

Міністерство освіти і науки України
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт
з дисципліни
«ДИСКРЕТНІ СТРУКТУРИ»
для студентів напрямку підготовки
6.050103 “Програмна інженерія”
усіх форм навчання

2013

Методичні вказівки до виконання лабораторних робіт з дисципліни "Дискретні структури" для студентів напрямку підготовки 6.050103 "Програмна інженерія" усіх форм навчання / Уклад. С.О. Субботін. – Запоріжжя: ЗНТУ, 2013. – 46 с.

Автор: Сергій Олександрович Субботін,
кандидат технічних наук, доцент,
професор кафедри програмних засобів

Рецензент: А.О. Олійник, к.т.н., доцент

Відповідальний
за випуск: В.І. Дубровін, зав. каф. "Програмні засоби"

Затверджено
на засіданні кафедри
"Програмні засоби"

Протокол № 11
від 4.06.2013 р.

ЗМІСТ

Загальні положення	4
Лабораторна робота № 1. Розробка семантичної мережі. . . .	5
Лабораторна робота № 2. Розробка фреймової моделі.	8
Лабораторна робота № 3. Побудова продукційної моделі. . .	10
Література	13
Додаток А. Приклади виконання елементів звітів з лабораторних робіт.	14
Додаток Б. Перелік предметних областей для побудови моделей знань	28
Додаток В. Побудова інтернет-орієнтованих експертних систем за допомогою компоненту e2glite.	29
Додаток Г. Моделювання семантичних мереж за допомогою бібліотеки SNToolbox у пакеті MATLAB	44

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Дане видання призначене для вивчення та практичного освоєння студентами усіх форм навчання дискретних структур для побудови інтелектуальних систем.

Відповідно до графіка студенти перед виконанням лабораторної роботи повинні ознайомитися з конспектом лекцій та рекомендованою літературою.

Для одержання заліку по кожній роботі студент здає викладачу цілком оформлений звіт, а також CD (DVD) –RW диск у форматі MS – DOS / Windows, перевірений на відсутність вірусів, з текстами розроблених програм, файлами програм, що виконуються, файлами даних і текстом звіту.

Звіт (приклад оформлення - див. додаток А) має містити:

- титульний аркуш (на ньому вказують назву міністерства, назву університету, назву кафедри, номер, вид і тему роботи, виконавця та особу, що приймає звіт, рік);

- мету, варіант і завдання роботи;
- лаконічний опис теоретичних відомостей;
- текст програми, що обов'язково містить коментарі;
- вхідні та вихідні дані програми;
- змістовний аналіз отриманих результатів та висновки.

Звіт виконують на білому папері формату А4 (210 x 297 мм). Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші скріплюють за допомогою канцелярських скріпок. Для набору тексту звіту використовують редактор MS Word 97: шрифт Times New Roman, 12 пунктів. Міжрядковий інтервал: полуторний – для тексту звіту, одинарний – для листингів програм, таблиць і роздруківок даних.

Під час співбесіди студент повинний виявити знання про мету роботи, по теоретичному матеріалу, про методи виконання кожного етапу роботи, по змісту основних розділів розробленого звіту з демонстрацією результатів на конкретних прикладах. Студент повинний вміти правильно аналізувати отримані результати. Для самоперевірки при підготовці до виконання і здачі роботи студент повинний відповісти на контрольні питання, приведені наприкінці опису відповідної роботи. Загальний залік студент одержує після виконання і здачі останньої роботи.

1 Лабораторна робота №1

РОЗРОБКА СЕМАНТИЧНОЇ МЕРЕЖІ

Мета роботи: Навчитися аналізувати й описувати предметну область у вигляді семантичної мережі.

Завдання до роботи

1. Ознайомитися з конспектом лекцій та рекомендованою літературою, а також додатком Д.
2. Обрати та узгодити з викладачем предметну область для побудови семантичної мережі.
3. Проаналізувати предметну область, виділити основні об'єкти області та відношення між ними в контексті поставленої мети побудови бази знань.
4. Розробити короткий опис предметної області, що містить формулювання мети і призначення бази знань, актуальність створення експертної системи для предметної області.
5. Виділити входи та виходи для експертної системи.
6. Побудувати словник для предметної області, яка аналізується, що повинен містити такі стовпці:
 - Назва об'єкта (події),
 - Визначення сутності,
 - Назва атрибута (властивості) об'єкта,
 - Допустимі значення атрибута.
7. Виконати узагальнення у предметній області, виділити (за можливістю) ієрархічні зв'язки.
8. Створити базу знань у вигляді семантичної мережі для експертної системи. Зобразити її схему вручну (показати вузли із підписаними назвами та відношення із підписаними назвами).
9. Розробити програму на мові пакету MATLAB із використанням функцій бібліотеки SNTtoolbox, що реалізує базу знань у вигляді семантичної мережі та графічно зображує її схему як ієрархію.
10. Розробити мережу запиту для семантичної мережі. Зобразити її вручну. Доповнити MATLAB програму реалізацією мережі запиту, зобразити мережу запиту програмно із круговим розташуванням вузлів. Виконати запит до семантичної мережі у пакеті MATLAB.
11. Оформити звіт з роботи.
12. Відповісти на контрольні запитання.

Зміст звіту

1. Тема та мета роботи.
2. Короткі теоретичні відомості.
3. Назва предметної області.
4. Мета створення бази знань експертної системи.
5. Конкретизація вхідних та вихідних даних.
6. Словник предметної області.
7. Узагальнення (опис узагальнюючих об'єктів предметної області).
8. Сценарії (позитивний і негативний) для тестування моделі.
9. Схема семантичної мережі, виконана вручну.
10. Схема семантичної мережі, побудована програмою.
11. Схема мережі запиту, побудована вручну.
12. Текст програми, що реалізує базу знань на основі семантичної мережі та операції з нею.
13. Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

Приклади виконання елементів звіту наведено у додатку А.

Контрольні запитання

1. Поняття інтелектуальної системи.
2. Властивості інтелектуальних систем.
3. Характеристика систем, заснованих на знаннях.
4. Архітектура експертної системи.
5. Поняття знання.
6. Відміна знань від даних.
7. Поняття екстенсіоналу та інтенсіоналу.
8. Загальні відомості про моделі представлення знань.
9. Поняття семантичної мережі.
10. Формалізація семантичної мережі.
11. Прості та ієрархічні мережі.
12. Основні типи об'єктів та зв'язків між ними.
13. Види семантичних відношень.
14. Модифікація баз знань на семантичних мережах.
15. Операція порівняння із зразком.
16. Яким чином виконується подання знань в семантичній моделі?
17. Для чого використовується операція порівняння із зразком в семантичній мережі?
18. Принципи наслідування інформації в мережній моделі?

19. Які основні типи об'єктів і відношень між ними в семантичній мережі є обов'язкові?
20. Як здійснюється пошук інформації в мережній базі знань?
21. Назвіть достоїнства та недоліки семантичної мережі.
22. За допомогою яких відношень в мережі можна виразити ієрархію об'єктів?
23. Однорідна та неоднорідна мережа, дискретна мережа. Що їх відрізняє?
24. Що зветься сценарієм?
25. За заданим висловом виділіть об'єкти–поняття (концепти), об'єкти–події, об'єкти–характеристики.
26. Бібліотека SNTtoolbox для моделювання семантичних мереж у пакеті MATLAB.

2 Лабораторна робота № 2

РОЗРОБКА ФРЕЙМОВОЇ МОДЕЛІ

Мета роботи: Навчитися аналізувати й описувати предметну область та подавати бази знань інтелектуальних систем у вигляді фреймової моделі

Завдання до роботи

1. Ознайомитися з конспектом лекцій, рекомендованою літературою та теоретичними відомостями даної роботи.
2. Обрати та узгодити з викладачем предметну область для побудови продукційної моделі (рекомендується обрати предметну область, що моделювалася у попередній роботі).
3. Проаналізувати предметну область, виділити основні об'єкти області та відношення між ними в контексті поставленої мети побудови бази знань.
4. Розробити короткий опис предметної області, що містить формулювання мети і призначення бази знань, актуальність створення експертної системи для предметної області.
5. Виділити входи та виходи для експертної системи.
6. Побудувати словник для предметної області, яка аналізується, що повинен містити такі стовпці: Назва об'єкта (події), Визначення сутності, Назва атрибута (властивості) об'єкта, Допустимі значення атрибута.
7. Виконати узагальнення у предметній області, виділити (за можливістю) ієрархічні зв'язки.
8. Побудувати фреймову модель для предметної області.
9. Оформити звіт з роботи.
10. Відповісти на контрольні запитання.

Зміст звіту

1. Тема та мета роботи.
2. Короткі теоретичні відомості.
3. Назва предметної області.
4. Мета створення бази знань.
5. Конкретизація вхідних та вихідних даних.
6. Словник предметної області.
7. Узагальнення (опис узагальнюючих об'єктів предметної області).
8. Сценарії (позитивний і негативний) для тестування моделі.

9. Опис фреймової моделі.
 10. Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз. У висновках також необхідно порівняти семантичні мережі, продукційні моделі та фреймові моделі.
- Приклади виконання елементів звіту наведено у додатку А.

Контрольні запитання

1. Формальний опис фрейму.
2. Класифікація фреймів.
3. Структури даних фрейму.
4. Процедури-демони та процедури-слуги.
5. Фреймові мережі.
6. Визначення структури фрейму як моделі подання знань про поняття.
7. Модель бази знань в поєднанні фреймового і мережного подання.
8. Для чого використовується операція порівняння із зразком в мережі?
9. Принципи наслідування інформації у фреймовій мережній моделі
10. Які типи відношень існують між фреймами в семантичній мережі?
11. Як здійснюється пошук інформації в базі знань, створеній на основі фреймів?
12. Яка різниця між фреймом-прототипом та фреймом-екземпляром?
13. За допомогою яких дій можливо виразити ієрархію фреймів?
14. Яка інформація може бути представлена у слотах фреймів?
15. Що зветься приєднаною процедурою?
16. За заданим висловом створіть фрейм-опис або рольовий фрейм поняття, події.
17. Визначте переваги та недоліки фреймів.

3 Лабораторна робота № 3

ПОБУДОВА ПРОДУКЦІЙНОЇ МОДЕЛІ

Мета роботи: Навчитися аналізувати й описувати предметну область у вигляді семантичної мережі.

Завдання до роботи

1. Ознайомитися з конспектом лекцій та рекомендованою літературою.
2. Обрати та узгодити з викладачем предметну область для побудови продукційної моделі (рекомендується обрати предметну область, що моделювалася у попередній роботі).
3. Проаналізувати предметну область, виділити основні об'єкти області та відношення між ними в контексті поставленої мети побудови бази знань.
4. Розробити короткий опис предметної області, що містить формулювання мети і призначення бази знань, актуальність створення експертної системи для предметної області.
5. Виділити входи та виходи для експертної системи.
6. Побудувати словник для предметної області, яка аналізується, що повинен містити такі стовпці: Назва об'єкта (події), Визначення сутності, Назва атрибута (властивості) об'єкта, Допустимі значення атрибута.
7. Виконати узагальнення у предметній області, виділити (за можливістю) ієрархічні зв'язки.
8. Побудувати продукційну модель для експертної системи.
9. Провести тестування продукційної моделі шляхом виконання позитивного та негативного сценаріїв.
10. Оформити звіт з роботи.
11. Відповісти на контрольні запитання.

Зміст звіту

1. Тема та мета роботи.
2. Короткі теоретичні відомості.
3. Назва предметної області.
4. Мета створення бази знань експертної системи.
5. Конкретизація вхідних та вихідних даних.
6. Словник предметної області.

7. Узагальнення (опис узагальнюючих об'єктів предметної області).
 8. Сценарії (позитивний і негативний) для тестування моделі.
 9. Текст програми для продукційної моделі.
 10. Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз. У висновках також необхідно порівняти семантичні мережі та продукційні моделі.
- Приклади виконання елементів звіту наведено у додатку А.

Контрольні запитання

- 1 Декларативні та процедуральні моделі. Області застосування.
- 2 Продукційні моделі: Основні визначення.
- 3 База правил.
- 4 Робоча область.
- 5 Інтерпретатор правил.
- 6 Управління виведенням у продукційній системі.
- 7 Характеристика продукційних моделей.
- 8 Архітектура експертної системи.
- 9 Інтерфейс користувача експертної системи.
- 10 Модуль придбання знань.
- 11 Машина логічного виведення.
- 12 Підсистема роз'яснень.
- 13 Етапи проектування експертної системи.
- 14 Етап інтерпретації.
- 15 Етап концептуалізації.
- 16 Етап формалізації.
- 17 Етап тестування.
- 18 Дослідна експлуатація.
- 19 Концепція «швидкого прототипу».
- 20 Впровадження в промислову експлуатацію.
- 21 Особливості модифікації і супроводу в експлуатації експертної системи.
- 22 Склад розроблювачів експертної системи, роль і задачі кожного з членів групи.
- 23 Які властивості предметної області (об'єкта автоматизації) є передумовою для створення експертної системи.
- 24 Назвіть області людської діяльності в який застосовуються експертні системи.
- 25 Стратегії керування виведенням.
- 26 Пряме та зворотне виведення.

- 27 Представлення задач у просторі станів.
- 28 Представлення за допомогою зведення задач до підзадач.
- 29 Представлення у вигляді теорем комбінування представлень.
- 30 Пошук рішень у просторі станів.
- 31 Метод перебору всередину. Метод перебору в ширину.
- 32 Пошук з поверненням.
- 33 Евристичний пошук.
- 34 Використання метазнань для обмеження області пошуку рішень.
- 35 Пошук методом редукції.
- 36 Робоча пам'ять у продукційній системі.
- 37 Цикл роботи інтерпретація правил.
- 38 Механізм виведення в продукційній системі.
- 39 Характеристика систем, заснованих на знаннях.
- 40 Чим експертні системи відрізняються від звичайних програмних додатків та типових програм штучного інтелекту?
- 41 Чи може програма, яка не використовує методи штучного інтелекту, мати такі ж властивості як експертна система?
- 42 У чому різниця між експертною системою та системою, що ґрунтується на знаннях?
- 43 Чи буде експертною системою програма передбачення погоди у Криму, що виводить повідомлення такого роду: "Завтра погода не буде відрізнитися від сьогоднішньої"?
- 44 Нехай вона представляє сьогоднішню погоду у символічному вигляді, легко модифікується та здатна до розширення, прекрасно працює та може пояснити, чому вона прийшла до певного висновку.
- 45 Чи є експертною системою програма, котра формує прогноз погоди на певну дату шляхом усереднення температури повітря, кількості опадів та кількості сонячних годин у цю дату за всі роки, починаючи з 1900?
- 46 Чи є система пошуку у мережі World Wide Web експертною? Якщо ні, то яких властивостей їй бракує для того, щоб кваліфікувати її як експертну систему пошуку потрібної Web-сторінки?
- 47 Чому задача набуття знань є вузьким місцем у проектуванні експертних систем? Які рішення пропонуються для виходу з такої ситуації?
- 48 Чому пакет програм статистичного аналізу не можна вважати програмою штучного інтелекту?
- 49 Як ви розумієте термін "простір пошуку"?
- 50 Як ви розумієте термін "простір рішень"?

ЛІТЕРАТУРА

Основна література

1. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навчальний посібник. – Запоріжжя: ЗНТУ, 2008. – 341 с.
2. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – С. – Пб.: Питер, 2001. – 384 с.

Додаткова література

3. Базы данных. Интеллектуальная обработка информации / В.В. Корнеев , А.Ф. Гареев, С.В. Васютин , В.В. Райх. – М.: Нолидж, 2000. – 352 с.
4. Бакаев А.А. и др. Методы организации и обработки баз знаний. – К.: Наукова думка, 1993. – 147с.
5. Джексон П. Введение в экспертные системы. Уч. пособие. – М.: Издательский дом "Вильямс", 2001. – 624 с.
6. Левин Р. Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрациями на Бейсике. – М.: Финансы и статистика, 1991. – 239 с.
7. Представление и использование знаний / Под ред. Х. Уэно, М. Исидзука. – М.: Мир, 1989. – 220 с.
8. Рідкокаша А.А., Голдер К.К. Основи систем штучного інтелекту. Навчальний посібник. Черкаси, "ВІДЛУННЯ – ПЛЮС", 2002. – 240 с.
9. Робототехника и гибкие автоматизированные производства. В 9 – ти кн. Кн. 6. Техническая имитация интеллекта / В.М. Назаретов, Д.П. Ким. – М.: Высш. шк., 1986. – 144 с.
10. Сойер Б., Форстер Д. Программирование экспертных систем на Паскале. – М.: Финансы и статистика, 1990. – 191 с.
11. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ. – М.: Финансы и статистика, 1990. – 320 с.
12. Уотерман Д. Руководство по экспертным системам. – М.:Мир, 1989. – 388 с.
13. Экспертные системы. Принципы работы и примеры. Под редакцией Р. Форсайта. – М.: Радио и связь, 1987. – 224с.
14. Элти Дж., Кубмс М. Экспертные системы: концепции и примеры. – М.: Финансы и статистика, 1987. – 191 с.

Додаток А
Приклади виконання елементів звітів
з лабораторних робіт

Міністерство освіти і науки України
Запорізький національний технічний університет
Інститут інформатики та радіoeлектроніки

кафедра програмних засобів

ЗВІТ
з лабораторної роботи № 1
з дисципліни "Дискретні структури"
на тему:
"РОЗРОБКА СЕМАНТИЧНОЇ МЕРЕЖІ"

Виконав: ст. гр. ІОТ-413 _____ А.Б. Іванов

Прийняв: к.т.н., доцент _____ С.О. Субботін

Запоріжжя
2013

Приклад 1. Розглянемо виконання певних елементів звітів на прикладі побудови експертної системи для діагностики внутрішніх хвороб. Усі наведені приклади є вигаданими.

Назва предметної області: Діагностика внутрішніх хвороб.

Мета створення експертної системи

Внутрішні хвороби є такими, що складно визначаються як для неспеціаліста, так і для професіонала. Їхня діагностика вимагає врахування великої кількості показників, що характеризують стан пацієнта. Кількість відомих хвороб внутрішніх органів та їх різновидів також є значною. Тому дуже актуально є розробка експертної системи для діагностики внутрішніх хвороб.

Мета експертної системи - визначення діагнозу пацієнта за значеннями симптомів, що характеризують його стан.

Конкретизація вхідних та вихідних даних

Вхідними параметрами є симптоми пацієнта: температура тіла, місце болю, вид болю.

Вихідним параметром є діагноз пацієнта.

Таблиця А.1 - Словник предметної області

Назва об'єкта (події)	Визначення сутності	Назва атрибута (властивості) об'єкта	Допустимі значення атрибута
Пацієнт	людина, що проходить медичне обстеження	Симптоми	
		Діагноз	
Симптоми	показники, що відображують стан пацієнта	Температура тіла	Нормальна
			Підвищена
		Місце болю	У животі
			У попереку
		Вид болю	Сильний
			Слабкий
Температура тіла	характеристика температури тіла пацієнта у °C	Нормальна	
		Підвищена	

Нормальна	температура тіла, що менше 37°C		
Підвищена	температура тіла, що більше або дорівнює 37°C		
Місце болю	область тіла пацієнта, де відчувається біль	У животі	
		У попереку	
Вид болю	Умовний рівень відчуття болю	Сильний	
		Слабкий	
Діагноз	вид стану пацієнта	Гострий апендицит	
		Хронічний апендицит	
		Гострий пієлонефрит	
		Хронічний пієлонефрит	
Гострий апендицит	хвороба кишок, що проявляється у запаленні апендиксу		
Хронічний апендицит	застаріле запалення апендиксу		
Гострий пієлонефрит	хвороба нирок у стадії загострення		
Хронічний пієлонефрит	застаріла хвороба нирок		

Узагальнення

Узагальненнями для різних значень, що поступають на входи, є такі класи: апендицит, пієлонефрит.

Сценарії

Позитивний сценарій:

Якщо температура тіла - нормальна
та місце болю - у животі
та вид болю - слабкий,
то діагноз - хронічний апендицит.

Негативний сценарій:

Якщо температура тіла - нормальна
та місце болю - у животі
та вид болю - сильний,
то діагноз - невідомий.

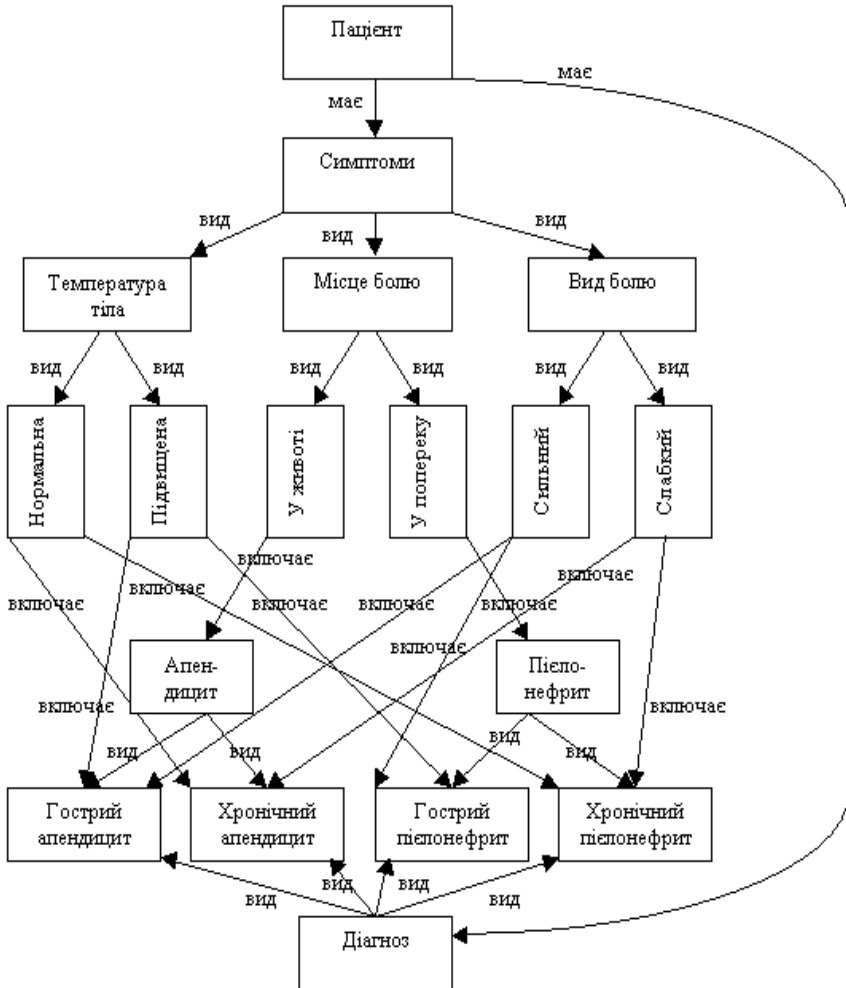


Рисунок А.2 - Схема семантичної мережі, побудована вручну

Текст програми, що реалізує семантичну мережу із використанням функцій бібліотеки SNToolbox пакету MATLAB (Зауваження: в тексті програми коментарі наведено українською мовою для спрощення сприйняття. Використання кириличних літер у пакеті MATLAB може призводити до збоїв, тому коментарі бажано писати латинськими літерами).

```
% створюємо структуру для семантичної мережі
SN=SNnew;
```

```
% додаємо вузли типу "АБО" - значення атрибутів
SN=SNaddORnode(SN, 'high','normal','acute', 'weak','in stomach', 'in spine');
```

```
% додаємо вузли типу "ТА" - об'єкти і назви атрибутів
SN=SNaddANDnode(SN, 'patient', 'symptoms', 'temperature', 'pain type', 'pain',
'diagnosis', 'apendicitis', 'pielonefrit', 'acute apendicitis', 'acute pielonefrit', 'chronic
apendicitis', 'chronic pielonefrit');
```

```
% додаємо відносини між вузлами
SN=SNaddrelation(SN, 'patient', 'has', 'symptoms');
SN=SNaddrelation(SN, 'patient', 'has', 'diagnosis');
SN=SNaddrelation(SN, 'symptoms', 'type', 'temperature');
SN=SNaddrelation(SN, 'symptoms', 'type', 'pain type');
SN=SNaddrelation(SN, 'symptoms', 'type', 'pain');
SN=SNaddrelation(SN, 'diagnosis', 'type', 'acute apendicitis');
SN=SNaddrelation(SN, 'diagnosis', 'type', 'acute pielonefrit');
SN=SNaddrelation(SN, 'diagnosis', 'type', 'chronic apendicitis');
SN=SNaddrelation(SN, 'diagnosis', 'type', 'chronic pielonefrit');
SN=SNaddrelation(SN, 'temperature', 'type', 'high');
SN=SNaddrelation(SN, 'temperature', 'type', 'normal');
SN=SNaddrelation(SN, 'pain type', 'type', 'acute');
SN=SNaddrelation(SN, 'pain type', 'type', 'weak');
SN=SNaddrelation(SN, 'pain', 'type', 'in stomach');
SN=SNaddrelation(SN, 'pain', 'type', 'in spine');
SN=SNaddrelation(SN, 'in stomach','include', 'apendicitis');
SN=SNaddrelation(SN, 'in spine', 'include', 'pielonefrit');
SN=SNaddrelation(SN, 'apendicitis', 'include', 'acute apendicitis');
SN=SNaddrelation(SN, 'apendicitis', 'include', 'chronic apendicitis');
SN=SNaddrelation(SN, 'pielonefrit', 'include', 'acute pielonefrit');
SN=SNaddrelation(SN, 'pielonefrit', 'include', 'chronic pielonefrit');
SN=SNaddrelation(SN, 'high', 'include', 'acute apendicitis');
SN=SNaddrelation(SN, 'normal', 'include', 'chronic apendicitis');
```

```

SN=SNaddrelation(SN, 'high', 'include', 'acute pielonefrit');
SN=SNaddrelation(SN, 'normal', 'include', 'chronic pielonefrit');
SN=SNaddrelation(SN, 'acute', 'include', 'acute apendicitis');
SN=SNaddrelation(SN, 'weak', 'include', 'chronic apendicitis');
SN=SNaddrelation(SN, 'acute', 'include', 'acute pielonefrit');
SN=SNaddrelation(SN, 'weak', 'include', 'chronic pielonefrit');

```

```

% будуємо графічні зображення схеми семантичної мережі
SNplot(SN, 'hierarchy'); % ієрархічне розташування вузлів
figure; % створюємо нове вікно для іншої схеми
SNplot(SN, 'circle'); % кругове розташування вузлів

```

```

% створюємо мережу-запит на основі мережі бази знань
SN1=SN;
% видаляємо з мережі-запиту зайві вузли
SN1=SNdelnode(SN1, 'normal', 'in spine', 'weak', 'apendicitis', 'pielonefrit', 'acute
apendicitis', 'acute pielonefrit', 'chronic apendicitis', 'chronic pielonefrit');
% додаємо до мережі-запиту цільовий вузол
SN1=SNaddANDnode(SN1, '?');
% додаємо до мережі-запиту відношення для цільового вузла
SN1=SNaddrelation(SN1, 'diagnosis', 'type', '?');
figure; % створюємо нове вікно для схеми
% будуємо графічне зображення схеми мережі-запиту
SNplot(SN1, 'random'); % випадкове розташування вузлів

% виконуємо запит до семантичної мережі,
% результати якого видаємо на екран
Res=SNfind(SN, SN1)

```

Результати роботи програми

Повідомлення у командному вікні пакету MATLAB.

```

RESULT: The number of founded solutions is 1.

```

```

Res =

```

```

'acute apendicitis'

```

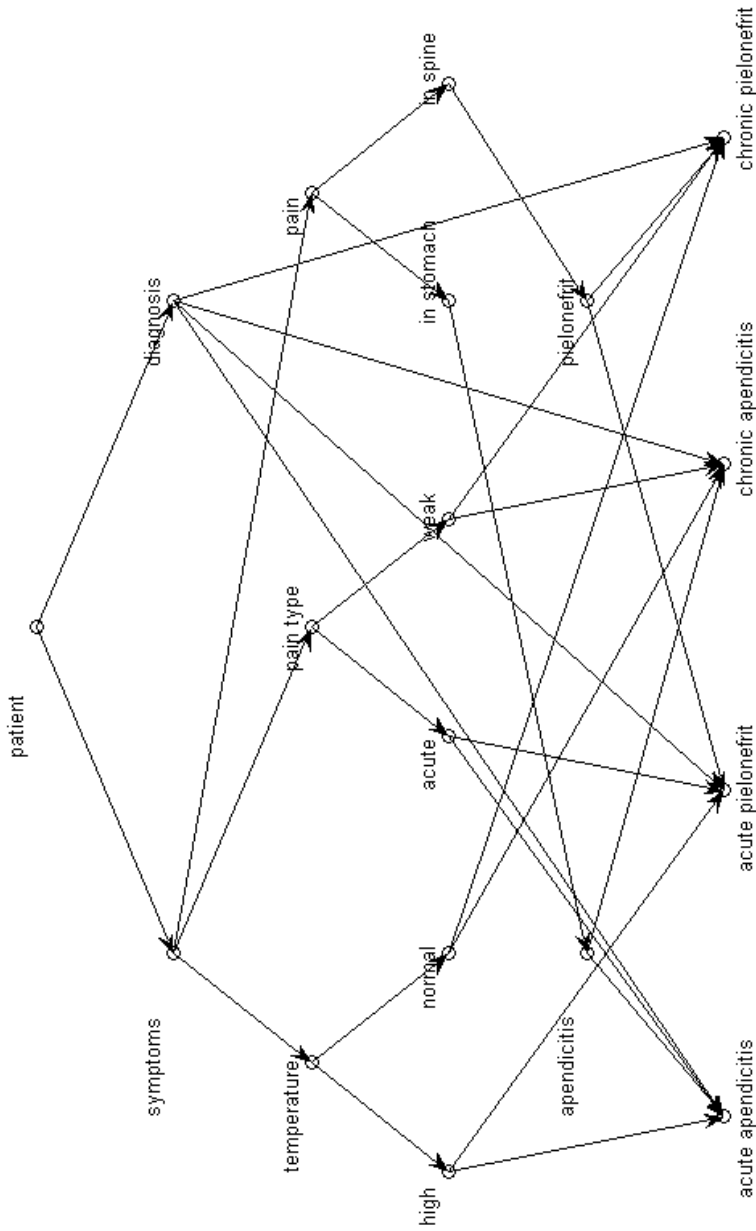


Рисунок А.3 -. Програмно побудована схема семантичної мережі з ієрархічним розташуванням вузлів

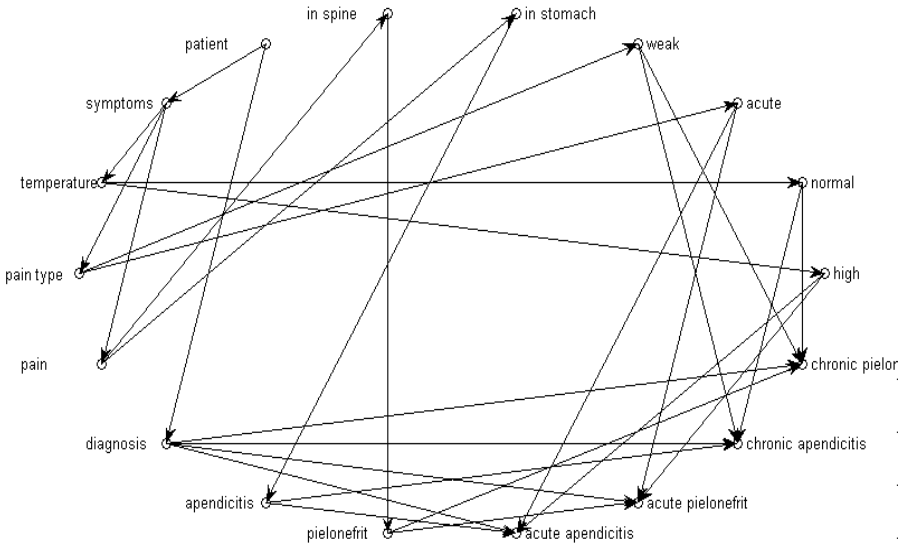


Рисунок А.4 - Програмно побудована схема семантичної мережі з круговим розташуванням вузлів.

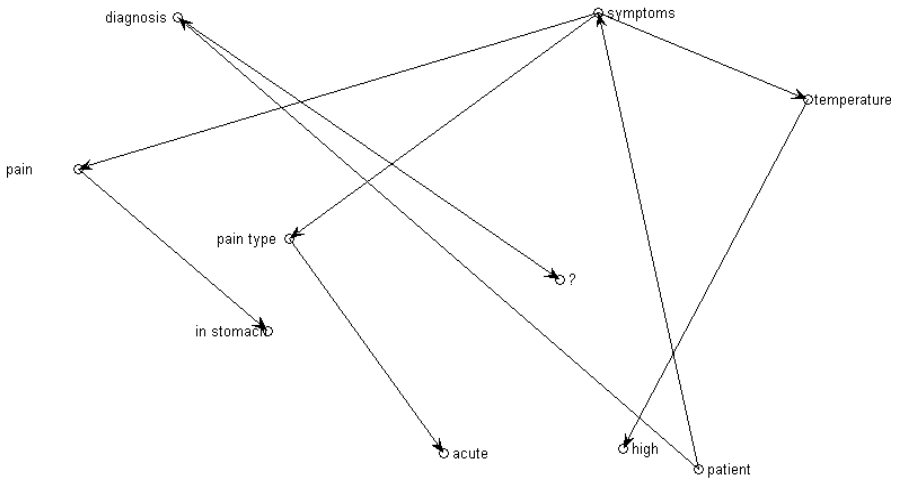


Рисунок А.5 - Програмно побудована схема семантичної мережі-запиту з випадковим розташуванням вузлів.

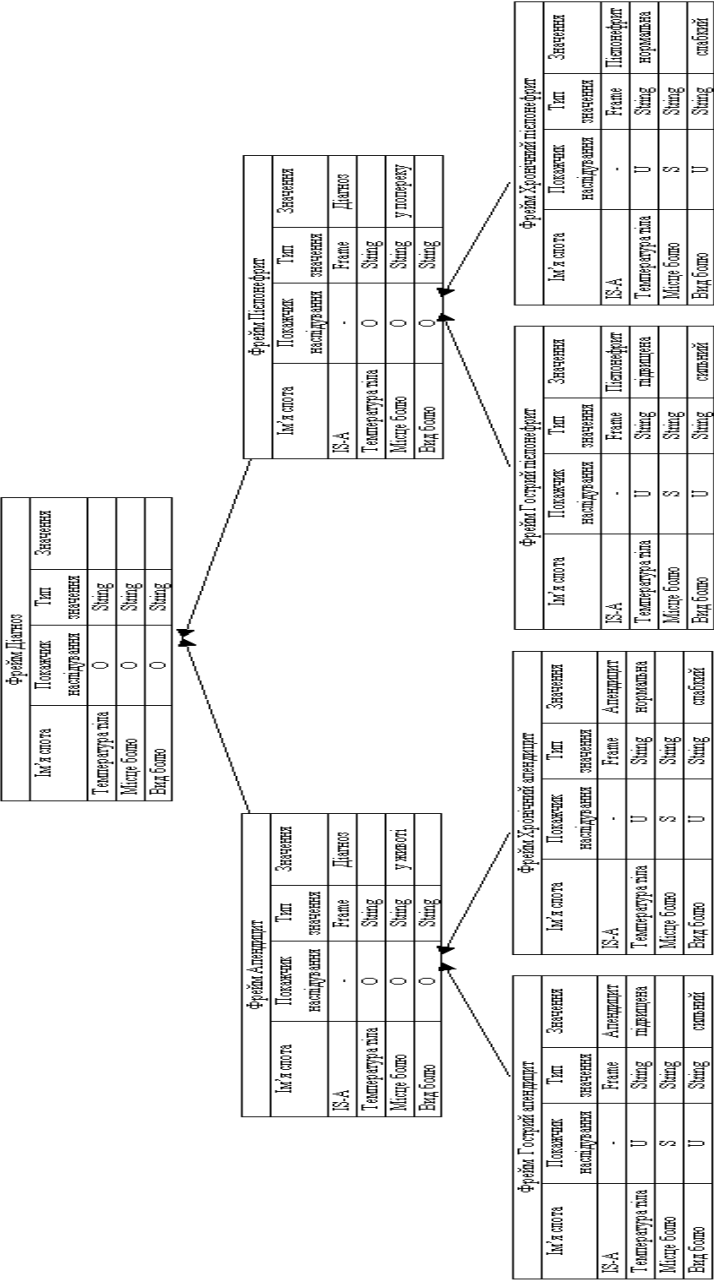


Рисунок А.6 - Схема ієрархічної фреймової моделі

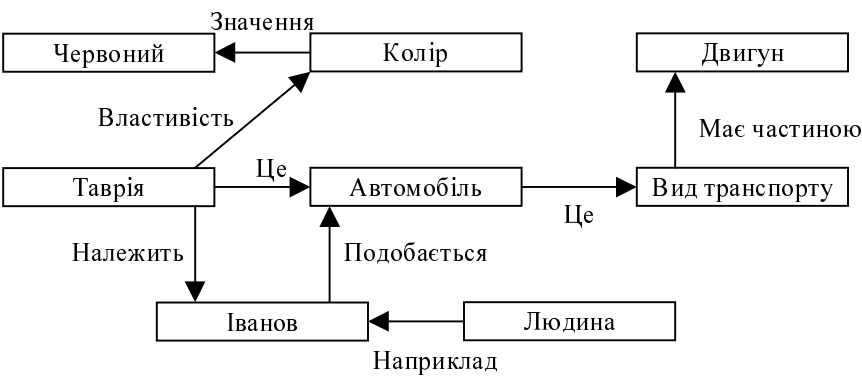
Продукційна модель

Лістинг бази знань мовою експертної системи "Expertise2Go".

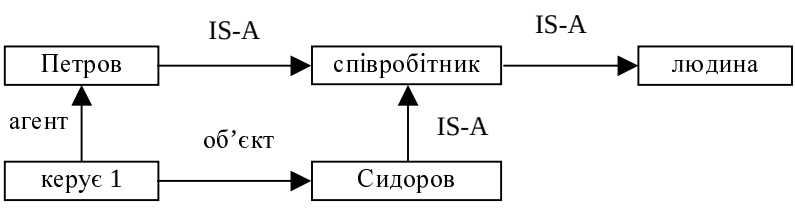
англійською мовою	українською мовою
REM Medical expert system	REM Медична експертна система
RULE [appendicitis] If [pain] = "in stomach" Then [disease] = "appendicitis"	RULE [апендицит] If [біль] = "у животі" Then [хвороба] = "апендицит"
RULE [pielonephritis] If [pain] = "in spine" Then [disease] = "pielonephritis"	RULE [пієлонефрит] If [біль] = "у попереку" Then [хвороба] = "пієлонефрит"
RULE [chronic appendicitis] If [temperature] = "normal" and [disease] = "appendicitis" and [pain type] = "weak" Then [diagnosis] = "chronic appendicitis"	RULE [хронічний апендицит] If [температура] = "нормальна" and [хвороба] = "апендицит" and [вид болю] = "слабкий" Then [діагноз] = "хронічний апендицит"
RULE [acute appendicitis] If [temperature] = "high" and [disease] = "appendicitis" and [pain type] = "acute" Then [diagnosis] = "acute appendicitis"	RULE [гострий апендицит] If [температура] = "підвищена" and [хвороба] = "апендицит" and [вид болю] = "сильний" Then [діагноз] = "гострий апендицит"
RULE [chronic pielonephritis] If [temperature] = "normal" and [disease] = "pielonephritis" and [pain type] = "weak" Then [diagnosis] = "chronic pielonephritis"	RULE [хронічний пієлонефрит] If [температура] = "нормальна" and [хвороба] = "пієлонефрит" and [вид болю] = "слабкий" Then [діагноз] = "хронічний пієлонефрит"
RULE [acute pielonephritis] If [temperature] = "high" and [disease] = "pielonephritis" and [pain type] = "acute" Then [diagnosis] = "acute pielonephritis"	RULE [гострий пієлонефрит] If [температура] = "підвищена" and [хвороба] = "пієлонефрит" and [вид болю] = "сильний" Then [діагноз] = "гострий пієлонефрит"
REM User dialogs	REM Діалоги користувача
PROMPT [pain] MultChoice "Where is the pain sensed?" "in stomach" "in spine"	PROMPT [біль] MultChoice "Де відчувається біль?" "у животі" "у попереку"

PROMPT [temperature] MultChoice "What is the body temperature?" "normal" "high" PROMPT [pain type] MultChoice "What is the type of pain?" "acute" "weak" REM The goal of decision search in knowledgebase GOAL [diagnosis]	PROMPT [температура] MultChoice "Яка температура тіла?" "нормальна" "підвищена" PROMPT [вид болю] MultChoice "Який біль?" "сильний" "слабкий" REM Мета пошуку рішення у базі знань GOAL [діагноз]
--	---

Приклад 2. Зобразимо у вигляді семантичної мережі відносини між поняттями «людина», «Іванов», «Таврія», «автомобіль», «вид транспорту» та «двигун».

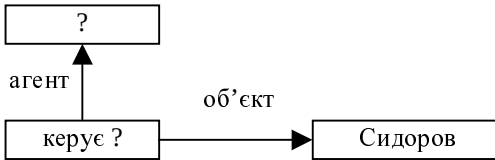


Приклад 3. Підпорядкованість співробітників організації може бути представлена такою семантичною мережею.



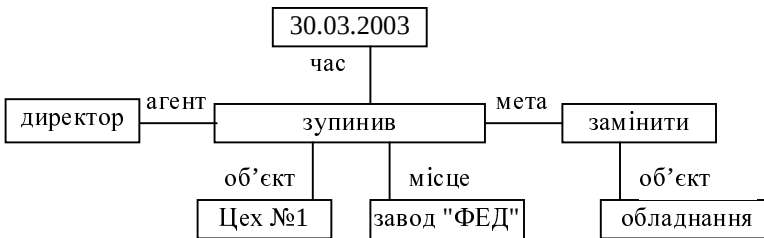
Приведені зв'язки показують підпорядкованість першого співробітника. Інші співробітники зв'язуються через вершини мережі. Інші співробітники зв'язуються через вершини мережі “керує 2”, “керує 3” і т.д.

Запит: “Хто керує Сидоровим?” представимо у вигляді такої підмережі.



Зіставлення загальної мережі з мережею запиту починається з пошуку вершини “керує”, що має гілку “об’єкт”, спрямовану до вершини “Сидоров”. Потім здійснюється перехід по гілці “агент”, що і приводить до відповіді “Петров”.

Приклад 4. Семантична структура знання про подію “Директор заводу “ФЕД” зупинив 30.03.2003 цех № 1 щоб замінити обладнання” може бути представлена таким чином.



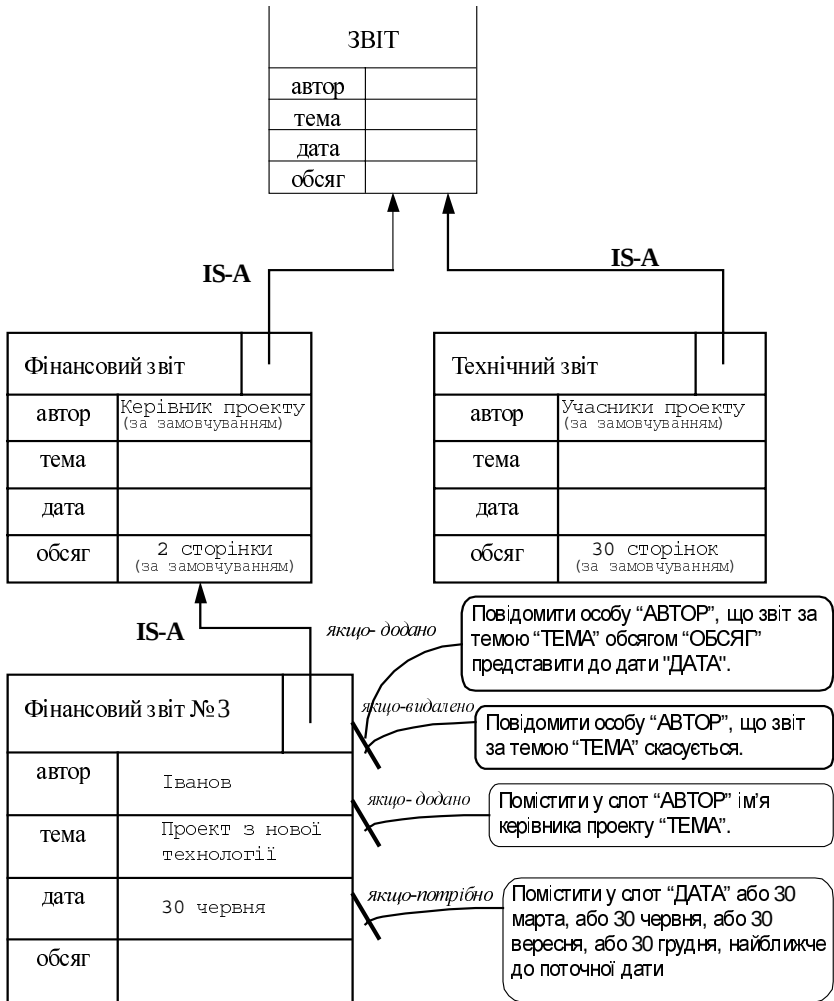
Приклад 5. Поняття про діловий звіт у системі, заснованій на фреймах, може мати такий вигляд, як зображено на наведеному нижче рисунку.

Припустимо, що деякі слоти мають значення за замовчуванням. Тоді дані, що мають таку організацію, можна використовувати відповідним чином.

Якщо системі заданий запит: “Мені потрібний фінансовий звіт про виконання проекту з нової технології”, то інтерфейсна програма аналізує його і виконує такі дії.

1. Вносить “проект з нової технології” у слот “ТЕМА” наступного порожнього вузла “Фінансовий звіт”. Далі усе відбувається автоматично:

2. Процедура “якщо – додано”, зв'язана зі слотом “ТЕМА”, здійснює пошук керівника цього проекту. Допустимо, що його прізвище Іванов.



Процедура вписує його прізвище в слот "АВТОР" фінансового звіту №3. Якщо керівник цієї теми не буде знайдений, у слот "АВТОР" буде успадковане значення класу, а саме текст "КЕРІВНИК ПРОЕКТУ".

3. Процедура "якщо – додано", зв'язана зі слотом "АВТОР", починає виконуватися, тому що в слот тільки що було уписано нове значення. Ця процедура починає складати повідомлення, щоб відправити його Іванову, але виявляє, що немає потрібної дати виконання.

4. Процедура “якщо – додано”, переглядаючи слот “ДАТА” і знайшовши його порожнім, активує процедуру “якщо – потрібно”, зв’язану з цим слотом, що, аналізуючи поточну дату (наприклад 12.04.96), вирішить, що “30 червня” найближче до неї і впише цю дату в слот “ДАТА”.

5. Тепер процедура “якщо – додано”, зв’язана зі слотом “АВТОР”, знайде, що ще одне значення, яке потрібно включити в повідомлення, тобто обсяг звіту, є відсутнім. Слот “ОБСЯГ” не зв’язаний із процедурами і нічим допомогти не може. Однак вище даного фрейму існує фрейм загальної концепції фінансового звіту, що містить значення обсягу. Процедура, використовуючи концепцію спадкування властивостей класу, використовує значення обсягу і складає наступне повідомлення: “Пан Іванов, підготуйте фінансовий звіт з проекту нової технології до 30 червня обсягом 2 сторінки”.

Якщо в якийсь момент прізвище Іванов буде вилучено зі слота “АВТОР”, то система автоматично відправить йому повідомлення, що його звіт є непотрібним.

Приклад 5. У наведеній нижче мережі фреймів поняття «учень» успадковує властивості фреймів «дитина» і «людина», що знаходяться на більш високому рівні ієрархії.



Так, на питання «чи люблять учні солодке?» впливає відповідь «так», тому що цю властивість мають усі діти, що зазначено у фреймі «дитина». Спадкування властивостей може бути частковим, тому що вік для учнів не успадковується з фрейму «дитина», оскільки він зазначений явно у своєму власному фреймі.

Додаток Б

Перелік предметних областей для побудови моделей знань

При виборі теми студент повинен оцінити свої реальні можливості щодо отримання необхідної інформації для опису предметної області і створення бази знань. Експертна система – це відтворення практичного досвіду фахівців – експертів. Джерелами інформації можуть бути: знайомі фахівці – практики (митці, лікарі, кулінари, автослюсарі, і т.д.), книжки за темою аматорських захоплень, довідники, методики, толкові словники, енциклопедії. Бажано щоб тема була цікавою або добре відомою особисто для студента.

Пропонуються узагальнені теми, які необхідно доповнити конкретними поняттями згідно вибору студента і узгодити з викладачем.

Клас діагностичних систем:

1. Діагностика захворювань (органи дихання, кровообіг, травлення, зір та ін.).
2. Діагностика несправностей пристроїв (телевізор, комп'ютер, автомобіль та ін.).
3. Розпізнання видів (комах, тварин, рослин та ін.).
4. Вибір стратегії виходу фірми з кризової ситуації.
5. Експертна система з індивідуального підбору (косметики, одягу, взуття, мобільного телефону, відеомагнітофону, музичного центру, відеокамери, телевізора, комп'ютера, периферійного пристрою ЕОМ, автомобіля, житла, туристичної путівки, спеціальності для абітурієнтів та ін.).

Клас конструювання:

1. Розміщення меблів (станків, приміщень в споруді, що будується).
2. Планування (ремонтів, бізнес-процесів).
3. Комплектація комп'ютерних систем за даними критеріями: функції, швидкодія, ціна, ефективність.
4. Складення планів завантаження обладнання та людей для цехів механообробки та конвеєрно-поточних ліній.
5. Рекомендації по створюванню матеріалів за даними властивостями.
6. Планування промислових замовлень.
7. Планування експериментів, поведінки роботів.

Додаток В

Побудова інтернет-орієнтованих експертних систем за допомогою компоненту E2GLITE

E2gLite - це Java-аплет, що реалізує оболонку експертної системи, яка включається у веб-сторінку та завантажується із веб-сервера браузером користувача. Аплет завантажує базу знань із сервера, після чого запускається у браузері. Програмне забезпечення e2gLite постачається в архіві e2glite.jar.

Аплет e2gLite підключають до html-сторінки, використовуючи конструкцію:

```
<APPLET CODE="e2glite.e2g.class"
ARCHIVE="e2glite.jar" WIDTH=450 HEIGHT=300 MAYSCRIPT>
<PARAM NAME="KBURL" VALUE="knowledgebasefile.kb">
(опційні <PARAM> теги)
.....
</APPLET>
```

У цій конструкції APPLET CODE містить ім'я класу, а ARCHIVE - назву файлу з архівом аплету. Назви класу та файлу є чутливими до регістру та мають бути наведені у нижньому регістрі. Якщо файл e2glite.jar розміщено у іншій директорії ніж html-документ, то параметр ARCHIVE повинен містити шлях до файлу. Параметр MAYSCRIPT включається тільки, коли e2gLite буде викликати функцію Java-script (Див. JSFUNCTION нижче).

Параметри аплету e2gLite

KBURL - ім'я бази знань із шляхом до директорії, де знаходиться файл бази знань (або URL бази знань). Приклади використання: <PARAM NAME="KBURL" VALUE="auto.kb"> - база знань у тій же директорії, що й html-документ; <PARAM NAME="KBURL" VALUE="http://www.mysite.com/kb/auto.kb"> - URL адреса файлу бази знань

APPTITLE - будь-який рядок символів - перший рядок заголовку на початковому екрані. Значення за замовчанням: Blank. Приклад використання: <PARAM NAME="APPTITLE" VALUE="Auto Diagnosis">

APPSUBTITLE - будь-який рядок символів - другий рядок заголовку на початковому екрані. Значення за замовчанням: Blank. Приклад використання: <PARAM NAME="APPSUBTITLE" VALUE="by Ace Auto Repair">

BGCOLOR - Колір фону аплету. Задається як шістнадцятиричний колір RGB, що використовується у html-документах. Значення за замовчанням: #FFEE33 (блідо-жовтий). Приклад використання: <PARAM NAME="BGCOLOR" VALUE="#00FF00"> - зелений.

TITLECOLOR - колір тексту APPTITLE та APPSUBTITLE, визначений як шістнадцятирічний RGB колір. Значення за замовчанням: #000000 (чорний). Приклад використання: <PARAM NAME="TITLECOLOR" VALUE="#FFFFFF"> - білий.

PROMPTCOLOR - колір тексту діалогів, визначений як шістнадцятирічний RGB колір. Значення за замовчанням: #000000 (black). Приклад використання: <PARAM NAME = "PROMPTCOLOR" VALUE = "#FFFFFF"> - білий

DEBUG - якщо має значення TRUE, то вікно відлагоджувача буде створюватися коли база знань стартує. Значення за замовчанням: FALSE. Приклад використання: <PARAM NAME="DEBUG" VALUE="TRUE">

STARTBUTTON - текст на кнопці, що починає консультацію. Значення за замовчанням: "Start the consultation". Приклад використання: <PARAM NAME="STARTBUTTON" VALUE="Help me!">

LOADMSG - повідомлення, яке відображується на екрані, що відкривається, коли завантажується база знань. Значення за замовчанням: "Loading knowledge base...". Приклад використання: <PARAM NAME = "LOADMSG" VALUE="Chargeant la base de connaissance...">

PROMPTFIRST - якщо має значення TRUE, тоді діалог використовується для атрибута перед зворотнім виведенням. Значення за замовчанням: TRUE. Приклад використання: <PARAM NAME = "PROMPTFIRST" VALUE="false">

JSFUNCTION - ім'я функції Java-скрипта, що викликається кожний раз, коли кнопка e2gLite натискається. Значення за замовчанням: null (no function is called). Приклад використання: <PARAM NAME="JSFUNCTION" VALUE="buttonPush">

Зауваження:

1. Ім'я параметрів аплету є чутливими до регістру та завжди мають бути визначені у верхньому регістрі в e2gLite.
2. Значення параметрів аплету e2gLite є нечутливими до регістру.
3. Єдиний параметр аплету, що обов'язково має бути визначений - KBURL.
4. Якщо значення KBURL не починається з http://, то значення буде додано до директорії, у якій знаходиться html-документ, що завантажує аплет e2gLite.
5. База знань має бути завантажена з того ж веб-сайту, що й архів аплету e2gLite.jar. Це вимога безпеки Java-аплетів.
6. Коли використовується параметр JSFUNCTION, MAYSCRIPT повинен включатися у тег <applet>.

Коди кнопок e2gLite

Натиснутій кнопці "Start the consultation" відповідає значення - 1 (ця кнопка може бути перейменована через параметр STARTBUTTON; інші кнопки можуть бути перейменовані командами бази знань TRANSLATE), кнопки "Submit your response" - 2, кнопки "Why Ask?" - 3, кнопки "Return (from Why Ask?)" - 4, кнопки "Explain" - 5, кнопки "Restart" - 6.

Методи e2gLite, що доступні з Java-скрипту

Ім'я методу	Значення, що повертається, та його тип	Аргументи та їхні типи
getAttrCF(attrName, valIx)	(Float) Коефіцієнт упевненості, асоційований з певним прикладом поименованого атрибуту (0.0, якщо не існує або ще не визначено)	(String) attrName: ім'я атрибуту (Integer) valIx: приклад (між 1 та максимальним значенням) поименованого атрибуту для дозволяння багатозначних атрибутів
getAttrCount()	(Integer) Кількість атрибутів, що визначені у базі знань	
getAttrIx(attrName, valIx)	(Integer) Індекс атрибуту, асоційованого з певним прикладом поименованого атрибуту (0, якщо не існує)	(String) attrName: ім'я атрибуту (Integer) valIx: приклад (між 1 та максимальним значенням) поименованого атрибуту для дозволяння багатозначних атрибутів
getAttrName(attrIx)	(String) Ім'я атрибуту (якщо немає - рядок нульової довжини)	(Integer) attrIx: індекс атрибуту
getAttrValue(attrIx)	(String) Строкове представлення поточного значення (рядок нульової довжини, якщо не існує або ще не визначено)	(Integer) attrIx: індекс атрибуту
getAttrValue(attrName, valIx)	(String) Строкове представлення поточного значення (рядок нульової довжини, якщо не існує або ще не визначено)	(String) attrName: ім'я атрибуту (Integer) valIx: приклад (між 1 та максимальним значенням) поименованого

		атрибуту для дозволяння багатозначних атрибутів
getMaxValues(attrName)	(Integer) Кількість значень (прикладів) дозволених для поійменованого атрибуту (0, якщо не існує)	(String) attrName: ім'я атрибуту
getGoalAttr()	(Integer) індекс атрибуту поточної цілі або підцілі (0, якщо виведення завершено)	

Атрибути e2gLite позначаються ім'ям та номером індексу, що починається з 1. Кожне значення атрибуту має унікальний індекс, але усі значення багатозначних атрибутів зберігаються під одним і тим же ім'ям. Значення атрибуту не буде повернено доти, поки значення не стане відомим з коефіцієнтом упевненості що найменше рівним мінімальному CF.

Формат бази знань e2gLite

Команда	Опис	Приклади
REM	Рядок коментаря	REM Diagnose
RULE	Починає правило, яке слідує за описом правила у квадратних дужках	RULE [Is the battery dead?]
If	починає передумову правила: за ним слідує перша передумова правила (ім'я атрибута в квадратних дужках, реляційний оператор і порівнюване значення), необов'язковий логічний оператор <i>and/or</i> (якщо існує інша передумова правила).	If [the result of switching on the headlights] = "nothing happens" or [the result of trying the starter] = "nothing happens"
Then	Починає наслідок правила, за ним слідує перше речення наслідку (назва атрибута в квадратних дужках, знак "=", значення коефіцієнта упевненості (необов'язково): @xxx, де xxx представляє відсоток і за замовчуванням дорівнює 100), необов'язковий AND (потрібно, якщо мається інше речення наслідку)	Then [the problem] = "the battery is discharged or dead" @ 90 and [the recommended action] = "recharge or replace the battery"

PROMPT	Визначає формат запиту користувача про значення атрибута, коли воно необхідне машині виведення. Типи діалогів: YesNo (булеве введення), MultChoice (багатозначний вибір текстових рядків), Choice (вибір текстових рядків у списку, що випадає в низ), AllChoice (багатозначний вибір текстових рядків), Numeric (введення із перевіркою діапазону). Необов'язковий параметр CF указує на необхідність запиту коефіцієнта упевненості). Питання, що задається користувачу, знаходиться на наступному рядку в одинарних чи подвійних лапках. Варіанти відповідей (якщо потрібно діалогу) знаходяться на послідовних рядках у лапках. Для числового діалогу мінімальні і максимальні припустимі значення знаходяться на послідовних рядках у лапках.	PROMPT [the stoplights work] YesNo CF "Do the stoplights light when you step on the brake?" PROMPT [the gas tank] MultChoice CF "According to the fuel gauge, the gas tank is:" "empty" "not empty" PROMPT [the result of trying the starter] Choice CF "What happens when you turn the key to try to start the car?" "the car cranks normally" "the car cranks slowly" "nothing happens" PROMPT [an acceptable time to pick up your car] Choice What time may we pick up your car? (Check up to three) "7 - 9 AM" "9 AM -12 PM" "12 - 3 PM" "3 - 7 PM" PROMPT [the amount you are willing to spend on repairs] Numeric CF "How much are you willing to spend on repairs? (enter value 0->500)" "0" "500.0"
GOAL	Знаходження значення (чи значень) для цільових перемінних - мета консультації експертної системи. Коли всі мети вирішені (або визначено, що вони не можуть бути вирішені) консультація завершується. Потрібна принаймні одна мета.	GOAL [the problem]

MINCF	Мінімальне значення коефіцієнта впевненості для прийняття значення як факту. За замовчуванням: 80.0	MINCF 70
MAX-VALS	(Необов'язково) максимальна кількість різних значень, які може мати одна ознака. За замовчуванням - 1. Атрибути з діалогами типу AllChoice повинні мати MAXVALS більше ніж 1.	MAXVALS [an acceptable time to pick up your car] 3
DEFAULT	(Необов'язково) Установлює значення атрибута за замовчуванням, якщо машина виведення не може одержати її значення з діалогів або від правил.	DEFAULT [the gas tank] = "not empty" DEFAULT [the stoplights work] = true DEFAULT [the amount you are willing to spend on repairs] = 50
TRANSLATE	(Необов'язково) Налаштовує виведення тексту. Також підтримує переклад бази знань на мови, відмінні від англійської мови.	TRANSLATE B_SUBMIT = "Soumettez votre réponse" TRANSLATE TR_YES = "Oui"

Зауваження:

1. Всі правила RULE мають бути визначені перед використанням будь-якої команди, за виключенням REM.
2. Ім'я атрибутів мають бути взяті у квадратні дужки [].
3. Текстові рядки, що представляють значення атрибутів мають бути взяті у одинарні або подвійні лапки.
4. Команди, ім'я атрибутів та текстові значення є нечутливі до регістру.
5. Проміжки всередині квадратних дужок або лапок є значущими. Проміжки між іншими елементами є незначущими.
6. Речення з багатьма передумовами правил можуть бути пов'язані за допомогою *And* або *Or*, але не одночасно в одному і тому ж правилі.

Установки коментарів REM - це однострокові коментарі, що ігноруються, коли обробляється база знань. Команда REM повинна бути на початку рядка, а всі інші символи в тому ж рядку ігноруються.

Порожні рядки є необов'язковими й ігноруються. Вони можуть використовуватися для поліпшення зручності читання бази знань.

Установки правил RULE: Можливі елементи кожного правила включають:

- Перший рядок починається з RULE, що супроводжується коротким ідентифікуючим описом правила, взятим у квадратні дужки.

- Наступний рядок починає передумову правила з IF і супроводжується логічним виразом, що складається з: назви атрибута, взятої в квадратні дужки; реляційного оператора: = (дорівнює), < (менше ніж), > (більше ніж), ! (не дорівнює), : (дорівнює кожному з); порівнюване значення: число, булеві значення TRUE чи FALSE або рядок, взятий в одинарні чи подвійні лапки.

- (Необов'язково) один з логічних операторів AND або OR, якщо в передумові правила є кілька логічних виражень. Якщо є більш ніж один логічний вираз передумови, вони повинні усі зв'язуватися тим самим логічним оператором. Кожен пункт передумови, що складається з назви атрибута, реляційного оператора і порівнюваного значення, вводиться в окремому рядку. Заклучний пункт передумови не повинний закінчуватися AND чи OR.

- Наступний рядок після логічного виразу передумови, що не закінчується AND чи OR, починає наслідок правила з THEN, супроводжуваний присвоюванням, що складається з: назви атрибута, взятої у квадратні дужки; оператора присвоювання (=); значення, що буде присвоєне ознаці (число, булеві значення TRUE чи FALSE, або рядок, укладений в одинарні чи подвійні лапки).

- (Необов'язково) Якщо значення, зіставлене атрибуту з менш ніж 100% вірогідністю, воно супроводжується @ і числовим значенням коефіцієнта упевненості від 1 до 100%.

- (Необов'язково) Якщо є ще одна установка значення в наслідку правила, завершуйте рядок за допомогою AND. Рядок у наслідку правила, що не закінчується за допомогою AND, завершує правило.

- Елементи правила є не чутливими до регістра символів.

- Кожен елемент правила повинний бути на окремому рядку.

- Атрибути в e2gLite мають один із трьох типів: строковий (текстовий), числовий і булевий. Тип атрибута встановлюється в перший раз, коли вона з'являється в передумові правила або в наслідку правила при установці значення. Будь-яке наступне використання атрибута з іншим типом приводить до помилки. Використання атрибута, що попередньо не був визначений у правилі, у будь-якій іншій установці бази знань типу PROMPT чи GOAL також є помилкою.

- Коли правила представляються користувачу експертної системи як частина механізму пояснення, вони форматуються для зручності читання. Квадратні дужки і лапки опускаються, а реляційні оператори перетворюються в текст. Наприклад, = переводиться в is, і передумова правила: If

[precipitation] = "expected" переводиться в: **If precipitation is expected.** Ретельний вибір назв і значень ознак дозволяє створювати правила, зрозумілі для користувача.

Визначення діалогів: діалоги (PROMPT) повинні бути включені в базу знань після того, як визначені всі правила. Можливі елементи кожного діалогу включають:

- Перший рядок починається з PROMPT, супроводжуваного назвою атрибута, укладеною у квадратні дужки, і типом діалогу. Типи діалогу: MultChoice (багаторазовий вибір), Choice (список вибору, що випадає вниз), AllChoice (дозволяє вибирати кілька відповідей одночасно), YesNo (булеве введення) і Numeric (введення числового значення).

- (Необов'язково) закінчення першого рядка PROMPT з CF дозволяє користувачу експертної системи вводити коефіцієнт упевненості, коли відображається діалог. Якщо CF не визначений, введення користувача буде прийнято з вірогідністю 100 %.

- Другий рядок містить питання діалогу, взяті в одинарні або подвійні лапки. Це питання буде представлено користувачу експертної системи, якщо діалог буде викликаний машиною виведення.

- Для діалогів типів MultChoice, Choice та AllChoice, альтернативні значення, що будуть пред'явлені користувачу експертної системи, записуються по одному в послідовних рядках, укладені в одинарні або подвійні лапки. Для діалогів числового типу на двох наступних рядках записуються мінімальні і максимальні прийнятні значення, взяті в одинарні або подвійні лапки.

Визначення мети: Кожна мета задається в окремому рядку, що починається з GOAL, супроводжуваного назвою цільового атрибута у квадратних дужках. Мети можуть бути визначені перед або після діалогів, але вони повинні слідувати за визначеннями правил. У базі знань повинна міститися принаймні одна установка мети: мети - це атрибути, для яких машина виведення шукає значення. Процес виведення закінчується, коли всі мети були задоволені або є нерозв'язними.

Оператори відношень e2gLite

оператор	інтерпретація	форматований переклад правила	Приклад неформатованого / форматованого запису передумови речення
=	дорівнює	є	<i>[the result of trying the starter] = "nothing happens"</i> the result of trying the starter is nothing happens
<	строго менше ніж	менше ніж	<i>[the age of the battery in months] < 24</i> the age of the battery in months is less than 24

>	строго більше ніж	більше ніж	<i>[the age of the battery in months] > 12</i> the age of the battery in months is greater than 12
!	не дорівнює	не є	<i>[the gas tank] ! "empty"</i> the gas tank is not empty
:	дорівнює будь-якому з (використовується тільки з текстовими даними)	є значення ¹ , значення ² ,... або значення ^N	<i>[the battery brand] : "Delco" "Mopar" "Interstate"</i> the battery brand is Delco, Mopar or Interstate

По обидва боки оператор відношення може бути оточений одним і більш необов'язковими пробілами.

Перекладене значення оператору відношення має братися до уваги, коли створюються ім'я та значення атрибутів, щоб покращити читабельність правил.

За винятком ":" (дорівнює кожному з), цінність багатозначного атрибута з найвищим CF використовується для обчислення виразу. Вираз із оператором "дорівнює кожному з" розглядається як істинний, якщо будь-яке значення атрибута з коефіцієнтом упевненості дорівнює або більше ніж MINCF відповідає будь-якому значенню в списку кількостей порівняння.

Параметри команди TRANSLATE

Команда TRANSLATE дозволяє розробнику бази знань змінювати текст кнопок та повідомлень. Наведемо перелік параметрів команди TRANSLATE та їх значення за замовчанням.

Тексти кнопок:

TRANSLATE B_SUBMIT = "Submit your response"

TRANSLATE B_EXPLAIN = "Explain"

TRANSLATE B_WHYASK = "Why ask?"

TRANSLATE B_RESTART = "Restart"

TRANSLATE B_RETURN = "Return"

Тексти повідомлень:

TRANSLATE TR_KB = "Knowledge base:"

TRANSLATE TR_NORESP = "I don't know/would rather not answer"

TRANSLATE TR_HOWCONF = "How confident do you feel about your response?"

TRANSLATE TR_LOWCONF = "Very uncertain (50%)"

TRANSLATE TR_HICONF = "Very certain (100%)"

TRANSLATE TR_YES = "yes"

TRANSLATE TR_NO = "no"

TRANSLATE TR_FALSE = "false"
 TRANSLATE TR_RESULTS = "FINAL RESULTS:"
 TRANSLATE TR_MINCF = "Minimum confidence factor for accepting a
 value as a fact:"
 TRANSLATE TR_NOTDETERMINED = "could not be determined"
 TRANSLATE TR_ISRESULT = "is:"
 TRANSLATE TR_WITH = "with"
 TRANSLATE TR_CONF = "% confidence"
 TRANSLATE TR_ALLGOALS = "all conclusions"
 TRANSLATE TR_VALUE = "Value"
 TRANSLATE TR_OF = "of"
 TRANSLATE TR_THISRULE = "Rule below fired at CF=" "
 TRANSLATE TR_RULEASGN = "and assigned the value"
 TRANSLATE TR_TOFIND = "To find"
 TRANSLATE TR_AVALUE = "a value for"
 TRANSLATE TR_ISNEEDED = "is needed to try this rule:"
 TRANSLATE TR_RULE = "RULE:"
 TRANSLATE TR_IF = "IF:"
 TRANSLATE TR_THEN = "THEN:"
 TRANSLATE TR_AND = "and"
 TRANSLATE TR_OR = "or"
 TRANSLATE TR_EQUAL = "is"
 TRANSLATE TR_LESSTHAN = "is less than"
 TRANSLATE TR_GREATER = "is greater than"
 TRANSLATE TR_NOTEQUAL = "is not"
 TRANSLATE TR_VALUEFOR = "A value for:"
 TRANSLATE TR_FOUND = "has been determined"
 TRANSLATE TR_NOTFOUND = "has not yet been determined"
 TRANSLATE TR_WASINPUT = "was input with "
 TRANSLATE TR_DETERMINED = "Determined"
 TRANSLATE TR_IS = "is"
 TRANSLATE TR_FROM = "from:"
 TRANSLATE TR_DEFAULTED = "was set by default with"

Коди помилок e2gLite

- 150 - Помилка ініціалізації
- 205 - Зациклення процесу виведення
- 210 - Забагато установок GOAL
- 220 - Немає установки GOAL
- 230 - Забагато атрибутів
- 240 - Забагато невідомих атрибутів

- 250 - Забагато правил
- 260 - Забагато допустимих речень RULE
- 270 - Забагато послідовних речень RULE
- 280 - Немає місця для даних правил
- 290 - Забагато діалогів
- 300 - Немає місця для даних діалогів
- 400 - Очікується а розмежоване ім'я або значення атрибуту на цьому місці
- 410 - Немає кінця бази знань
- 420 - Очікується установка IF на цьому місці
- 430 - Очікується ім'я атрибута на цьому місці
- 440 - Очікується значення на цьому місці
- 450 - Очікується установка THEN на цьому місці
- 460 - Очікується реляційний оператор (= < > ! :) на цьому місці
- 470 - Очікується логічний оператор (and / or) на цьому місці
- 480 - Очікується оператор (=) на цьому місці
- 490 - Очікується AND на цьому місці
- 500 - Очікується коефіцієнт упевненості між 1 та 100 на цьому місці
- 510 - Очікується числове значення MAXVALS на цьому місці
- 520 - Очікується тип діалогу (YesNo, MultChoice,...) на цьому місці
- 530 - Перекладне значення з цим ім'ям не існує
- 540 - Атрибут з цим ім'ям не існує
- 600 - Спробуйте змінити тип атрибута
- 610 - Недопустиме числове значення
- 620 - Погане числове значення в установці MAXVALS
- 630 - Логічний оператор не розпізнаний
- 700 - Тип діалогу не розпізнаний
- 750 - Команда бази знань не розпізнана
- 760 - Нерозпізнаний тип даних
- 800 - Відсутня або порожня база знань

Більшість нумерованих помилок знаходиться коли база знань завантажена. Завантаження припиняється коли перша помилка знаходиться та рядок, який містить помилку відображується у вікні відлагоджувача (DEBUG), якщо запуск у режимі відлагоджувача.

Вивід результатів бази знань

Statement	Опис	Приклади
STARTING NEW CONSULTATION WITH THE FOLLOWING GOALS:	Список усіх ознак, названих в установках GOAL	>the recommendation

Minimum confidence factor for accepting a value as a fact:	Мінімальний коефіцієнт упевненості для прийняття значення як факт. Коли починається нова консультація приводиться після списку цілей.	80%
TRYING RULE:	Опис для правила, яке машина виведення збирається розглядати.	Is it going to rain?
>Add to goal stack	Містить назву атрибута, доданого до стеку мети. Значення цього атрибута є необхідним для обчислення правила, яке машина виведення розглядала останнім рухаючи до статусу Unknown.	precipitation
>Rule status after evaluation is:	Статус правила, яке машина виведення тільки що розглянула. Можливі значення: Unknown, True (істина)/fired (попадання), False (хибність) і Failed (невдача) .	Unknown
PROMPT FOR:	Діалог буде викликаний, щоб одержати значення для названого атрибута.	precipitation
*Prompt assigned:	Значення, отримане від останнього викликаного діалогу, супроводжуване впевненістю, з якою значення було введено.	precipitation=expected (100.0%)
*Rule assigned:	Правило знайдено та у наслідку названа ознака отримала певне значення з рівнем упевненості, показаним у круглих дужках.	the recommendation=wear a raincoat (100.0%)
*Default assigned:	Значення, зазначене в команді бази знань DEFAULT, було призначено названій ознаці з рівнем упевненості, показаним в круглих дужках після того, як машина виведення не зуміла визначити його значення.	the recommendation=wear whatever you want to (90.0%)
FORWARD CHAIN:	Значення атрибута було отримано з діалогу, оскільки знайдено правило, або з установки за замовчуванням з достатньою вірогідністю, що дозволяє розглядати його як факт. Воно буде тепер використовуватися для обчислення придатних правил у базі знань.	precipitation=expected (100.0%)
>Forward chain - trying rule:	Містить назву правила, для якого буде пробуватися процес прямого виведення. Після >Rule status after evaluation is: (статус правила після обчислення) буде впливати виведення трасування.	Is it going to rain?

No eligible rules for:	Значення названого атрибута було знайдене, але відсутні прийнятні правила для прямого пошуку.	the recommendation
REMOVE FROM GOAL STACK	Причина для видалення атрибута зі стека мети, супроводжувана назвою атрибута. Причини: (Resolved-Вирішено) зміст значення був визначений чи (Failed-Невдача) значення атрибута не могло бути визначено.	(Resolved): precipitation
REPLACE ON GOAL STACK:	Коли значення знайдене для багатозначного атрибута (MAXVALS > 1), він видаляється зі стека мети. Якщо існують якінебудь збережені невирішені значення, ознака потрапляє назад у стек мети.	the recommendation
GOAL STACK EMPTY: END CONSULTATION	Стек мети порожній, так що консультація закінчена. Кожна змінна мети була вирішена, або машина виведення вирішила, що вона не може бути вирішена з доступними фактами.	

Трасування бази знань доступне, коли запуск у режимі відлагодження здійснений установкою параметра DEBUG у TRUE в параметрах <APPLET>. Кнопка "**Trace is ON/OFF**" вмикає / вимикає виведення результатів трасування. Ця кнопка може використовуватися в будь-який час протягом консультації.

Вивід аналізу бази знань

Компонент	Опис	Приклади
ATTRIBUTE USAGE	Усі назви ознак, знайдені в базі знань заносяться в список за абеткою (ігнорується регістр символів). Діалоги і правила, що можуть визначати значення атрибута, показуються. Потім у списку в окремих рядках показуються правила, що використовують ознаку. Ознака, що використовується, але ніколи не визначається, представляє логічну помилку та позначається знаком питання перед назвою атрибута. Це, як правило, відбувається коли назва атрибута була написана	[precipitation] Determined by PROMPT 1 [precipitation] Used by rule(s) 1,2 [the recommendation] Determined by rule(s) 1,2 Error (caused by space at end of attribute name in PROMPT): [precipitation] Determined by PROMPT 1 ?[precipitation] Used by rule(s) 1,2 [the recommendation]

	з орфографічною помилкою.	Determined by rule(s) 1,2
VALUE USAGE	У секції звіту VALUE USAGE, усі текстові значення, знайдені в базі знань, перелічуються за абеткою (ігноруючи регістр символів). Кожне значення супроводжується його довжиною, укладеною в круглі дужки. Потім його місце розташування в базі знань (у передумові правила (IF), пропозиції правила (THEN) чи діалозі) ідентифікується поряд зі зв'язаною ознакою. Це виведення допомагає ідентифікувати значення із записом, що слабо відрізняється, інтервалом або іншою пунктуацією, яка робить два значення, що повинні бути відповідними, невідповідними для машини виведення. Довжина кожного атрибута корисна для ідентифікації типографських помилок типу двох пробілів між словами, коли повинний був бути тільки один пробіл.	"expected" (8) prompt 1 value of [precipitation] "expected" (8) rule 1 premise value of [precipitation] "expected" (8) rule 2 premise value of [precipitation] Error (caused by misspelling "expected" in rule 2's premise) "expected" (8) rule 1 premise value of [precipitation] "expected" (9) rule 2 premise value of [precipitation]

Аналізи бази знань доступні, коли запуск у режимі відлагоджувача отриманий встановленням параметру DEBUG у TRUE в параметрах <APPLET>. Для отримання цього виводу натисніть кнопку "**Analyze KB**" у вікні "**KNOWLEDGE BASE DEVELOPER'S OUTPUT**". Аналіз представляє статичну інформацію про базу знань та може бути викликаний у будь-який час впродовж консультації.

Вивід дампа бази знань

Компонент	Опис	Приклади
Minimum CF	Мінімальний коефіцієнт упевненості, що повинний бути досягнутий перед тим, як значення атрибута буде прийняте як відоме.	Minimum confidence factor for accepting a value as a fact: 80%
ATTRIBUTES	Поточне значення атрибута, супроводжуване його типом (S для строкового, N для числового і B для булевого) і вірогідністю в поточний момент.	Attribute 3: the recommendation=wear your boots (S)(95%) DEFAULT: wear

	мент. Якщо база знань включає установку DEFAULT для цього атрибута, приймається значення за замовчуванням, тип і вірогідність, з якою воно зв'язано.	whatever you want to (90%)
RULES	Перший рядок для кожного правила містить опис правила, поточний статус (U для невідомого, T для доведеного істинного, F для доведеного хибного та X для відмовлення (не може бути визначено)), супроводжуваний поточним коефіцієнтом упевненості, зв'язаним із правилом. Кожна наступна передумова (IF:): і наслідок (THEN:) речення представлені поряд з типом значення. Послідовні речення також містять коефіцієнт упевненості, асоційований з установкою значення в круглих дужках.	RULE 1: [Is it going to rain?] Status: U (0.0%) IF: precipitation=expected(S) and the expected temperature>32.0(N) THEN: the recommendation=wear a raincoat(S) (100.0%)
GOAL STACK	Стек мети містить список атрибутів, що машина виведення пробує вирішувати. Остання (з найвищим номером) мета - це поточна мета виведення. Мета видаляється, коли значення визначене або не існує способу визначити значення. Коли стек мети порожній, консультація завершується.	Goal 1: the recommendation Goal 2: the expected temperature

Дамп бази знань є доступним, коли запуск у режимі відлагоджувача отриманий встановленням параметра DEBUG у TRUE в параметрах <APPLET>. Для одержання дампа натисніть кнопку "**Display KB dump**" у вікні "**KNOWLEDGE BASE DEVELOPER OUTPUT**". Дампи можуть бути викликані в будь-який час впродовж консультації.

Якщо атрибуту дозволено мати множинні значення (існує установка MAXVALS для даного атрибута зі значенням більшим за 1), він може з'являтися багато разів у секції ATTRIBUTE дампа.

Додаток Г

Моделювання семантичних мереж за допомогою бібліотеки SNTtoolbox у пакеті MATLAB

Бібліотека SNTtoolbox (Semantic Network Toolbox) містить функції для роботи із семантичними мережами у пакеті MATLAB. Бібліотека розроблена для пакету MATLAB версії 7, проте її функції можуть працювати і в попередніх версіях пакету.

Для роботи із бібліотекою на ЕОМ треба встановити пакет MATLAB (можна встановити тільки ядро пакету) та у окрему директорію на диску (бажано назвати її SNTtoolbox) записати файли бібліотеки SNTtoolbox. У середовищі пакету MATLAB треба вказати шлях до директорії з функціями бібліотеки SNTtoolbox (це можна зробити програмно за допомогою команди `addpath` 'шлях', або за допомогою опції `Set path` підменю `File` головного меню пакету MATLAB).

Процес створення та використання семантичних мереж у пакеті MATLAB при використанні бібліотеки SNTtoolbox полягає у послідовному виконанні таких етапів:

Етап 1. *Побудова семантичної мережі*

1.1. Створення структури семантичної мережі та занесення її до змінної середовища пакету MATLAB.

1.2. Додавання до структури семантичної мережі вузлів.

1.3. Додавання до структури семантичної мережі відношень.

1.4. (Не обов'язково) Збереження семантичної мережі зі змінної середовища MATLAB у файл на диску.

Етап 2. *Візуалізація семантичної мережі.*

2.1. Завантаження семантичної мережі з диску у змінну середовища MATLAB, якщо вона відсутня у ньому.

2.2. Виклик функції візуалізації семантичної мережі.

2.3. (Не обов'язково) Збереження у файлі на диску (або виведення до друку на принтері) побудованої схеми семантичної мережі. Цей етап може бути виконаний як програмно - за допомогою відповідних функцій пакету MATLAB, так і за допомогою графічного інтерфейсу користувача із використанням команд меню фігури, на якій зображено схему семантичної мережі.

Етап 3. *Пошук у семантичній мережі.*

3.1. Для семантичної мережі - бази знань сформувати семантичну мережу-запит у окремій змінній середовища MATLAB. Для цього потрібно для мережі-запиту виконати підетапи 1.1-1.4 або скопіювати мережу бази знань до змінної мережі-запиту та видалити з неї зайві вузли і зв'язки та додати вузол мети із відповідними зв'язками.

3.2. Викликати функцію пошуку, яка поверне результати пошуку.

Розглянемо основні *функції бібліотеки*.

Функція $SN=Snew$ створює нову структуру для семантичної мережі у змінній SN .

До *структури семантичної мережі* змінної SN будуть входити такі поля: $SN.node$ - клітковий (cell) одновимірний масив, що містить назви вузлів мережі, які є рядками; $SN.relation$ - клітковий (cell) двовимірний масив, що містить назви відношень між вузлами мережі, які є рядками (рядки масиву відповідають вузлам від яких, а стовпці - вузлам до яких направлені відношення); $SN.nodetype$ - звичайний одновимірний масив, що містить коди типів вузлів мережі (0 - AND-вузол, 1 - OR-вузол);

Вузли семантичної мережі у бібліотеці поділяються на два типи: AND-вузли та OR-вузли.

AND-вузли (ТА-вузли), як правило, повинні зіставлятися поняттям, об'єктам, назвам атрибутів. У процесі виведення AND-вузли вважаються істинними (спрацювавшими, конкретизованими) у одному з таких випадків: 1) якщо вони є як у мережі бази знань, так і у мережі-запиті; 2) якщо всі вузли, що є їхніми батьками у ієрархії (тобто мають зв'язки направлені до них) є істинними.

OR-вузли (АБО-вузли), як правило, повинні зіставлятися значенням атрибутів. У процесі виведення OR-вузли вважаються істинними (спрацювавшими, конкретизованими) у одному з таких випадків: 1) якщо вони є як у мережі бази знань, так і у мережі-запиті; 2) якщо існує більше одного батьківського вузла, та хоча б один з них є істинним. Якщо OR-вузол має лише один батьківський вузол, то він не буде наслідувати його істинність.

Функція $Res=SNaddANDnode(SN, 'node1', 'node2', ..., 'nodeN')$ додає вузли типу "AND" $'node1', 'node2', ..., 'nodeN'$ до семантичної мережі змінної SN (змінна SN при цьому не модифікується). Результат повертає у змінній Res .

Функція $Res=SNaddORnode(SN, 'node1', 'node2', ..., 'nodeN')$ додає вузли типу "OR" $'node1', 'node2', ..., 'nodeN'$ до семантичної мережі змінної SN (змінна SN при цьому не модифікується). Результат повертає у змінній Res .

Функція $Res=SNdelnode(SN, 'node1', 'node2', ..., 'nodeN')$ видаляє вузли $'node1', 'node2', ..., 'nodeN'$ із семантичної мережі змінної SN (змінна SN при цьому не модифікується). Результат повертає у змінній Res .

Функція $Res=SNaddrelation(SN, 'node1', 'relation', 'node2')$ додає відношення $'relation'$ від вузла $'node1'$ до вузла $'node2'$ семантичної мережі змінної SN (змінна SN при цьому не модифікується). Результат повертає у змінній Res .

Функція $Res=SNdelrelation(SN, 'node1', 'relation', 'node2')$ видаляє відношення $'relation'$ від вузла $'node1'$ до вузла $'node2'$ семантичної мережі змінної SN (змінна SN при цьому не модифікується). Результат повертає у змінній Res .

Функція *SNplot(SN, type, NodeColour, RelationColour)* будує графічне зображення структури семантичної мережі змінної *SN*. Параметр *type* має бути рядком та задає тип розташування вузлів семантичної мережі на графіку: 'circle' - вузли розташовуються по колу, 'random' - випадкове розташування вузлів, 'hierarchy' - ієрархічне розташування вузлів. Параметри *NodeColour* та *RelationColour* є необов'язковими. *NodeColour* має бути рядком та визначає колір вузлів мережі. *RelationColour* має бути рядком та визначає колір зв'язків мережі. Кольори мають бути задані символами подібно до стандартної функції *plot*: 'b' - blue (синій), 'g' - green (зелений), 'r' - red (червоний), 'c' - cyan (світло-голубий), 'm' - magenta (бузковий), 'y' - yellow (жовтий), 'k' - black (чорний).

Функції *Sncircleplot*, *Snhierarchyplot* та *Snrandomplot* є внутрішніми і використовуються функцією *SNplot* для побудови графіків із круговим, ієрархічним та випадковим розташуванням вузлів, відповідно.

Функція *Res=SNfind(SN, SN1)* шукає цільовий вузол семантичної мережі змінної *SN1* у семантичній мережі змінної *SN*. Мережа змінної *SN1* має бути підмережею мережі змінної *SN*. Цільовий вузол у змінній *SN1* має мати назву '?'. Функція повертає результат у змінній *Res*. Результатом є значення [], якщо цільовий вузол не був знайдений, або ім'я цільового вузла, якщо він був знайдений.

Функція *h=SNhierarchy(SN)* повертає у змінній *h* масив з номерами рівнів ієрархії вузлів семантичної мережі змінної *SN*. Номери рівнів ієрархії вузлів є цілими числами у діапазоні [1, K], де K - кількість встановлених рівнів ієрархії. Чим вище знаходиться вузол в ієрархії тим менший рівень ієрархії він має. Вузли одного рівня мають однакові номери рівня ієрархії.

Функція *k=isnodepresent(S, 'node')* повертає у змінну *k* номер вузла з ім'ям 'node' у одновимірному клітковому масиві рядків *S*. Якщо вузол 'node' у одновимірному клітковому масиві рядків *S* відсутній, тоді функція повертає 0.

Функції бібліотеки висувають певні *вимоги* до семантичної мережі:

1. Кожен вузол повинний мати унікальне ім'я.
2. Мережа повинна мати вузли із вказаним типом (AND або OR).
3. Може бути тільки один цільовий вузол для пошуку.
4. Забезпечується тільки низхідне виведення - від передумов до висновків.

5. Функція *SNfind*, функція *SNhierarchy* та функція *SNplot* у режимі 'hierarchy' працюють виключно з ієрархічними (шаруватими) мережами. Тобто для коректного використання цих функцій не можна задавати повнозв'язні мережі, мережі де є латеральні зв'язки (зв'язки між вузлами одного шару) та мережі з елементами, які пов'язані самі з собою.