

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра інформатики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: **«ДОСЛІДЖЕННЯ МЕТОДОЛОГІЇ РОЗРОБКИ ЕКСПЕРТНИХ СИСТЕМ У РІЗНИХ СФЕРАХ»**

Виконав: студент 4-го року
навчання,

Освітньої програми «Комп'ютерні
науки», 122

Ладошка Артур Сергійович

Керівник

Курочкін А.В.

Рецензент

Салата К.В.

Кваліфікаційна робота захищена
з оцінкою

Секретар ЕК

« ____ » _____

20 ____ р.

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики
к.ф-м.н. доц Гороховський С.С.

“ ____ ” _____ 2023 року

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
для кваліфікаційної роботи

студента Ладоски Артура Сергійовича факультету інформатики 4-го курсу

Тема: Дослідження методології розробки експертних систем у різних сферах

Зміст ТЧ до кваліфікаційної роботи:

1. Індивідуальне завдання
2. Календарний план
3. Зміст
4. Анотація
5. Вступ
6. Методології розробки ЕС
7. Сфери застосування ЕС та відмінності у їх розробці
8. Розробка ЕС у медичній сфері
9. Висновки
10. Список використаної літератури

Дата видачі „ ____ ” _____ 2023 р.

Керівник Курочкін А.В. _____
(підпис)

Завдання отримав _____
(підпис)

Календарний план виконання роботи:

№	Назва етапу	Дата	Примітка
1	Отримання теми кваліфікаційної роботи	29.06.2022	
2	Пошук тематичної літератури	02.12.2022	
3	Ознайомлення з літературою	10.12.2022	
4	Аналіз теоретичних засад теорії штучного інтелекту	03.12.2022	
5	Аналіз теоретичних засад експертних систем	16.01.2023	
6	Аналіз методологій розробки експертних систем	30.01.2023	
7	Дослідження сфер та галузей для експертних систем	06.02.2023	
8	Проектування власної експертної системи	13.02.2023	
9	Створення бази знань	27.02.2023	
10	Програмна реалізація логічного виведення	14.03.2023	
11	Створення графічного інтерфейсу	28.03.2023	
12	Створення презентації	08.05.2023	
13	Попередній захист роботи	12.05.2023	
14	Здача роботи в електронному вигляді	21.05.2023	
15	Захист роботи	30.05.2023	

Перелік прийнятих скорочень

ЕС – експертна система

ШІ – штучний інтелект

БЗ – база знань

БД – база даних

Зміст

Анотація	6
Вступ.....	7
1. Методології розробки ЕС.....	10
1.1 Поняття ЕС.....	10
1.2 Будова ЕС.....	12
1.3 Характеристики ЕС.....	13
1.4 Історія ЕС.....	14
2. Сфери застосування ЕС та відмінності у їх розробці.....	16
2.1 Методології розробки ЕС.....	16
2.2 Аспекти проектування ЕС.....	17
2.3 Сфери застосування ЕС та предметні області.....	21
2.4 Приклади ЕС.....	22
3. Етапи розробки ЕС.....	25
3.1 Ідентифікація.....	25
3.2 Концептуалізація.....	26
3.3 Проектування.....	27
3.4 Реалізація.....	28
3.5 Тестування.....	29
4. Розробка ЕС у медичній сфері.....	31
4.1 Визначення мети застосунку.....	31
4.2 Вибір засобів розробки.....	31
4.3 Розробка застосунку.....	32
Висновки	34
Список використаної літератури	35

Анотація

У роботі розглянуто загальні методології розробки програмних проектів та систем штучного інтелекту, висвітлена тема, що саме таке експертні системи та відповідно як до їх розробки можна адаптувати існуючі методології. Проведений порівняльний аналіз особливостей розробки експертних систем у різних галузях та сферах життєдіяльності. Розроблено експертну систему, головна функція якої – діагностування поширених хвороб за симптомами, віком та статтю пацієнта.

Ключові слова: експертна система, методологія розробки, інтелектуальні системи, штучний інтелект, база знань.

Вступ

У епоху поширення інтелектуальних систем, та дедалі ширшої діджиталізації різноманітних галузей, набули поширеності ЕС, які використовуються для автоматизації процесу прийняття рішень, оптимізуючи робочий процес у цих галузях та полегшуючи роботу вузьких спеціалістів. Дослідження ЕС пов'язане з цілою сферою ШІ, його підгалуззю інтелектуальними інформаційними системами. Згідно з опитуванням О'Ріеллі проведеним за 2022 рік, розподіл проектів, що знаходяться у розробці та пов'язані з ШІ, по галузях використання, 17% проектів ШІ пов'язані з технологіями виробництва товарів промислового призначення, 15% - з фінансовою сферою, 9% - з галуззю охорони здоров'я та медицини, 8% - зі сферою освіти, 6% - з наданням адміністративних послуг різними державними структурами та органами місцевого самоврядування, 5% - з телекомунікаційною сферою, по 4% - з виробництвом та роздрібною торгівлею, по 3% - зі сферою безпеки, засобів масової інформації та енергетикою, і 22% інших різноманітних областей застосування, що говорить про багатополярність застосування інтелектуальних систем. Ці знання знадобляться розробникам ЕС, інтелектуальних інформаційних систем та інтелектуальних пошукових систем, які будуть залучатися до розробки систем у вузьких галузях знань. Питання методології розробки ЕС було розглянуто дослідником Чіп Гюен у її праці "Designing Machine Learning Systems".

Метою мого дослідження є з'ясування основних методологій розробки ЕС, опис галузей науки та сфер діяльності людини, для яких актуальна розробка ЕС, з'ясування сучасних тенденцій ШІ для написання власної ЕС для сфери медицини. Для досягнення цієї мети виконані наступні пункти наукового завдання:

1. Визначити поняття ЕС, їх будову
2. Охарактеризувати області застосування ЕС
3. Визначити методології розробки ЕС

4. Спроекувати та реалізувати ЕС у медичній галузі, основною функцією якої є діагностування хвороб за симптомами.

Об'єктом мого дослідження стали ЕС та підходи і методи їх розробки. Я використав метод пошуку актуальних методологій розробки ЕС, систематизував та класифікував результати пошуку, змодельовав та шляхом написання власної ЕС реалізував поставлене завдання.

Робота складається з чотирьох розділів.

У першій частині роботи зроблено вступ до поняття ЕС, її структуру та особливості, проведено пошук різних підходів до методології розробки програмного забезпечення, та їх адаптації для написання ЕС, історію ЕС.

У другій частині описані сфери діяльності людини у сьогоденні, галузі науки, які є замовниками ЕС, адже тенденції до цифрової трансформації державного сектору, використання інформаційних технологій у агробізнесі, промисловому виробництві, та сфері послуг. Виконано порівняння особливостей розробки ЕС у сферах, у рамках структури БЗ та методів логічного виводу. Також наводиться історія розвитку ЕС, для того аби прослідкувати хід тенденцій, використання нових технологій та практик, й оглянуто тенденції й перспективи подальшого розвитку

У третій частині розглянуто детально життєвий цикл розробки ЕС, його 5 основних фаз, детальну взаємодію експерта, інженера та розробника.

У четвертій, практичній частині було розроблено ЕС, із застосуванням мови програмування Java, бібліотеки розпізнавання тексту українською мовою, графічної бібліотеки. За допомогою додатку, який розроблений для використання на робочих комп'ютерах спеціалістів та користувачів програмного продукту. Він включає усі складові ЕС, БЗ, реалізовану за допомогою, інженер знань використовуватиме графічний інтерфейс програми щоб передати в неї знання, отримані від висококваліфікованого експерта у медицині, в нашому випадку, в терапії, адже за наявними симптомами, необхідно визначити діагноз,

спрямувати на медичні обстеження й аналізи, та охарактеризувати потенційні методи лікування й подальшої її профілактики, якщо хвороба відноситься до категорії недугів з легким перебігом.

1. Введення в експертні системи

1.1 Поняття ЕС

ЕС - це програма для комп'ютера, яка виконує операції зі знаннями в певній предметній області з метою розв'язання задачі. Вона може повністю перебрати на себе функції, виконання яких зазвичай потребує залучення досвіду фахівця, або відігравати роль асистента для людини, яка приймає рішення. Система, що потребує ухвалення рішення, може отримати його безпосередньо від програми або через проміжну ланку - людину, яка комунікує з програмою. ЕС допомагає спеціалісту підвищити ефективність його роботи, та він може домогтися за її допомогою результатів вищої якості. При їх проектуванні для ефективності варто приділити увагу правильному розподілу функцій між людиною і машиною. ЕС є одним з напрямів ШІ, дослідження якого сконцентровані на розробці та впровадженні комп'ютерних програм, здатних емулювати ті галузі діяльності людини, що потребують мислення, певної майстерності та накопиченого досвіду.

ЕС вважається повноцінною, якщо виконується ряд умов. По-перше, програма має володіти знаннями, однак здатність виконувати певний алгоритм, наприклад діставати елементи з колекції за фільтром - недостатня для виконання цієї вимоги. По-друге, знання, якими володіє програма, мають бути сконцентровані у межах певної предметної області, знання припускають певну організацію та інтеграцію - тобто окремі відомості мають співвідноситися один з одним й утворювати логічний ряд, в якому одна ланка тягне за собою наступну. По-третє, з цих знань має безпосередньо впливати розв'язання задачі, за аналогією, отримання доступу до певної оперативної документації - не те саме, що отримання у

своє розпорядження фахівця (або програми), здатного впоратися з проблемою, що виникла.

Функції ЕС:

- Отримання знань - це передача потенційного досвіду розв'язання проблеми від деякого джерела знань і перетворення його на вигляд, що дає змогу використовувати ці знання в програмі, шляхом співбесіди інженера знань з експертом предметної області.
- Представлення знань – це методи асоціативного зберігання інформації, подібні до тих, які існують у мозку людини, а також засіб пошуку методів формального опису великих масивів корисної інформації з метою їх подальшого опрацювання за допомогою символічних обчислень. Формальний опис означає впорядкування в межах будь-якої мови, що має досить чітко формалізований синтаксис побудови виразів і такого ж рівня семантику, яка пов'язує сенс виразу з його формою.
- Керування процесом пошуку рішення - протягом проектування ЕС серйозна увага має бути приділена і тому, як здійснюється доступ до знань і як вони використовуються під час пошуку рішення. Знання про те, які знання потрібні в тій чи іншій конкретній ситуації, і вміння ними розпорядитися, називаються метазнання і означають знання про знання. Використання різних стратегій перебору наявних знань, як правило, чинить досить істотний вплив на характеристики ефективності програми.
- Пояснення прийнятого рішення – це розробка методів подання інформації про поведінку програми в процесі формування ланцюжка логічних висновків під час пошуку рішення. Користувачі, які працюють із системою, отримують підтвердження

того, що в кожному конкретному кроці впливає висновок, до якого дійшла програма, а інженери знань мають переконатися, що сформульовані ними знання застосовано правильно. Експерти у предметній області мають простежити перебіг міркувань ЕС і спосіб використання тих правил, які з їхніх консультацій було введено в БЗ, для коректного використання у визначеній ситуації.

1.2 Будова ЕС

ЕС складається з:

- Графічного інтерфейсу (GUI) – механізму за допомогою якого відбувається взаємодія користувача та ЕС.
- Машини логічного виводу – частини програми, що забезпечує формування логічного виведення, приймаючи рішення про те, які факти чи об'єкти задовольняють певні правила, і розташовує правила за пріоритетністю та виконує перше з них.
- БЗ – це центральне сховище спеціалізованої інформації та правил. Вона інкапсулює знання та досвід експертів у вигляді фактів, евристичних засобів і логічних правил. ЕС звертається до цієї БЗ, щоб зробити висновки і надати рішення завдання, імітуючи здатність експерта вирішувати проблеми.
- Робочої пам'яті фактів, що використовуються у правилах.
- Робочого списку правил – створеного машиною логічного виводу й впорядкованого за пріоритетом списку правил, шаблони яких задовольняють факти та об'єкти, що знаходяться в робочій пам'яті.
- Засобу пояснення – складової що дозволяє роз'яснити користувачеві хід роздумів системи.

- Засобу набуття (отримання) знань – автоматизованого способу, що дозволяє користувачу вводити знання в систему, не залучаючи для цього інженера знань

1.3 Характеристики ЕС

Одна з примітних характеристик ЕС - це імітація людського мислення, яка суттєво відрізняє їх від простих математичних моделей або комп'ютерних анімацій. Основний акцент цих систем - це моделювання стратегії вирішення проблем, яку застосовує експерт у процесі виконання своєї роботи. Іншою важливою особливістю ЕС є те, що вони не обмежуються тільки виконанням обчислювальних операцій. Вони генерують розуміння та формулюють висновки, використовуючи свою внутрішню БЗ. Важливо відмітити, що знання в цих системах кодуються на спеціалізованій мові, і вони зберігаються окремо від основного програмного коду, що відповідає за формування висновків.

Щодо роботи ЕС, їх основні інструменти - це евристичні і приблизні методи. Вони відрізняються від алгоритмічних методів тим, що не можуть гарантувати успіх, але вони використовують практичний досвід, накопичений людством для розв'язання подібних проблем. Вони називаються приблизними, тому що не вимагають повної вхідної інформації і враховують ступінь впевненості або невпевненості в правильності пропонованого рішення, бо ЕС часто працюють з реальними ситуаціями, де досвід відіграє ключову роль. Незважаючи на те, що багато програм з ШІ є дослідницькими і зосереджені на абстрактних математичних проблемах або спрощених сценаріях реального світу, ЕС мають практичне застосування в науковій та комерційній сфері. ЕС вимагають ефективної та вчасної роботи. Навіть якщо дослідницькі програми можуть виявитися повільними або

ненадійними, ЕС зобов'язані надати рішення, яке не поступається тому, що міг би запропонувати професіонал в даній області. Крім того, однією з ключових функцій ЕС є можливість обґрунтувати та пояснити своє рішення. Користувачі мають бути впевненими, що прийняте рішення є обґрунтованим. Тому при проектуванні ЕС враховуються потреби користувачів, для яких важлива прозорість роботи системи.

1.4 Історія ЕС

У 1959 році Ньюелл, Шоу та Саймон розробили систему GPS (General Problem Solver), який є узагальненням інтелектуальної діяльності людей для вирішення проблем. Вони виявили, що коли люди вирішують проблеми, мислительна діяльність включає три кроки, а саме розробка приблизного плану, відповідно до пам'яті аксіоми, теореми і плану вирішення проблеми, вирішення проблеми за планом, а також в процесі реалізації процесу вирішення проблеми, постійний аналіз методу і мети, а також перегляд плану. У 1965 році, відповідно до вимог Національного управління з аеронавтики і дослідження космічного простору, в Стенфордському університеті була успішно розроблена система DENDRAL. За попередньо введеним емпіричним правилом ця система може автоматично генерувати молекулярні структури, які можуть інтерпретувати спектральні дані. Програма є першим успішним застосунком, який використовує знання самої проблеми, а не складну технологію пошуку. DENDRAL допомагає дослідникам ШІ усвідомити, що інтелектуальна поведінка залежить не лише від методів втручання, але й від знань, які використовуються в цьому втручанні. Дослідники починають створювати програму, яка використовує кодування для представлення знань для вирішення вхідної проблеми.

З того часу в Массачусетському технологічному інституті розробили систему MACSYMA, як помічника математика, яка використовує евристики для перетворення алгебраїчних виразів. Після постійного розширення вона може розв'язувати понад 600 видів математичних задач, включаючи обчислення, роботу з матрицями, розв'язування рівнянь тощо. Успішний розвиток цих систем створює підґрунтя, щоб дана ЕС користувалась попитом в академічних та інженерних колах. Багато дослідників у процесі розробки ЕС усвідомили, що представлення знань, використання знань і набуття знань є трьома основними питаннями систем ШІ. Наприкінці 1980-х років у поле зору людей почали потрапляти ЕС на основі фреймворків. Завдяки своїй вищій здатності представляти описову та поведінкову інформацію про об'єкт, ЕС базована на фреймворках, могла вирішувати більш складні проблеми, ніж ЕС, заснована на правилах. У той же час, дослідження ЕС зіткнулося з труднощами, викриваючи дефекти систем ШІ, такі як вузькі області застосування, труднощі придбання знань, механізм міркувань і так далі. Дослідникам необхідно позбутися цієї дилеми шляхом вивчення основної точки зору та використання нових методів і теорій. У 1990-х роках ЕС зазнали суттєвих змін завдяки стрімким здобуткам у галузі машинного навчання, та його інтеграції в ЕС задля автоматизації процесу отримання знань. Крім того, поширеність мережі Інтернет, та веб-технологій дозволила реалізовувати графічний інтерфейс системи через браузері. Зараз існує різноманіття хмарних сервісів та розподілених систем, що суттєво покращує досвід користування та розробки програми.

2. Сфери застосування ЕС та відмінності у їх розробці

2.1 Методології розробки ЕС

1. Базовані на правилах - містять інформацію, отриману від людини-експерта, і представляють цю інформацію у вигляді правил. Правила використовуються для оперування даними, щоб зробити висновок і дійти до відповідного висновку. Ці висновки складаються з комп'ютерної програми, яка надає методологію для міркувань про інформацію в базі правил. Вони працюють на БД.

2. Базовані на знаннях - визначаються як "орієнтовані на людину". Вони є спробами зрозуміти та ініціювати людські знання в комп'ютерній системі. Багато таких додатків існують у сфері медичного лікування.

3. Базовані на нейронних мережах - імітують біологічну нейронну мережу. Ця концепція використовується для реалізації програмних симуляцій для паралельних процесів, які включають елементи обробки, пов'язані між собою в мережевій архітектурі.

4. Нечіткі ЕС - використовують метод нечіткої логіки, який має справу з невизначеністю. Ця модель, яка використовує математичну теорію нечітких множин, імітує процес нормального людського мислення, дозволяючи комп'ютеру поводитися менш точно і логічно, ніж звичайні комп'ютери. Цей підхід застосовується для того, щоб більш точно відповідати процесам прийняття рішень, оскільки вони не завжди є точними, істинними чи хибними, а часто включають в себе зони невизначеності.

5. Базовані на конкретних ситуаціях - адаптують рішення, життєздатні для попередніх проблем, і використовувати їх для вирішення нових проблем. У таких системах описи минулого досвіду фахівців, зображені у

вигляді кейсів, зберігаються в БД для подальшого пошуку, коли користувач стикається з новим випадком зі схожими параметрами. Таким чином, система шукає збережені кейси зі схожими характеристиками проблеми, знаходить найближчу відповідність і застосовує рішення старого випадку до поточного.

6. ЕС на основі онтологій: Використовуються для розробки систематичного аналізу знань в межах предметної області, що нас цікавить. Тому, використовуючи онтологічне моделювання, відбувається дискретизація знання про предметну область і формальний опис заданої проблеми. Потім, використовуючи механізми міркувань (на основі правил), онтологічні ЕС оперують цими знаннями, отримуючи рішення.

7. Гібридні, що використовують декілька підходів, та мають поширеність у сферах, де використовується BigData, найпоширенішою комбінацією нині є поєднання методології базованої на нейронних мережах з заснованою на онтологіях чи правилах.

2.2 Аспекти проектування ЕС

У багатьох випадках було продемонстровано, що якщо інструмент добре підходить для вирішення проблеми, пов'язаної з певним видом діяльності (наприклад, проектуванням), той самий інструмент не підходить для вирішення проблеми, пов'язаної з іншим видом діяльності (наприклад, плануванням). Щоб вирішити цю ситуацію, галузь досліджень ШІ, включаючи дослідників, розробників і постачальників програмного забезпечення, звернула увагу на розробку так званих "інструментів для конкретної галузі" (domain-specific tools). Інструмент для конкретної предметної області визначається як інструмент, призначений для розробки ЕС для вирішення конкретної задачі.

Першим аспектом побудови ЕС є визначення особливостей та елементів проектування: це передбачає визначення та класифікацію відповідних елементів або особливостей в остаточному проекті. Модель проектної задачі повинна містити групування, яке дозволить провести подальший аналіз проектних особливостей та елементів. Визначення робочих завдань і відносин пріоритету в процесі планування добре відома в будівництві і ґрунтується на створенні відповідних структур розбиття робіт. У зв'язку з цим, існують два основні підходи проектування. Перший використовує ієрархічну декомпозицію проектних завдань у трьох аспектах: фізичному - основні кінцеві елементи, системи і компоненти, організаційному (пов'язаному з робочими завданнями і з частиною організації, відповідальною за завдання), ресурсному (групує робочі завдання за типом ресурсів, які вони використовують). Робочі пакети є результатом групування простих операцій на основі декомпозиції проекту з використанням цих перспектив і є вагомими елементами проекту для цілей планування та моніторингу. Другий підхід ставить мету інтегрувати оцінку тривалості і вартості в модель, що включає інформацію про проект і майданчик. Елементи робіт, пов'язані з кожним елементом проекту, ідентифікуються планувальником за допомогою унікального коду як діяльність, що виконується для побудови цього конкретного елемента проекту. Проектні роботи створюються з об'єднанням елементів робіт і можуть бути використані для формування мережі та кошторису на основі обсягу робіт або кількісних показників, і планувальник визначає пріоритетність серед них.

Крім того, були вивчені інші системи, що стосуються проблеми автоматичного створення мереж проектної діяльності. Наприклад, GHOST (Generator of Hierarchical schedules for construction), здатний знаходити

пріоритети серед набору заданих робіт за допомогою набору критеріїв. Спочатку система створює мережу, де всі роботи виконуються паралельно; потім, застосовуючи критерії для пошуку фізичних прецедентів між роботами, використовуючи знання про будівництво та фізичні взаємозв'язки, може впорядкувати їх на наступному рівні деталізації, доки не буде отримано задовільного рішення. Незважаючи на те, що під час ієрархічного успадкування створюється надмірність, і хоча система не позбавлена інших обмежень, вона проклала шлях для подальшого розвитку інших ЕС для планування будівництва (PIPPA, OARPLAN, LIFT- і т.д.).

Для вибору технології для виконання завдань необхідно розглянути можливі пакети робочої сили та обладнання, що є в наявності. Технологічний пакет або бригада зазвичай обираються на основі знань, отриманих з кошторисів, посібників або, в більш загальному випадку, з безпосереднього досвіду планувальника. Потім необхідно визначити кількість машин або бригад обраного типу, які будуть задіяні у виконанні робіт. У цьому відношенні планувальник зазвичай може покладатися на два підходи: по-перше, він приблизно знає з попередніх проектів тривалість роботи і коригує кількість бригад, виділених для досягнення цієї тривалості, не беручи до уваги взаємодію між роботами; по-друге, за допомогою так званого балансування конвеєра, можна провести моделювання способу, в який різні бригади і типи машин можуть бути використані для виконання робіт, намагаючись досягти безперервного використання бригади. Більшість розроблених допоміжних засобів для вибору технології є імітаційними моделями процесів, як, наприклад, CYCLONE (CYCLic Operations NEtwork). Ці інструменти дозволяють швидко оцінити альтернативні технології або методи, хоча користувач повинен вручну визначити операційну мережу, і її топологія не може бути змінена під час

симуляції. Тому вони вважаються скоріше аналітичними, ніж синтетичними інструментами для планувальника.

Незважаючи на те, що оцінка тривалості робіт може розглядатися як одне з найважливіших завдань у процесі планування, вона все ще виконується на основі середніх значень продуктивності, знайдених в літературі або в документації компанії, з можливими коригуваннями, зробленими планувальниками на основі їх власного досвіду. У зв'язку з цим були розроблені ЕС для збору і використання досвіду планувальників (наприклад, MASON для оцінки тривалості будівельних робіт), які використовують ієрархічні, засновані на правилах процеси, що починаються з максимальної продуктивності бригади і враховують різні модифікації, простої, специфічні характеристики будівельного майданчика або роботи, щоб оцінити точну тривалість діяльності.

Можна виділити дві основні фази планування процесу. Перша фаза перед будівництвом або перед виробництвом включає деякі з вищезгаданих видів діяльності з планування, такі як оцінка тривалості робіт, вибір технології, і в кінцевому підсумку призводить до підготовки робочого графіка. Однак, при переході до більш пізньої фази процесу, інші комп'ютерні системи повинні використовуватися для допомоги планувальнику в підтримці та оцінці графіків проекту, щоб уникнути того, що оновлення графіку зводиться до тривіального процесу ведення архівних записів, здійснюваного інженерами-планувальниками нижчого рівня, а не до процесу перепланування, що є фундаментальним для підтримки прийняття рішень під час виконання проекту. У зв'язку з цим, як приклад ЕС, розробленої для цієї мети, можна навести PLATFORM. Маючи розклад проекту і реєструючи фактичну тривалість виконаних робіт, система визначає два типи факторів, що впливають на завершення робіт порівняно з

їх початковою оціночною тривалістю: позитивний (коротший час) або негативний (довший час). Після включення в модельний графік кожної роботи оптимістичної та песимістичної тривалості застосовуються системою для динамічного перенесення незавершених робіт в мережі, встановлюючи їх тривалість відповідно до їх оптимістичного/песимістичного результату на основі зібраних даних. Варто зазначити, що у користувача, однак, завжди залишається можливість скасувати рекомендацію системи після процесу оновлення.

2.3 Сфери застосування ЕС та предметні області

ЕС застосовуються в широкому спектрі галузей. Вони допомагають в роботі з великими масивами даних та прийнятті обґрунтованих рішень - від медицини до телекомунікацій. У медичній сфері такі системи, як CADUCEUS, MYCIN та ONCOCIN, допомагають у діагностиці, плануванні лікування, моніторингу стану пацієнтів та навчанні молодих лікарів. У сфері фінансів системи, на зразок PROMETHEE, INVESTOR та LOANALYST, дозволяють прогнозувати ринкові тренди, оцінювати кредитну спроможність, керувати інвестиційними наборами та аналізувати фінансову інформацію. В юридичному контексті, ЕС, такі як JUDITH, TAXMAN та LABYRINTH, застосовуються для аналізу судових випадків, оцінювання потенційних наслідків судових рішень, формулювання стратегій захисту та вивчення законодавства. Системи GUIDON, SOPHIE та WEST, використовуються в освіті для індивідуального навчання, розробки планів уроків та оцінки навчальних досягнень студентів. У галузі інженерії та дизайну інструменти, такі як R1/XCON, DART та STRIPS, допомагають у розробці нових продуктів, оптимізації процесів, контролі якості та оцінці

безпеки, здійснюють вибір конфігурації складних багатокomпонентних систем (наприклад, розподілених комп'ютерних систем).

ЕС, на кшталт GEMS, MEDIAS та CERES, використовуються для моніторингу екосистем, аналізу впливу людської діяльності на навколишнє середовище та розробки стратегій збереження природних ресурсів, витягують інформацію з первинних даних (для прикладу сигнали, що надходять від гідролокатора). У логістичних процесах системи, на зразок INTEGRATION, IMPRESS та LOGISTAR, оптимізують доставку, керують запасами та координують взаємодію різних ланок ланцюга поставок, реалізують планування послідовності виконання операцій, що приводять до оптимізацію логістичних шляхів. Автоматизація виробництва стала можливою завдяки таким системам, як ADEPT, MICON та SPARC, які контролюють якість продукції, планують виробничий процес, діагностують технічні проблеми та оптимізують виробничі процеси, проводять структурний аналіз складних об'єктів (хімічних сполук).

2.4 Приклади ЕС

DESIGNER (1982) допомагає в загальних процесах проектування будівлі. Ключовою характеристикою проектування інженерних систем є складність: завдання проектувальника - визначити характеристики системи, враховуючи набір необхідних функціональних цілей, які мають бути досягнуті в даному середовищі. Система працює з базовими концепціями в процесі проектування і застосовує їх до загальної задачі, щоб створити прийнятний проект. Система була розроблена в Університеті Стратклайда, Великобританія.

GOES (Graphics-Oriented Expert Shell, 1995) - це оболонка, вбудована в стандартне середовище системи прийняття рішень. Вона призначена для

зв'язування логічних процедур інженерного проектування з графічними представленнями. Вона особливо корисна для управління та документування великомасштабних робіт з проектування систем управління та автоматизації. GOES працює як інтелектуальний CASE-інструмент і полегшує завдання ідентифікації, складання і параметризації підпрограм функціональних блоків розподілених систем управління. Розроблено корпорацією CMS-CAD Inc. з Монреалю, Канади.

ORBIS (Object-oriented Rule Base Interactive System, 1994) - це інструмент моделювання ЕС, призначений для використання в різних середовищах моделювання, наприклад: автономні інтерактивні симуляції, пакетні прогони для збору статистики Монте-Карло, а також симуляції в реальному часі, в циклі. Інструмент ЕС містить об'єкти, дані, алгоритми і набори правил, специфічні для моделювання, які слугують для генерування бажаної поведінки моделювання. Важливою особливістю програмного забезпечення для розробки симуляцій ORBIS є те, що ЕС, реалізована у вигляді наборів правил, керує об'єктами, що моделюються.

GHOST - це система планування загального призначення в галузі будівництва. Як згадано у попередньому розділі, вона міркує про атрибути об'єкта та зв'язки між об'єктами для визначення проектних робіт. GHOST починає з високорівневого набору завдань і уточнює їх у підмережі більш детальних завдань. Система поєднує об'єктно-орієнтоване програмування з наборами правил.

KBLPS (Knowledge-Based Logistics Planning Shell) допомагає у розробці ЕС для планування розподілу та використання транспортних ресурсів. Вона складається з алгоритмів планування для проблем розподілу з надмірними обмеженнями і графічного інтерфейсу користувача, орієнтованого на прийняття рішень. Ця ЕС надає планувальникам

змодельовати умовний випадок , щоб побачити вплив зміненої інформації на очікувані результати. Програма може циклічно переглядати сценарії, змінюючи розподіл ресурсів, доки не буде розроблено найбільш прийнятну рекомендацію.

Instruction (1995) – ЕС що була створена у Вюрцбурзькому університеті, Німеччина, для створення інтелектуальних навчальних систем для навчання студентів-медиків діагностуванню випадків захворювання. Вона використовує дані про випадки і знання про вирішення проблем для вирішення проблем класифікації. Основна ідея системи полягає в тому, щоб представити дані про випадок і контролювати дії студента, порівнюючи їх з базовою ЕС.

SSI (Shell of Signal Interpretation, 1992), розроблена в університеті Осаки у Японії, є оболонкою для розробки ЕС інтерпретації сигналів. Оболонка є продуктом проектування двох ЕС для обробки мовних сигналів, де аналіз двох систем виявив спільні функції і модулі, застосовні до широкого кола проблем інтерпретації сигналів.

PREMON (Predictive Monitoring system, 1987) – ЕС що розроблена в дослідницькому центрі NASA Ames, використовує пристрій для здійснення моніторингу в реальному часі. PREMON була розроблена для виконання трьох взаємопов'язаних видів діяльності: причинно-наслідкове моделювання для створення прогнозів щодо поведінки фізичної системи, планування датчиків для оцінки важливості поведінки пристрою і належного розподілу ресурсів датчиків, і інтерпретація датчиків для перевірки очікуваних значень датчиків з фактичними показаннями датчиків і підняття тривоги, коли це необхідно.

3. Етапи розробки ЕС

3.1 Ідентифікація

Перш ніж розпочати розробку ЕС, важливо якомога точніше описати проблему, яку вона має вирішити. Недостатньо просто відчувати, що ЕС буде корисною в певній ситуації, команда розробників, інженерів знань та експертів повинні визначити точну природу проблеми і сформулювати чіткі цілі, які визначатимуть адаптацію ЕС для її вирішення. Спочатку інженер знань, який може бути незнайомий з даною конкретною предметною областю, читає, ознайомлюючись та аналізуючи посібники, щоб отримати базове уявлення про тему роботи. Після цього експерт з предметної області описує кілька типових проблемних станів, а інженер знань намагається виділити фундаментальні концепції з подібних випадків, щоб розробити більш загальне уявлення про призначення ЕС. Після того, як експерт описав кілька випадків, інженер розробляє опис проблеми першої версії. Може трапитись, що експерт побачить, що опис не повністю відображає проблему, тоді він пропонує внести зміни в опис і надає інженеру знань додаткові приклади, щоб проілюструвати особливі випадки.

Далі інженер знань переглядає опис, а експерт пропонує подальші зміни. Цей процес повторюється до того моменту, доки фахівець галузі не переконається, що інженер розбирається у задачі предметної області, і обидва не переконуються, що опис адекватно відображає проблему, яку має вирішити ЕС. Ця ітеративна процедура є типовою для всього процесу розробки ЕС. На кожному етапі процесу оцінюються результати і порівнюються з очікуваннями. Якщо результати не відповідають очікуванням, вносяться корективи на цьому етапі процесу, і оцінюються нові результати. Це триватиме, доки не буде досягнуто задовільних результатів. Важливо також визначити наші джерела, а саме хто має брати

участь у процесі розробки в якості експерта, чи один спеціаліст даної сфери володіє всіма необхідними знаннями, або вони розподілені між кількома фахівцями. Крім цього у інженера та експерта може виникнути проблема в комунікації, адже фахівці у вузькій галузі, як правило, користуються власним жаргоном, який важко перекласти звичайною людською мовою, а факти і правила, що лежать в основі багатьох специфічних галузей знань експерта, не можуть бути чітко сформульовані в термінах математичної теорії або детермінованої моделі, властивості якої добре зрозумілі інженеру. Статистичні моделі можуть допомогти зробити загальний довготривалий прогноз, але такі методи не працюють щодо курсів конкретних випадків на коротких часових інтервалах. Експерти не є єдиним джерелом знань потрібної сфери, тому інженеру також варто користуватись джерелами інформації, довідниками та підручниками, які максимально наближені до знань потрібного фахівця.

3.2 Концептуалізація

Після визначення проблеми, яку має вирішувати ЕС, наступний етап передбачає подальший аналіз проблеми для забезпечення розуміння її специфіки, а також загальних особливостей. На етапі концептуалізації інженер зі знань часто створює діаграму проблеми, щоб графічно зобразити взаємозв'язки між об'єктами і процесами в проблемній області. На цьому етапі необхідно розділити проблему на ряд підпроблем і відобразити на діаграмі взаємозв'язки між частинами кожної підпроблеми і взаємозв'язки між різними підпроблемами. Аналогічно попередньому етапу, стадія концептуалізації включає в себе ітеративну процедуру та повторення між інженером по знаннях та експертом предметної області. Коли обидва учасники розробки системи погоджуються з тим, що ключові поняття і

взаємозв'язки між ними були адекватно концептуалізовані, цей етап завершується. кожна стадія процесу розробки ЕС є циклічною, а також взаємозв'язки між стадіями також можуть бути циклічними. Оскільки кожна стадія процесу розробки додає рівень деталізації до попереднього кроку, то цей етап може виявити слабкі місця попереднього. Проблема з описом, створеним на етапі ідентифікації, може бути виявлена під час другого кроку концептуалізації. Можливо, було пропущено ключовий елемент опису або неправильно сформульовано мету, бо через труднощі в комунікації виявити в процесі співбесіди такого роду знання, що ґрунтуються на особистому досвіді та погано піддаються формалізації, значно складніше, ніж отримати простий перелік якихось фактів чи загальних принципів.

3.3 Проектування

На попередніх етапах не було докладено жодних зусиль, щоб пов'язати проблему предметної області з технологією ШІ, яка може її вирішити. На етапах ідентифікації та формалізації основна увага приділяється розумінню проблеми. Тепер, на етапі формалізації, проблема пов'язується з пропонованим рішенням, ЕС розглядається шляхом аналізу взаємозв'язків, зображених на етапі концептуалізації. Інженер знань починає підбирати методи, які підходять для розробки саме цієї ЕС.

Під час формалізації важливо, щоб інженер був знайомий з різними методами представлення знань та методами інтелектуального пошуку, що використовуються в ЕС, інструментальними засобами ЕС, які можуть значно прискорити процес розробки та іншими ЕС, які можуть вирішувати подібні проблеми і, таким чином, можуть бути адаптовані до поставленої задачі. Часто бажано вибрати єдину методику розробки або інструмент, який можна використовувати у всіх сегментах ЕС. Однак інженер знань

може визначити, що жодна конкретна техніка не підходить для всієї ЕС, що робить необхідним використання різних технік для різних підпроблем. Після того, як визначено, який методи буде використано, інженер розроблятиме формальну специфікацію, яка може бути використана для розробки прототипу ЕС.

Наприклад, у випадку системи, що базується на правилах, інженер знань розробляє набір правил, призначених для представлення знань, наданих експертом. Багато експертів предметної області можуть пояснити, що саме вони роблять, але не вдаючись у причину, тому одним з основних обов'язків інженера по знаннях є аналіз прикладів ситуацій і фільтрація з цих прикладів набору правил, які описують знання спеціаліста. Процес формалізації часто є найбільш інтерактивним етапом розробки ЕС, а також найбільш ресурсозатратним та трудоємним. Інженер знань повинен розробити набір правил і запитати фахівця галузі, аби правила адекватно відображали знання експерта. Експерт переглядає правила, запропоновані інженером знань, і пропонує зміни, які потім вносяться інженером до БЗ. Як і на інших етапах розробки, цей процес також є ітеративним: перегляд правил повторюється, і правила постійно вдосконалюються, поки результати не будуть задовільними. Нерідко процес формалізації складної ЕС може тривати кілька років.

3.4 Реалізація

На етапі реалізації формалізовані концепції вносяться у середовище розробки, з використанням заздалегідь визначених методів та інструментів для реалізації прототипу ЕС, що був отриманий внаслідок попередніх кроків. На практиці розробки ЕС дотримання всіх правил не гарантує, що система запрацює з першого разу, а з урахуванням попереднього досвіду

розробки ЕС для різних галузей, багів для подальшого відлагодження буде достатньо.

Якщо прототип взагалі працює, інженер знань може визначити, чи правильними були методи, обрані для реалізації ЕС, з іншого боку, інженер може виявити, що обрані методи просто неможливо реалізувати. Наприклад, може виявитися неможливим інтегрувати методи представлення знань, обрані для різних підзадач. В такому випадку, можливо, доведеться переформалізувати концепції, або навіть створити нові інструменти розробки для ефективною реалізації системи. Після того, як прототип системи буде достатньо доопрацьований, щоб його можна було реалізувати, ЕС готова до ретельного тестування, щоб переконатися, що вона робить правильні висновки.

3.5 Тестування

Шанси на те, що прототип ЕС буде працювати бездоганно під час першого тестування досить невеликі. Інженер знань не очікує, що процес тестування підтвердить, що система була побудована абсолютно правильно. Скоріше, тестування дає можливість виявити слабкі місця в структурі та реалізації системи і внести відповідні корективи. Залежно від типів проблем, що виникають, процедура тестування може вказувати на те, що система була реалізована неправильно, або, можливо, на те, що правила були реалізовані правильно, але були погано або неповно сформульовані. Результати тестування використовуються як "зворотний зв'язок" для повернення на попередній етап і коригування роботи системи. Після того, як система довела, що здатна правильно вирішувати прості завдання, експерт з предметної області пропонує складні завдання, які зазвичай вимагають значного людського досвіду. Ці більш складні тести повинні

виявити більш серйозні недоліки і надати широкі можливості для подальшого глибокого налаштування системи. Зрештою, ЕС вважається повністю успішною лише тоді, коли вона працює на рівні висококваліфікованого спеціаліста предметної області. Процес тестування не завершується доти, доки він не покаже, що рішення, запропоновані ЕС, є такими ж правильними, як і рішення, надані спеціалістом у певній галузі.

4. Розробка ЕС у медичній сфері

4.1 Визначення мети застосування.

Реалізувати програму – ЕС на основі знань, за допомогою якої користувачі (пацієнти) можуть дізнатися діагноз своєї хвороби, за заданими симптомами, враховуючи вік та стать. Крім того, система має надавати рекомендований метод лікування. Інженери знань, використовуючи графічний інтерфейс даної програми, зможуть легше вводити правила в базу знань, що прискорить процес розробки та подальшої підтримки даної ЕС.

4.2 Вибір засобів розробки

1) Мова програмування Java

2) Графічний фреймворк JavaFX

При створенні експертної системи, вибір мови програмування і фреймворку є ключовим. Java стає привабливим варіантом, в основному, через надійність і широку підтримку. Із наявністю потужного віртуального середовища виконання, Java пропонує переносність, яка підходить для розробки на різних платформах. Важливою характеристикою Java є її об'єктно-орієнтований підхід, що допомагає у створенні структурованого і легко модифікуємого коду. Це особливо цінно при розробці складних систем, таких як експертні. Безпека є ще однією суттєвою перевагою Java, оскільки вона включає в себе ряд захисних механізмів, таких як перевірка типів на етапі компіляції.

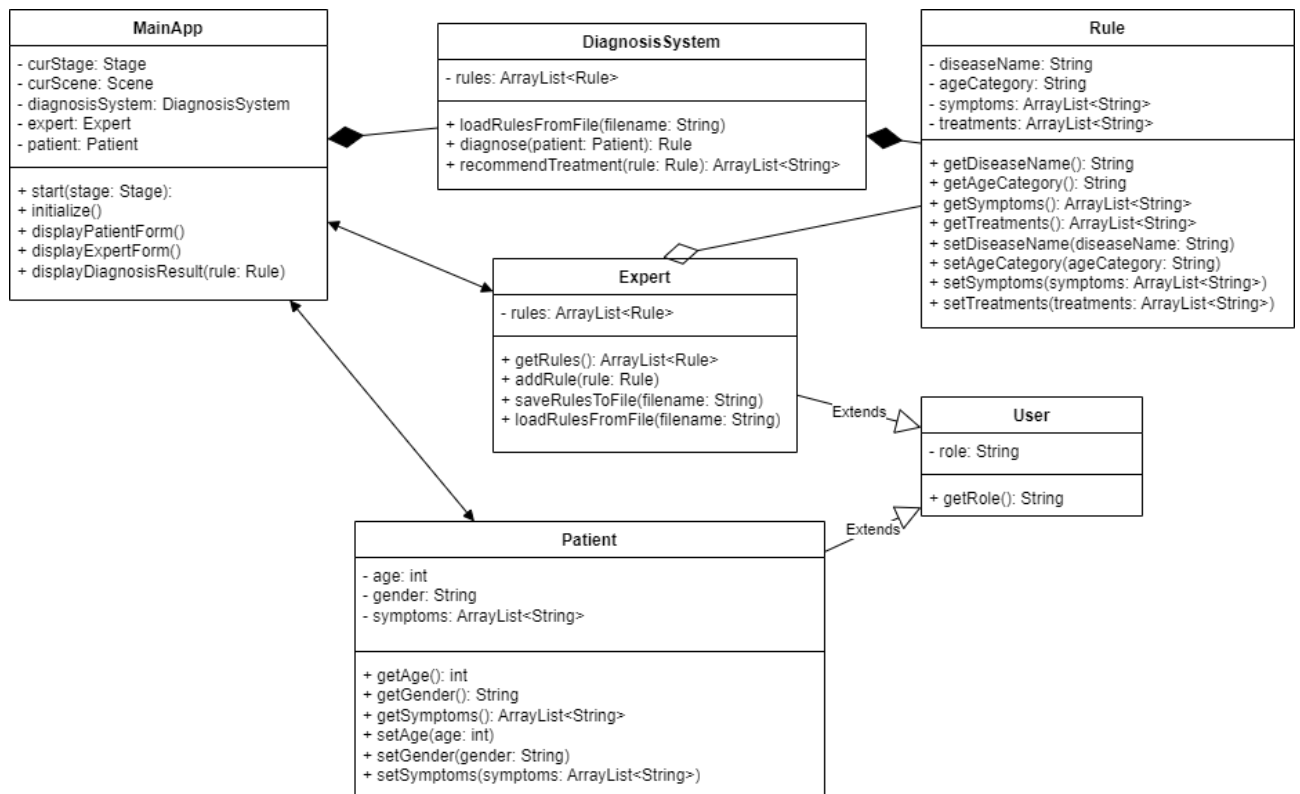
Стосовно вибору JavaFX, він надає привабливий набір інструментів для створення графічного інтерфейсу. Платформа забезпечує гнучкість в дизайні, дозволяючи створювати впізнавані і інтуїтивно зрозумілі

інтерфейси. JavaFX підтримує також використання CSS для стилізації, що додає ще більше гнучкості. Крім того, JavaFX інтегрується гладко з Java, спрощуючи процес розробки. Подієва модель в JavaFX сприяє ефективному обробленню дій користувача. Загалом, з урахуванням різноманітних переваг, які пропонують Java та JavaFX, їх комбінація стає ідеальною для розробки ефективної експертної системи.

Розроблено 4 основні класи, та 2 допоміжних – які вказують на роль користувача системи.

4.3 Розробка застосунку

Діаграма класів:



Висновки

- Реалізовано ЕС що заснована на знаннях, за допомогою мови програмування Java та графічного фреймворку JavaFX. При його побудові була використана загальна методологія проектування ЕС побудована на знаннях, хоча повноцінно всі етапи життєвого циклу неможливо пройти без залучення фахівця.
- Проведено аналіз – порівняння та ознайомлення з системами у різних галузях науки та сферах життєдіяльності
- ЕС – один з перспективних напрямків розвитку царини ІІІ, і тенденції вказують, що підхід, заснований на нейромережах стане ключовим аспектом еволюції ЕС у ХХІ сторіччі.

Список використаної літератури

- [1] Chip Huyen - Designing Machine Learning Systems. O’Rielly 2022
- [2] М.Глибовець, О.Олецький – Основи штучного інтелекту. Видавничий дім “КМ Академія”, 2002
- [3] Applying the Agile methodology to AI/ML projects – NealAnalytics [електронний ресурс] – Режим доступу до ресурсу: <https://nealanalytics.com/blog/applying-the-agile-methodology-to-ai-ml-projects/>
- [4] Agile Development Considerations for ML Projects - Microsoft Code With Engineering Playbook [електронний ресурс] – Режим доступу до ресурсу: <https://microsoft.github.io/code-with-engineering-playbook/machine-learning/ml-project-management/>
- [5] Russell, S. J., & Norvig, P. - Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited, 2016.
- [6] Luger, G.F., & Stubblefield, W.A. - Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th Edition). Addison Wesley, 2008.
- [7] Jess, the Java Expert System Shell [електронний ресурс] – Режим доступу до ресурсу: <https://www.csie.ntu.edu.tw/~sylee/courses/jess/docs/intro.html>
- [8] Java Swing vs Java FX | Know The 6 Most Awesome Differences [електронний ресурс] – Режим доступу до ресурсу: <https://www.educba.com/java-swing-vs-java-fx/>
- [9] Giarratano, J., & Riley, G. - Expert Systems: Principles and Programming, Fourth Edition. Course Technology, 2005.
- [10] Rich, E., & Knight, K. - Artificial Intelligence. McGraw-Hill, 1991.