3. Lilian Weng and Filippo Menczer, Computational Analysis of Collective Behaviors via Agent-Based Modeling
4. Malgwi YM, Wajiga GM and Garba EJ, Multi-Agent Based Diagnostic Model for Breast Tumour Classification, 2019
5. Deary IJ. Intelligence: A Very Short Introduction. Oxford, UK: Oxford University Press; 2001.
6. Bergner Y., Andrews J.J., Zhu M., Gonzales J.E., (2016) Agent-Based Modeling of Collaborative Problem Solving
7. Woolley AW, Chabris CF, Pentland A, Malone NHT. Evidence for a Collective Intelligence Factor in the Performance of Human Groups. Science 2010; 333: 686{688.
8. Yen J., Fan X., Sun S., Wand R., Chen C., Kamali K., Miller M., Volz R.A, On Modeling and Simulating Agent Teamwork in CAST
9. Burrus, J., Jackson, T., Xi, N., & Steinberg, J. (2013). Identifying the most important 21st century workforce competencies: An analysis of the occupational Information network (O* NET) (Research Report No. RR-13-21). Princeton, NJ: Educational Testing Service.
<http://dx.doi.org/10.1002/j.2333-8504.2013.tb02328.x>
10. PISA 2015 Results (Volume V) Collaborative Problem Solving

11. Donald T. Simpson, "Wilderness Survival: A Consensus-Seeking Task", ст 13
12. Документація Mesa. Електронний ресурс. Доступ 10.05.2021
<https://mesa.readthedocs.io/en/master/overview.html>
13. Посилання на репозиторій проекту
<https://github.com/SandraRadz/Collaborative-Agents>