

Міністерство освіти і науки України
Національний університет водного господарства та
природокористування
Навчально-науковий інститут енергетики, автоматики та
водного господарства

Кафедра автоматизації, електротехнічних та комп'ютерно-
інтегрованих технологій

04-03-392М

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з навчальної дисципліни
«Інтелектуальні системи управління та пристрої»
(Частина 1. Подання знань в інтелектуальних системах
управління)
для здобувачів вищої освіти другого (магістерського) рівня
за освітньо-професійною програмою
«Автоматизація, комп'ютерно-інтегровані технології та
робототехніка» спеціальності 174 «Автоматизація,
комп'ютерно-інтегровані технології та робототехніка»
денної та заочної форм навчання

Рекомендовано науково-методичною
радою з якості ННІ ЕАВГ
Протокол № 12 від 30 серпня 2024 р.

Рівне – 2024

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Інтелектуальні системи управління та пристрої» (Частина 1. Подання знань в інтелектуальних системах управління) для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» денної та заочної форми навчання. [Електронне видання] / Стець С. Є. – Рівне : НУВГП, 2024. – 30 с.

Укладач: Стець С. Є., к.т.н., доцент кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.

Відповідальний за випуск: Древецький В. В., д.т.н., професор, завідувач кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.

Керівник групи забезпечення спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»:
Рудик А. В., д.т.н., професор кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.

© С. Є. Стець, 2024
© Національний університет
водного господарства та
природокористування, 2024

Лабораторна робота № 1

Використання семантичних мереж для подання знань в інтелектуальних системах управління

Мета роботи

Навчитися створювати та застосовувати семантичні мережі для подання знань в інтелектуальних системах.

Теоретична частина

Проблема представлення знань є однією з центральних при побудові системи штучного інтелекту. Нагадаємо собі, що ж таке «знання». Із визначень, наведених в літературі, найбільш змістовним є наступне: **знання** - це закономірності наочної області (принципи, закони, зв'язки), отримані в результаті практичної діяльності і професійного досвіду, які дозволяють фахівцям ставити і вирішувати завдання в певній галузі.

Розрізняють такі види знань:

декларативні — факти, дані, параметри, властивості деяких об'єктів і т.д., рішення задачі, що не містять в явному вигляді процедури;

процедурні — алгоритми, правила, методики, що відповідають на питання: «Як вирішити завдання?»;

метазнання - знання про знання, що володіють високою мірою спільності.

Властивості знань:

- інтерпретація, тобто наявність семантики, сенсу;
- концептуальність, тобто понятійність, смислова виразність на рівні понять, що інтерпретують ці знання;
- структурованість, тобто впорядкованість за деяким принципом (наприклад, «частина - ціле», «клас - елемент» і так далі);
- зв'язність, тобто наявність різного роду зв'язків між знаннями (причинно-наслідкові, атрибутивні, оцінювані, тимчасові зв'язки і так далі);
- міра узгодженості (несуперечності).

Всі знання в системі штучного інтелекту (у контексті досліджуваної наочної області) зберігаються в базі знань.

Об'єктивній оцінці ситуації відповідають певні моделі знань, які умовно поділяються на концептуальні та емпіричні.

Семантична мережа - один із способів представлення знань. Спочатку семантична мережу розроблялася як модель уявлення довгострокової пам'яті у психології, але згодом стала одним із способів представлення знань у експертних системах.

Семантика - спільні відносини між символами та об'єктами цих символів.

Семантичні мережі є однорідним класом моделей представлення знань. Часто загальною основою для віднесення схеми представлення знань до семантичної мережі є те, що вона представлена у вигляді орієнтованого графа, вершини якого відповідають об'єктам (поняттям, сутностям) предметної області, а дуги – відносинам (зв'язкам) між ними. І вузли, і дуги, зазвичай, мають мітки (імена). Імена вершин і дуг зазвичай збігаються з іменами відповідних об'єктів та відносин предметної області.

Об'єкти предметної області, що відображаються в семантичній мережі, можна умовно поділити на три групи: узагальнені, індивідуальні (конкретні) та агрегатні.

Узагальнений об'єкт відповідає деякій абстракції збірної реальної об'єкта, процесу або явища предметної області. Наприклад, "виріб", "підприємство", "співробітник" і т. д. Узагальнені об'єкти фактично представляють певні класи предметної області.

Індивідуальний об'єкт є одиничним представником (примірником) класу, виділеного певним чином. Наприклад, «співробітник І. М. Петрів».

Агрегат - складний об'єкт, утворений з інших об'єктів, що розглядаються як його складові. Наприклад, виріб складається з набору деталей, підприємство складається із набору відділів, служб, цехів.

Введена класифікація об'єктів є відносною. Залежно від розв'язуваного завдання той самий об'єкт може розглядатися як узагальнений чи індивідуальний, як агрегатний чи неагрегатний.

Типи зв'язків між об'єктами семантичних мереж можуть бути будь-якими. Але найчастіше використовуються такі основні зв'язки (відносини): «рід-вид», «є представником», «є частиною».

Наявність зв'язку «рід-вид» між узагальненими об'єктами А і означає, що поняття А є більш загальним, ніж поняття В. Будь-який об'єкт, що відображається поняттям В, відображається і поняттям А, але не навпаки. Наприклад, поняття "тварина" є родовим поняттям для об'єкта "птиці". Як правило, всі властивості родового об'єкта А притаманні видовому об'єкту В. Інакше кажучи, об'єкт успадковує властивості об'єкта А.

Зв'язок «є представником» існує звичайно між узагальненим та індивідуальним об'єктом, коли індивідуальний об'єкт виступає в ролі представника деякого класу. Так, індивідуальний об'єкт «ноутбук Samsung» є представником (екземпляром) узагальненого об'єкта «ноутбук». Екземпляр може бути представником декількох узагальнених об'єктів. У цьому випадку йому притаманні властивості декількох узагальнених об'єктів, що відповідають множинному успадкуванню.

У ряді випадків між зв'язками «рід-вид» і «є представником» не роблять розбіжностей, відзначаючи, що ці зв'язки задають відношення «загальне-частка». Іноді це приводить до непорозумінь. Тому для формалізації таких зв'язків будемо використовувати відносини *ako* (від англ. *a-kind-of* - різновид) і *is a* (від англ. *is a member of the class* - бути представником класу).

Не менш важливе відношення «є частиною» (англ. *part of*). Дане відношення зв'язує агрегатний об'єкт із його складовими частинами. Воно дозволяє відбивати в базі знань структуру об'єктів предметної області. Іноді дане відношення позначають міткою «має» (англ. *has*).

Вперше семантичні мережі з'явилися в області машинної лінгвістики як засіб аналізу змісту (семантики) природної мови. М.Р. Куїлліаном була розроблена програма, що фіксує зміст англійських слів, побудована за принципом тлумачного словника. У цій програмі деяке слово англійської мови визначається в термінах, що виражаються іншими словами. Кожне слово визначається множиною покажчиків на інші слова. Сукупність вказівників утворює мережу, «подорожуючи» у якій можна з'ясувати зміст того чи іншого слова. У семантичній

мережі М.Р. Куїлліана вершини відповідають обумовленому поняттю і забезпечуються вказівниками на інші слова, що розкривають дане поняття. База знань організована у вигляді сторінок (площин), на кожній з яких представляється граф, що визначає одне слово. На рис. 1.1 зображені три різних визначення англійського слова «*plant*». Вказівники, позначені на рис. 1.1 суцільними лініями, формують наступні визначення слова «*plant*»;

1) жива (*live*) структура (*structure*), що не відноситься до тварин (*animal*), часто має листи (*leaf*), одержує (*get*) харчування (*food*) з (*from*) повітря (*air*), води (*water*) і ґрунту (*earth*);

2) устаткування (апаратура – *apparatus*), використовуване (*use*) для (*for*) реалізацій будь-якого процесу (*process*) у (*in*) промисловості (*industry*);

3) помішати (*put*) щось (*seed* – насіння, *plant* – рослина і т.п.) у ґрунт (*earth*) для (*for*) вирощування (*grow*).

Перше визначення (*plant 1*) відповідає іменнику «рослина», друге (*plant 2*) – іменнику «завод», третє (*plant 3*) – дієслову «саджати». Вказівники, позначені на рисунку штриховими лініями, посилаються на слова-поняття, представлені на інших сторінках.

Один із механізмів висновку, який використовується в обговорюваній моделі семантичної мережі, заснований на пошуку перетину. Це дозволяє встановити зв'язок між двома словами. Пошук виконується за допомогою методу «в ширину» з використанням зовнішніх покажчиків з двох вихідних сторінок, доки не буде знайдено спільну концепцію, яка відповідає вершині перетину двох напрямків пошуку. Його знаходять шляхом пошуку шляху від початкових вершин до вершини перетину та представляють зв'язки між вихідними словами-концептами.

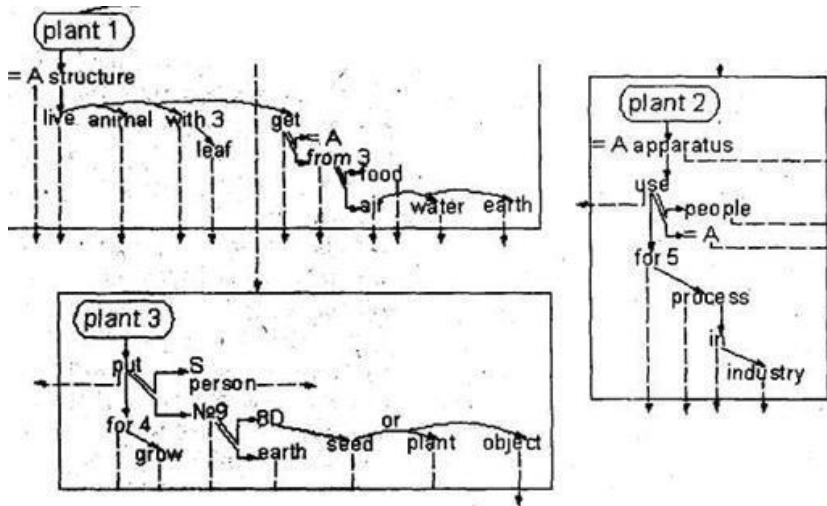


Рис. 1.1. Визначення слова «plant»

Наприклад, слова-поняття «cry» (плач, плакати) і «comfort» (розрада, утішати) приводять до перетинання шляхів пошуку у вершині «sad» (сумний, сумовитий). У результаті аналізу шляхів пошуку модель дозволяє зробити наступний висновок: «Плач – це один зі способів створення сумних звуків. Утішати – робити щось менш сумним».

Таким чином, модель дозволяє встановити представлені в ній зв'язки, отже, на її основі можна отримати нові знання. Це дозволяє створювати системи розуміння природної мови, які мають такі можливості:

1) визначення основного змісту тексту шляхом пошуку безлічі вершин перетину;

2) вибирати необхідне значення багатозначного слова, виходячи з найкоротшого шляху від цього слова до інших слів пропозиції (різновид семантичної метрики);

3) Формування відповіді різні запити шляхом встановлення зв'язку між словами-поняттями запиту і словами-поняттями, які у пам'яті системи.

Розглянутий підхід одержав деякий розвиток у системі TLC (*Teachable Language Comprehender* - навчальна система

розуміння мови), що розроблена Куїлліаном. Однак практичний успіх TLC був обмежений. Це тому, що набір використовуваних типів зв'язків (відносин) між поняттями був недостатній. Використовувалися тільки такі зв'язки, як: «клас - екземпляр класу», «атрибут – значення». Окрім цього, сам пошук не враховував зміст зв'язків. Представлення знань вимагає введення в розгляд більш багатого набору відносин між поняттями (об'єктами, сутностями) предметної області.

Подальший розвиток мережових моделей представлення знань здійснювався саме в цьому напрямку. Роботи Р.Симмонса і К.Филмора вказали на важливу роль дієслів при аналізі змісту пропозицій природної мови. Пропозицію можна представити вершиною-дієсловом і різними відмінковими зв'язками (відносинами). Таку структуру називають відмінковою рамкою. Серед відмінкових відносин виділяють наступні: агент – відношення між подією і тим хто (що) його робить; об'єкт – відношення між подією і тим, над чим виконується дія; інструмент – об'єкт, за допомогою якого відбувається дія; місце – місце здійснення події; час – час здійснення події. При аналізі пропозиції програма знаходить дієслово і встановлює відповідні відмінкові відносини між частинами пропозиції.

Як приклад на рис. 1.2 зображена відмінкова рамка пропозиції: «Іван закріпив деталь клеєм». Відмінкова рамка фіксує знання відмінкової (лінгвістичної) структури природних мов у вигляді мережного формалізму.

У відповідності з введеним раніше визначенням семантичної мережі, відсутні будь-які обмеження як на типи відносин, так і на типи об'єктів, відображуваних у мережі. У більшості випадків різноманіття об'єктів мережі можна розділити на три групи:

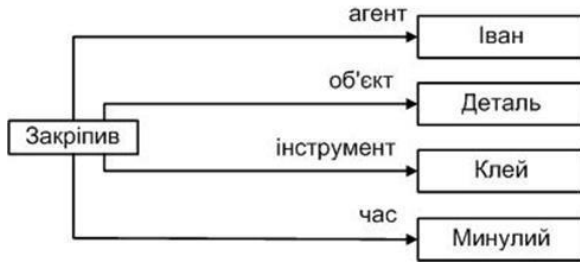


Рис. 1.2. Відмінкова рамка

1. об'єкта-поняття — зведення про фізичні й абстрактні об'єкти, предметну область;
2. об'єкта-події — абстрактні чи конкретні дії, що можуть привести до зміни стану предметної області;
3. об'єкта-властивості — уточнюють поняття і події, наприклад, вказують характеристики понять (колір, форму, розміри і т. п.), фіксують параметри подій (місце, час, тривалість).

Отже семантичні мережі - це графічні моделі у вигляді графів, вершинам (вузлам) яких відповідають деякі поняття (об'єкти, події, процеси), а зв'язкам (тобто дугам графа) - відносини між цими поняттями. Вузли мережі зображуються колами, прямокутниками, а зв'язки між ними лініями зі стрілками (рис. 1.3). Дані зв'язку можуть різнитися між собою, описуючи різні відносини: з спадкоємства; тимчасові співвідношення; просторове розташування; причинно-наслідкові зв'язки.

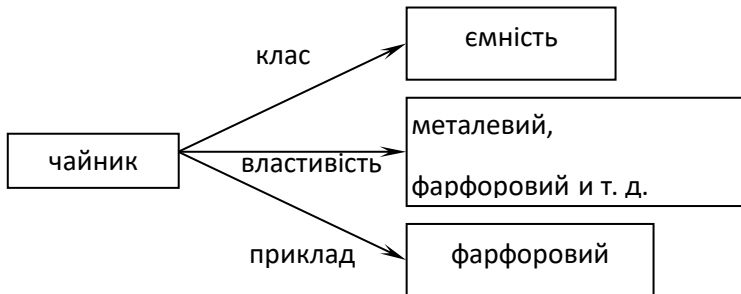


Рис. 1.3. Найпростіший зразок семантичної мережі

Вершини - це об'єкти, дуги - це відносини. Семантична модель не розкриває сама по собі яким чином здійснюється подання знань. Тому семантична мережа розглядається як метод подання знань та структурування знань.

Різноманіття відносин, що використовуються у семантичних мережах, підрозділяються на наступні групи:

1. лінгвістичні відносини, що, у свою чергу, підрозділяються на відмінкові (агент, об'єкт, інструмент, час, місце), дієслівні (нахилення, час, вид, число, застава) і атрибутивні (колір, розмір, форма і т.п.);

2. логічні відносини (кон'юнкція, диз'юнкція, заперечення, імплікація);

3. теоретико-множинні відносини містять у собі відносини типу «множина-підмножина» («рід-вид», «клас-підклас»), «ціле-частина», «елемент множини» і ін.

4. квантифіковані відносини, що підрозділяються на логічні квантори спільності й існування, нечіткі квантори (багато, трохи, часто т.д.).

Таким чином, розглянуті раніше відносини, («рід-вид», «є представником», «є частиною», відмінкові відносини) далеко не вичерпують усього набору відносин, що застосовуються у семантичних мережах. Але вони утворюють гарну основу для побудови прикладних баз знань. Особливе місце при цьому займають теоретико-множинні відносини, що володіють транзитивними властивостями. Відношення R називається транзитивним, якщо для будь-яких об'єктів α, β, γ , таких, що α знаходиться у відношенні $Rc\beta(aR\beta)i\beta$ – у відношенні $Rc\gamma(\beta R\gamma)$, впливає справедливність твердження “ α знаходиться у відношенні $Rc\gamma$ ”, тобто $aR\beta \wedge \beta R\gamma \Rightarrow aR\gamma$.

Наприклад, транзитивність родо-видових відносин забезпечує можливість спадкування властивостей від родового об'єкта до виду і підвиду. Механізм спадкування властивостей забезпечує проведення найпростіших дедуктивних висновків типу: «Усі люди смертні. Сократ – людина. Отже, Сократ смертний».

У базах знань виділяють інтенціональні і екстенціональні знання. Якщо є кінцева множина атрибутів $A = \{A_1, A_2, \dots, A_n\}$ і

кінцева множина відносин $R = \{ R_1, R_2, \dots, R_m \}$, то схемою чи інтенціоналом відносини $R_i \{i=1,2,\dots,m\}$ називається набір пар виду.

$$\text{INT}(R_i) = \{ \dots, [A_j, \text{DOM}(A_j)], \dots \},$$

де $\text{DOM}(A_j)$ – домен атрибута A_j , тобто множина можливих значень атрибута.

Екстенціонал відносини R_i , – це множина фактів, $\text{EXT}(R_i) = \{ F_1, F_2, \dots, F_p \}$, де F_1, F_2, \dots, F_p – факти відносини R_i , що задаються звичайно у вигляді сукупності пар «атрибут-значення». Факт конкретизує визначене відношення. У графічній інтерпретації він являє собою підграф, що має зіркоподібну структуру. Коренем підграфа є вершина предикатного типу з міткою, що відповідає імені відносин. Ребра підграфа відзначені іменами атрибутів.

Інтенціональна семантична мережа представляє предметну область, що моделюється на узагальненому, концептуальному рівні, а екстенціональна – наповнює її конкретними, фактичними даними. Таким чином, семантичну базу знань можна розглядати як сукупність об'єктів і відносин, частина з яких визначена інтенціонально, а частина – екстенціонально.

Способи опису семантичних мереж і логічний вивід.

Познайомившись із загальними поняттями семантичних мереж, розглянемо методи опису семантичних мереж. Найчастіше цих цілей використовуються концептуальні графи Дж. Сува і блокові структури Х. Хендрикса. Тому розглянемо опис семантичних мереж як концептуальних графів.

Вершинами концептуального графа є об'єкти (поняття, сутності) предметної галузі чи концептуальні відносини. Ребра концептуального графа з'єднують поняття-вершини-поняття та відносини-вершини-напрямки. У той самий час ребра можуть з концепції вершини і закінчуватися у бік вершин відносини, і навпаки. інжир. 1.4 Вершини «куля» та «червона» відповідають вершинам понять, а вершина «колір» відповідає бінарному відношенню. Щоб розрізнити ці типи вершин концептуальних графах, вони зображуються прямокутником і еліпсом відповідно.

Однією з переваг поділу відносин на незалежні вершини концептуального графа є спрощення уявлення пов'язаних

відносин. Такий зв'язок зображується n-ребрами-вершинами-версусами (рис. 1.4).

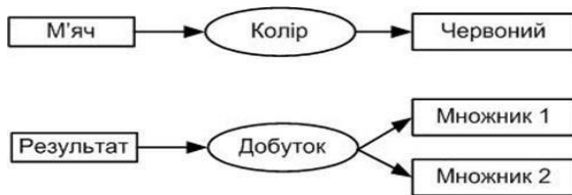


Рис. 1.4. Приклади концептуальних графів

Зазвичай кожний концептуальний граф фіксує одне речення. Тоді база знань буде представлятись у вигляді сукупності таких графів. Наприклад, верхній граф на рис. 1.4 відповідає реченню «М'яч має червоний колір». Граф, що зображений на рис. 1.5, відображає більш складне речення: «Іван закріпив деталь стільця клеєм».

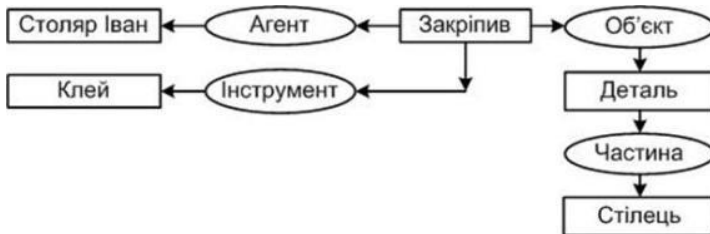


Рис. 1.5. Концептуальний граф пропозиції

Цей граф використовує відмінкові відносини дієслова «закріпити» і демонструє можливості концептуальних графів для представлення пропозицій природної мови.

Кожне поняття-вершина-поняття концептуального графа може мати мітку типу. Тип позначає клас приналежності вершини. На концептуальному графі мітку типу вершини відокремлюють двокрапкою від конкретного імені вершини. На рис. 1.5 за допомогою мітки типу показано, що Іван є столяром. На концептуальних графах можна також вводити індивідуальні поняття-вершини-поняття з однаковими іменами. Щоб розрізнити екземпляри об'єктів (понять) з однаковими іменами,

використовується спеціальний числовий маркер, перед яким записаний символ #. Граф на рис. 1.6 вказує, що конкретний “автомобіль” з маркером #1235 має зелений колір.

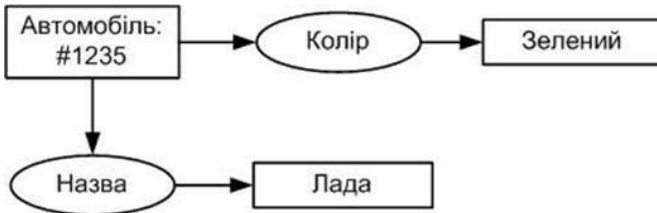


Рис. 1.6. Концептуальний граф з числовим маркером

Крім числового маркеру, на концептуальних графах можуть застосовуватися узагальнені маркери, які позначаються знаком *. У сполученні зі змінною, що записується після цього знаку, узагальнений маркер виявляється корисним у ситуаціях, коли дві різні вершини графа представляють той самий об’єкт. На відміну від числових маркерів, що застосовуються для виділення в семантичній мережі індивідуальних об’єктів, узагальнені маркери дозволяють виділяти той самий узагальнений об’єкт заданого типу (рис. 1.7). Як приклад на рисунку зображений концептуальний граф, що відповідає пропозиції: «Маляр забруднив свою руку фарбою».



Рис. 1.7. Концептуальний граф з загальним маркером

На концептуальних графах вводяться операції, що дозволяють виконувати їхнє перетворення. Нові графи виходять за допомогою чотирьох операцій: копіювання, спеціалізації, об’єднання і спрощення. Розглянемо їх докладніше.

Операція копіювання дозволяє сформувати з вихідного графа G його копію, наприклад, G_1 .

Операція спеціалізації полягає в заміні узагальненої вершини її індивідуальним (конкретним) варіантом. При цьому може бути два випадки:

1) якщо вершина відзначена узагальненим маркером, то узагальнений маркер замінюють конкретним індивідуальним маркером (іменним чи числовим).

2) тип вершини можна замінити відповідними підтипами, що володіють необхідними властивостями.

Операція об'єднання дозволяє одержувати з двох графів один. Якщо на графі G_1 є вершина-поняття V_1 , що аналогічна вершині-поняттю V_2 графа G_2 , то можна сформувати новий граф G_3 видаливши V_2 . При цьому всі зв'язки V_2 необхідно переорієнтувати з V_2 на V_1 .

Операція спрощення дозволяє виключити на графах дублікати відносин. При цьому виключаються ставлення-вершина-відношення і зв'язані з нею дуги. Дублювання відносин часто виникає в результаті виконання операції об'єднання.

Операція спеціалізації дозволяє зіставляти дві вершини концептуального графа і, якщо співставлення успішне, виконувати об'єднання. Спільне використання операцій спеціалізації й об'єднання забезпечує реалізацію механізму спадкування. Наприклад, замінивши узагальнений маркер індивідуальним, ми поширюємо властивості типу на конкретний індивідуальний об'єкт. Якщо у ході виконання операції обмеження відбувається заміна типу на його підтип, то має місце спадкування «підклас-клас-підклас».

Операції спеціалізації й об'єднання конкретизують вихідний граф. Якщо граф G_1 – це конкретизація графа G_2 , то можна сказати, що істинне і зворотне, тобто G_2 – є узагальнення G_1 . Узагальнення використовується при навчанні систем штучного інтелекту, дозволяючи виводити правдоподібні умови за приватними прикладами.

Розглянуті вище найпростіші операції не гарантують одержання правильних тверджень на основі перетворених графів. Конкретизація вершин-понять, що виконувалася в ході операцій

спеціалізації й об'єднання, не завжди відповідає дійсності. Наприклад, якщо в ході виконання операції об'єднання з'єднуються два графи, агенти яких позначені тим самим поняттям, то це зовсім не означає, що не може статись так, що ці поняття відповідають двом різним індивідуальним об'єктам. Наприклад, пропозиції «собака гризе кістку» і «собака знаходиться в парку» можна об'єднати в одне, якщо мова йде про одну й ту саму тварину, а не про двох різних тварин. Проте, розглянуті операції не приводять до тверджень, позбавлених змісту, і дозволяють виконувати правдоподібні міркування.

Концептуальні граfi можуть містити вершини, що є пропозиціями. Такі вершини зображуються у вигляді прямокутного блоку, що містить підграф, який відповідає пропозиції. Як приклад на рис. 1.8 зображений граф твердження: «Олексій думає, що Тетяні подобається іграшка».

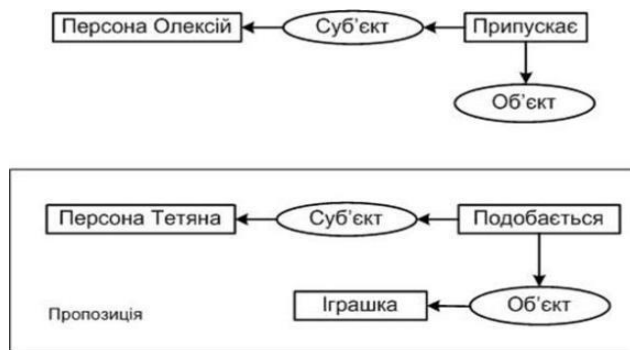


Рис. 1.8. Концептуальний граф, що містить пропозицію

Введення окремих блоків для позначення пропозицій дозволяє фіксувати відносини між пропозиціями. У приведеному прикладі дієслово «думає» представляє не конкретну фізичну дію, а виражає відчуття Олексія. Тому на графі введено нове відношення «суб'єкт», яке представляє обличчя, що випробовує ті чи інші відчуття. Твердження, що містять подібні лексичні одиниці (довіряти, вірити, мати намір і т.п.) є об'єктом досліджень модальної логіки.

Концептуальний граф дозволяє природним чином виражати кон'юнкцію. Наприклад, за допомогою концептуального графа ми легко можемо представити пропозицію «собака знаходилася в саду і гризла кісту». Складніше з іншими логічними операціями: диз'юнкцією, логічним запереченням, кванторами.

Для представлення заперечення на концептуальних графах застосовується логічний оператор «НЕ». Оператор застосовують до пропозиції. Твердження, що фіксується пропозицією, вважається в цьому випадку помилковим. Граф, зображений на рис. 1.9, відповідає пропозиції: «Немає іграшок, що подобаються Тетяні».

Використовуючи заперечення і кон'юнкцію, можна представити на концептуальних графах диз'юнкцію. Однак для спрощення концептуальних схем диз'юнкцію представляють у вигляді спеціального бінарного відношення «чи», аргументами якого є вершини-пропозиції.

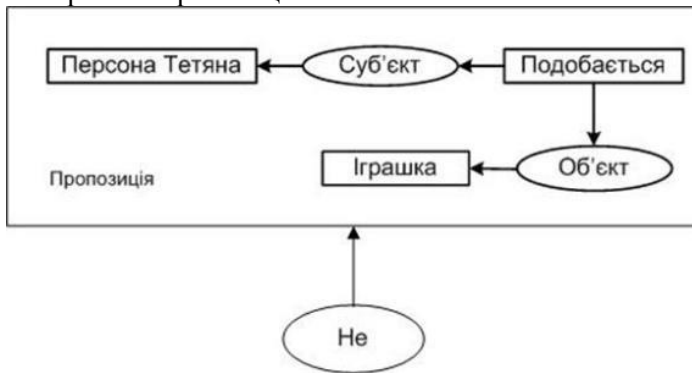


Рис. 1.9. Концептуальний граф з логічним запереченням

Узагальнені поняття-об'єкти-поняття, що відображаються на концептуальних графах, зв'язані квантором існування. Так, граф, зображений на рис. 1.4, можна описати формулою

$$\exists x \exists y (m'яч(x) \wedge колір(x, y) \wedge червоний(y)).$$

Квантор спільності виходить при запереченні пропозицій, зв'язаних квантором існування. Наприклад, застосувавши заперечення до графа, зображеного на рис. 1.4, одержимо

$\exists x \exists y \neg (m'яч(x) \wedge колір(x, y) \wedge червоний(y))$.

За своїми виразними можливостями концептуальні графи еквівалентні логіці числення предикатів. Існує можливість взаємного переходу від описів предметної області мовою числення предикатів до концептуальних графів, і навпаки. Вибір тих чи інших засобів визначається характером розв'язуваних задач і схильностями розроблювачів систем штучного інтелекту.

Методи виводу на семантичних мережах використовують асоціативні й зіставні алгоритми, що зводяться до перебування шляхів на графі, побудові транзитивних замикань, виділенню підграфів з визначеними властивостями. Основні підходи до побудови процедур виводу на семантичних мережах базуються на простих операціях над графами: видалення і додавання нових вершин і ребер, спеціалізація, пошук вершини чи ребра за іменем, перехід від однієї вершини до іншої за зв'язками, об'єднання підграфів і ін.

Одним із простих засобів виводу на семантичних мережах є пошук за перетинанням, що був розглянутий раніше.

Іншим потужним засобом виводу є співставлення зі зразком. У даному випадку відбувається співставлення окремих фрагментів семантичної мережі. При цьому запит до бази знань представляється у вигляді автономного підграфа, що будується за тими ж правилами, що і семантична мережа. Пошук відповіді на запит реалізується співставленням підграфа запиту з фрагментами семантичної мережі. Для цього здійснюють накладання підграфа запиту на відповідний фрагмент мережі. Успішним буде те накладення, у результаті якого фрагмент мережі виявиться ідентичним підграфу запиту. При цьому допускається використання в запиті змінних. Змінна запиту зіставляється з константою фрагмента мережі.

Нехай, наприклад, є запит: «чи існує такий птах X, що вміє співати і має жовтий колір?» (рис. 1.10).



Рис. 1.10. Підграф запиту

У загальному випадку в підграфі запиту можуть бути задані об'єкти, атрибути, імена відносин, не представлені в базі знань явно. Це вимагає виконання попередніх перетворень фрагментів семантичних мереж, що зіставляються. У результаті таких перетворень можуть бути отримані нові зв'язки. Зазначені перетворення виконуються на основі найпростіших базових операцій, розглянутих раніше.

Методи виводу на семантичних мережах, що використовують ідею перетинання шляхів співставлення фрагментів мереж, мають істотний недолік. Він зв'язаний з комбінаторним ростом числа зіставлень перетинань у мережах досить великої розмірності.

У ряді систем, заснованих на семантичних мережах, використовуються спеціалізовані правила виводу.

При розширенні семантичної мережі в ній виникають інші відносини: IS –A (належить) і PART OF (є частиною), тобто відношення: ціле → частина.

Для відображення ієрархічних відносин між об'єктами і введення єдиної семантики в семантичні мережі було запропоновано використовувати процедурні мережі. Мережа будується на основі класу (поняття), вершини, дуги і процедури в ній представлені як об'єкти.

Наведемо як приклад наступний текст: «Комп'ютер складається з процесора, пам'яті, пристрою вводу-виводу інформації та пристрою управління. Процесор зчитує дані з пам'яті і передає отримані результати для запису в пам'ять. Дані також надходять в пам'ять з пристрою вводу-виводу інформації та можуть зчитуватися з пам'яті за допомогою цього пристрою.

Робота процесора, пам'яті і пристрою вводу-виводу управляється за допомогою пристрою управління».

Уявімо інформацію, що міститься в даному тексті, за допомогою семантичної мережі (рис. 1.11). В якості вузлів цієї мережі виступають поняття «Комп'ютер», «Процесор», «Пам'ять», «Пристрій вводу-виводу», «Пристрій управління». Зв'язки (відносини) між цими поняттями називаються: «має», «записує дані», «зчитує дані», «управляє».

Можна стверджувати, що дана семантична мережа визначає деякий клас об'єктів («Комп'ютер»), званий структурою. Дана структура характеризується чотирма атрибутами («Процесор», «Пам'ять», «Пристрій вводу-виводу», «Пристрій управління»). Конкретний об'єкт (наприклад, «Персональний комп'ютер HP Compaq nx9010») належить даній структурі, володіючи всіма перерахованими атрибутами.

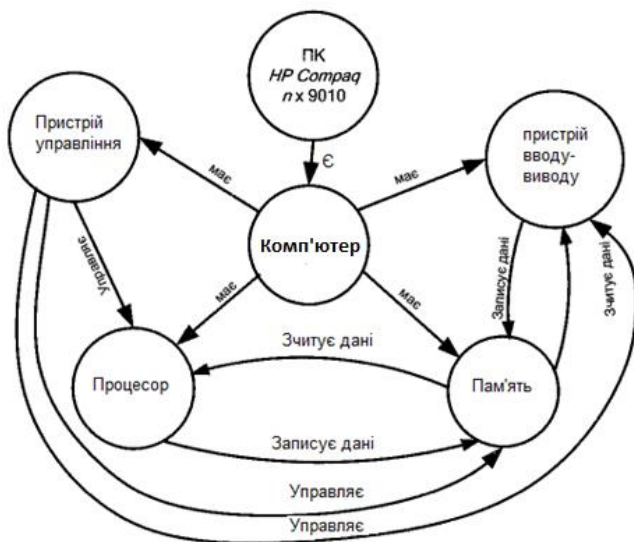


Рис. 1.11. Семантична мережа «Комп'ютер»

Для опису (подання в ЕОМ) семантичної мережі може використовуватися квадратна матриця (рис. 1.12) такого вигляду,

де використані скорочення: ПВВ - пристрій вводу-виводу; ПУ - пристрій управління.

	1. Комп'ютер	2. Процесор	3. Пам'ять	4. ПВВ	5. ПУ
1. Комп'ютер		Має	Має	Має	Має
2. Процесор			Записує дані		
3. Пам'ять		Зчитує дані		Зчитує дані	
4. ПВВ			Записує дані		
5. ПУ		Управляє	Управляє	Управляє	

Рис. 1.12. Матриця для опису семантичної мережі

На перетині i -го рядка і j -го стовпчика цієї матриці записується ім'я відповідного ставлення, якщо з i -го вузла в j -й вузол виходить лінія (зв'язок). У нашому прикладі (не рахуючи виділеного об'єкта «HP Compaq px9010») ми маємо п'ять рядків і п'ять стовпців (за числом понять — вузлів мережі). Загальна кількість не порожніх клітин матриці дорівнює 11, тобто числу відмічених зв'язків мережі.

Кожен вузол мережі може володіти власною структурою, тобто може бути представлений у вигляді власної семантичної мережі, яка розкриває суть даного поняття. Наприклад, поняття «Процесор» може бути визначене як безліч взаємодіючих між собою компонент «Суматор», «Регістри операндів», «Регістри зсуву» і т.д. В цьому випадку можна говорити про ієрархічну семантичну мережу, яка передбачає розбиття розглянутої мережі на ряд підмереж.

Переваги семантичних мереж:

- наочність (не випадковим є виникнення терміну «семантична візуалізація» інформації);
- багатство засобів відображення різних відношень між базовими поняттями розглянутої предметної області;
- можливість створення правил бази знань.

Порядок виконання роботи

1. Вивчити теоретичну частину за наведеними вище даними і додатковою літературою;
2. Отримати у викладача варіант завдання для виконання;
3. Побудувати семантичну модель заданого об'єкта;
4. Реалізувати програму з використанням семантичної моделі.

Варіанти завдань

Використовуючи відповідні дуги побудувати семантичну мережу, що стосується:

1. географії якого-небудь регіону. Дуги: держава, країна, континент, широта.
2. діагностики очних захворювань. Дуги: категорії хвороб, патолофізіологічний стан, спостереження, симптоми.
3. розпізнавання хімічних структур. Дуги: формула речовини, властивості речовини, область застосування, запобіжні заходи.
4. процедури пошуку корисних копалин. Дуги: найменування корисної копалини, розташування родовища, глибина залягання, методи видобутку.
5. судової процедури. Дуги: юридична особа, подія, заходи впливу, способи розслідування.
6. розподілу продуктів по магазинах. Дуги: джерело постачання, найменування продукту, спосіб транспортування, кінцевий пункт транспортування.
7. визначення приналежності тварини до певного виду, типу, сімейства. Дуги: місце проживання, будова, особливості поведінки, вид харчування.
8. класифікації харчових продуктів. Дуги: найменування продукту, складові частини, спосіб приготування, термін зберігання.
9. розпізнавання типу комп'ютера. Дуги: країна виробник, стандартна конфігурація, область застосування, використане програмне забезпечення.
10. ієрархічної структури бази даних. Дуги: система, стан, призначення, взаємодія складових.

Приклад виконання (варіант 9)

Використовуючи відповідні дуги побудувати семантичну мережу, що стосується розпізнавання типу комп'ютера. Дуги: країна виробник, стандартна конфігурація, область застосування, використане програмне забезпечення.

Розроблена семантична мережа:

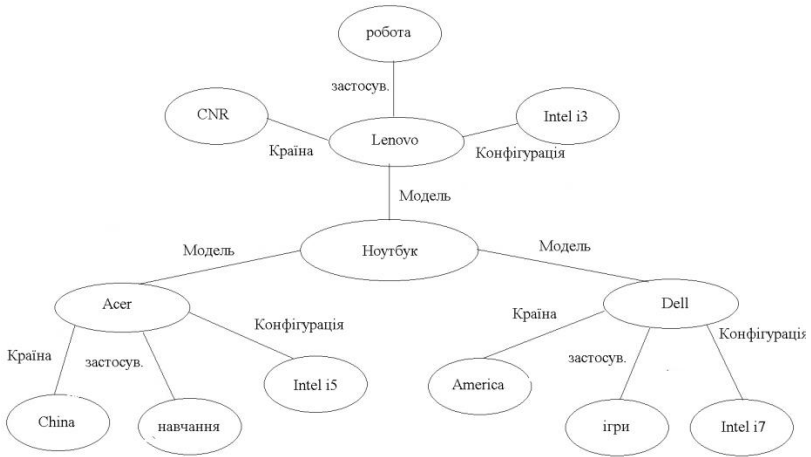


Рис. 1.13. Семантична мережа

Програма в MatLab:

```
>> SN=SNnew;
>> SN=SNaddORnode(SN,'China','America','CNR',
'education','games','work','Intel i5','Intel i7','Intel i3')
SN = node: {'China' 'America' 'CNR' 'education'
'games' 'work' 'Intel i5' 'Intel i7' 'Intel i3'}
relation: {9x9 cell}
nodetype: [1 1 1 1 1 1 1 1 1]
>> SN=SNaddANDnode(SN,'Model','Країна',
'застосув.','кофігурація','Acer','Dell','Lenovo')
SN = node: {1x16 cell}
relation: {16x16 cell}
nodetype: [1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0]
>> SN=SNaddrelation(SN,'Model','has','Країна');
>> SN=SNaddrelation(SN,'Model','has','застосув. ');
>> SN=SNaddrelation(SN,'Model','has','кофігурація');
>> SN=SNaddrelation(SN,'Країна','type','China');
>> SN=SNaddrelation(SN,'Країна','type','America');
>> SN=SNaddrelation(SN,'Країна','type','CNR');
>> SN=SNaddrelation(SN,'застосув.','type','education');
>> SN=SNaddrelation(SN,'застосув.','type','games');
>> SN=SNaddrelation(SN,'застосув.','type','work');
>> SN=SNaddrelation(SN,'кофігурація','type','Intel i5');
>> SN=SNaddrelation(SN,'кофігурація','type','Intel i7');
>> SN=SNaddrelation(SN,'кофігурація','type','Intel i3');
>> SN=SNaddrelation(SN,'China','include','Acer');
>> SN=SNaddrelation(SN,'America','include','Dell');
>> SN=SNaddrelation(SN,'CNR','include','Lenovo');
>> SN=SNaddrelation(SN,'education','include','Acer');
>> SN=SNaddrelation(SN,'games','include','Dell');
>> SN=SNaddrelation(SN,'work','include','Lenovo');
>> SN=SNaddrelation(SN,'Intel i5','include','Acer');
>> SN=SNaddrelation(SN,'Intel i7','include','Dell');
>> SN=SNaddrelation(SN,'Intel i3','include','Lenovo');
>> SNplot(SN, 'hierarchy');
>> SNplot(SN, 'circle');
```

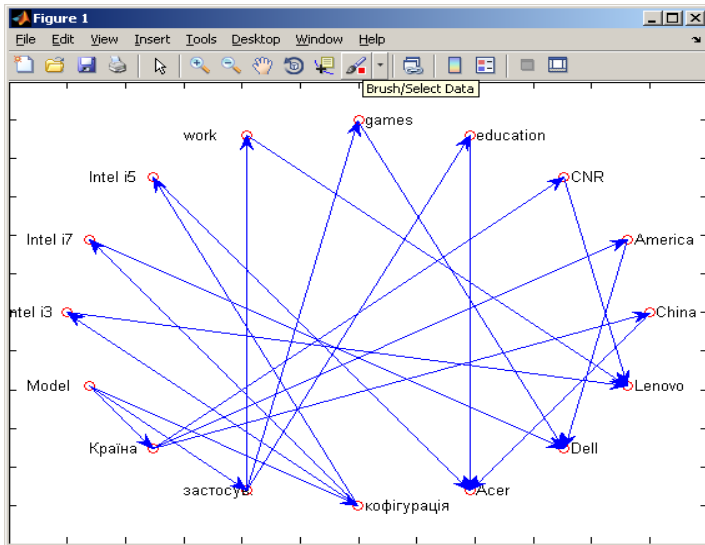


Рис. 1.14. Семантична мережа кругової структури виконана в MatLab

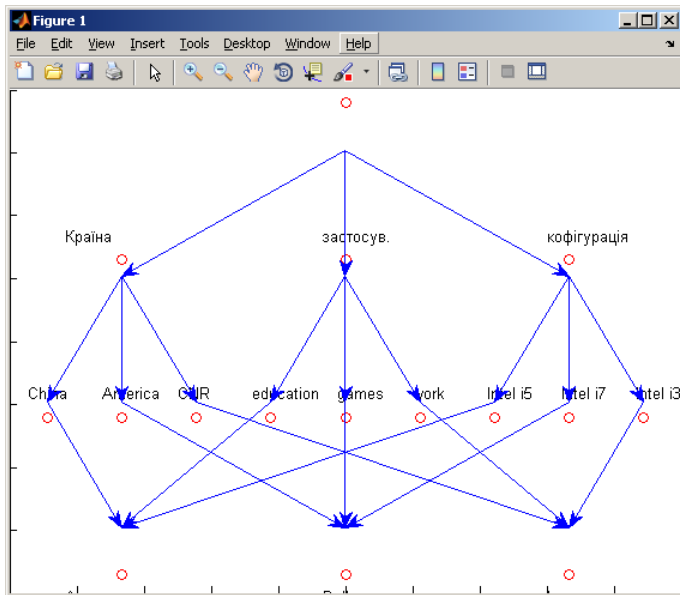


Рис. 1.15. Семантична мережа ієрархічної структури виконана в MatLab

Висновки до роботи:

Виконуючи дану лабораторну роботу ми ознайомилися з семантичними моделями, навчилися використовувати їх для подання знань в інтелектуальних системах управління. Відповідно до завдання, була розроблена семантична модель розпізнавання типу комп'ютера, яка також була відтворена у додатку Matlab SNTtoolbox.

Контрольні питання

1. Що таке семантична мережа і для чого її застосовують?
2. В чому полягає ідея створення семантичної мережі?
3. Яким чином представляються дані в семантичній мережі?
4. Чи існують обмеження на число зв'язків елементів, властивостей і складність при побудові семантичної мережі?
5. Які стосунки запропоновані в якості операторів відносини для групування вершин?

Лабораторна робота № 2

Використання фреймів для формування бази знань в інтелектуальних системах управління

Мета роботи

Навчитися використовувати фрейми для подання знань в інтелектуальних системах управління.

Теоретична частина

Фреймові моделі були вперше запропоновані одним з патріархів штучного інтелекту, професором Массачусетського технологічного інституту (США) Марвіном Мінскі.

Під фреймом {frame \rightarrow «рама», «основа», «каркас»} розуміється деякий мінімальний опис об'єкта, тобто набір таких його ознак (атрибутів), без будь-якого з яких описати цей об'єкт неможливо.

У загальному випадку фрейм може бути представлений у вигляді структури, показаної на рис. 2.1. Опис фрейму тут проводиться наступним чином: $\langle Z, (Y_1, T_1, A_1), \dots, (Y_n, T_n, A_n) \rangle$, де Z - ім'я фрейму; Y_i - ім'я слота; T_i - значення слота; A_i - ім'я приєднаної процедури (необов'язковий параметр).

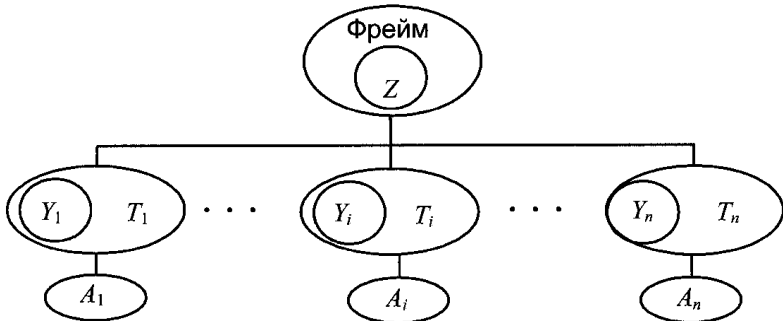


Рис. 2.1. Структура фрейму

Слоти — це деякі незаповнені структурні елементи фрейму, заповнення яких призводить до того, що даний фрейм ставиться

у відповідність ситуації, що розглядається (явищу, об'єкту). Фрейм з незаповненими слотами називається фреймом-прототипом (або протофреймом). Фрейм із заповненими слотами називається фреймом-прикладом (або фреймом-екземпляром). В якості даних фрейм може містити звернення до процедур (так звані приєднані процедури).

Існують різні способи заповнення значень слотів у фреймі-екземплярі:

- за замовчуванням від фрейму-прототипу, що зберігається в базі знань;

- через успадкування властивостей фрейму;

- через приєднану процедуру;

- явно з діалогу з користувачем;

- з бази даних.

На рис. 2.2 наведено приклад фрейму «Персональний комп'ютер».

Ім'я фрейма	Ім'я слота	Значення слота
Персональний комп'ютер <i>HP Compaq n x 9010</i>	Процесор	<i>Intel Pentium 4 2,66 ГГц</i>
	Монітор	<i>TFT 15,1» XGA</i>
	Оперативна пам'ять	256 МБ
	Жорсткий диск	40 ГБ
	Модем	56 К (V.92)
	Операційна система	<i>Microsoft Windows XP Home</i>

Рис. 2.2. Фрейм «Персональний комп'ютер»

Важливою особливістю фреймів є їх вкладеність, тобто значення слота, який, у свою чергу, може бути набором слотів нижчого рівня тощо. («Принцип матрешки»). Подібно до семантичних мереж, каркасні моделі подання знань дозволяють описати найважливіші зв'язки між атрибутами об'єктів і реалізувати швидкий висновок на основі принципу успадкування. Таким чином, рамкові моделі виступають

потужним і ефективним засобом організації знань при збереженні інформації про структуру досліджуваних об'єктів.

Фреймворки є одним із поширених формалізмів для представлення знань в експертних системах. Фрейм можна уявити як конструкцію, що складається з набору комірок - прорізів. Кожен слот складається з імені та пов'язаних значень. Значеннями можуть бути дані, процедури, посилання на інші кадри або порожні. Ця конструкція дуже зручна для моделювання аналогій, опису областей із загальними зв'язками понять тощо.

Будь-який кадр складається з кількох компонентів, назви та вміст яких описані нижче:

1. Назва фрейму. Це ідентифікатор, призначений кадру. Фреймворк повинен мати унікальне ім'я в даній системі фреймворків.

2. Назва слота. Це ідентифікатор, призначений слоту. Слот повинен мати унікальне ім'я в межах кадру, до якого він належить. Зазвичай назва слота не має семантичного значення і є лише ідентифікатором цього слота.

3. Показчики успадкування. Ці показчики стосуються тільки фреймових систем ієрархічного типу, засновані на відносинах «абстрактне – конкретне» вони показують, яку інформацію про атрибути слотів у фреймі верхнього рівня успадковують слоти з такими ж іменами у фреймі нижнього рівня. Типові показчики успадкування: Unique (U: - унікальний), Same (S: такий же), Range (R: встановлення кордонів), Override (O: ігнорувати) і т.п. U показує, що фрейм може мати слоти з різними значеннями: S — всі слоти повинні мати однакові значення, R — значення слотів фрейму нижнього рівня повинні знаходитися в межах, зазначених значеннями слотів фрейму верхнього рівня, O — при відсутності вказівки значення слота фрейму верхнього рівня стає значенням слота фрейму нижнього рівня, але в разі визначення нового значення слотів фреймів нижніх рівнів вказуються як значення слотів.

4. Тип даних. Тут вказується, що слот має чисельне значення, або служить показчиком іншого фрейму. До типів даних відносяться: FRAME (показчик), INTEGER (цілий), REAL (дійсний), BOOL (булевий), LISP (приєднана процедура), TEXT

(текст), LIST (список), TABLE (таблиця), EXPRESSION (вираз) та ін.

5. Значення слота. Пункт вводу значення слота. Значення слота повинне збігатися із зазначеним типом даних цього слота, крім того повинна виконуватися умова успадкування.

6. Демон. Тут дається визначення демонів типу IF-NEEDED, IF-ADDED, IF-REMOVED і т.д. Демоном називається процедура, що автоматично запускається при виконанні деякої умови. Демони запускаються при зверненні до відповідного слоту. Крім того, демон є різновидом приєднаної процедури.

7. Приєднана процедура. Як значення слоту можна використовувати програму процедурного типу. Коли ми говоримо, що в моделях представлення знань фреймами об'єднуються процедурні та декларативні знання, то вважаємо демони і приєднані процедури процедурними знаннями.

Особливістю ієрархічної структури є те, що інформація про атрибути фрейму на верхньому рівні спільно використовується всіма фреймами нижніх рівнів, пов'язаних з ним.

Порядок виконання роботи

1. Вивчити теоретичну частину за наведеними вище даними та додаткової літератури.
2. Отримати у викладача варіант завдання для виконання.
3. Побудувати фреймову модель заданого об'єкта;
4. Реалізувати програму з використанням фреймової моделі.

Варіанти завдань

Використовуючи фреймову модель подання знань реалізувати структуру відносин, що описують такі ситуації:

1. екзамен з дисципліни за семестр у викладача при складових: семестр, екзамен, викладач, оцінка, студент, отримувати.
2. відомість при складових: дисципліна, студент, іспит, семестр, викладач, оцінка.
3. конференція з комерційних питань при складових: дата, місце проведення, тема, мета, доповідачі.

4. отримання оцінки при складових: викладач, студент, оцінка, отримувати.

5. використання виробу при складових: організація, розробка технологічного рішення, дослідження «фізичного ефекту», методи створення виробу.

6. інформаційна структура бази даних в приладобудуванні при складових: фізичні ефекти, технічні рішення, виробу, об'єкт поставки виробу, стенди, нормативи.

7. класифікація продукту при складових: назва, область застосування, спосіб зберігання, спосіб транспортування.

8. аудиторія (опис) при складових: місткість, призначення, що складають, місцезнаходження.

9. тваринний світ при складових: вид, тип, місце існування, особливості поведінки.

Контрольні питання

1. Що представляє собою фрейм, його складові частини?

2. Що таке слот і з яких частин він складається?

3. Для чого служать ім'я фрейму та ім'я слота?

4. Для чого служать покажчики успадкування?

5. Для чого служать вказівка типу даних, демон?

6. Для чого служать приєднана процедура і значення слота?