УДК 004.89

**КОВАЛЕНКО А.Д.,** *Миколаївський державний гуманітарний університет ім. П. Могили*

**Коваленко Андрій Дмитрович** – аспірант факультету комп'ютерних наук Миколаївського державного гуманітарного університету ім. П. Могили.

# ВИКОРИСТАННЯ АГЕНТНИХ ТЕХНОЛОГІЙ ДЛЯ ПРЕДСТАВЛЕННЯ ЗНАНЬ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ

*У статті розглядаються методи представлення знань в мультиагентних інтелектуальних системах. Наводяться фреймовий та онтологічний підходи до представлення знань. Для створення онтологій предметних галузей використовується підхід METHONTOLOGY. Також розглядається архітектура багаторівневої мультагентної системи і агентів як керуючих та координуючих сутностей в процесі представлення та обробки знань. Також висвітлені проблеми створення подібних систем з точки зору розподіленої компонентної об'єктної моделі (COM+). Наведений приклад створення онтології предметної галузі – системи реального часу.*

*In the article the methods of representation of knowledges are examined in mul'tiagentnikh intellectual systems. The frame and ontological going is pointed near representation of knowledges. For creation of ontology of subject industries taken approach METHONTOLOGY. Architecture of the multilevel mul'tagentnoy system and agents is also examined as managing and co-ordinating essences in the process of presentation and treatment of knowledges. The problems of creation of the similar systems are also lighted up from the point of view the up-diffused component objective model (Com+). The example of creation of ontology of subject industry - real-time systems is resulted.*

This arcticle investigates the methods of knowledge representing in multiagent intelligent systems. There are knowledge representing examples as frame and ontological approaches. The METHONTOLOGY idea using for creation ontologies of data domains. The article also reviews an architecture of multiagent multilevel system, control and dispatch agents as entities in knowledge representation and processing. The system creation problems describes from the point of view of distributed component object model (COM+). As example of creation ontology for real-time system data domain.

## Вступ

В галузі експертних систем представлення знань означає не що інше, як систематизовану методику опису на машинному рівні того, що знає людина – експерт, що спеціалізується в конкретній предметній галузі. Вважається помилковою думка, що представлення знань зводиться до кодування аналогічному до шифрування. Якщо закодувати повідомлення, підставив деяким регулярним чином замість одних символів інші, то отриманий результат не має нічого спільного з представлення змісту повідомлення в тому сенсі, як це розуміється в теорії штучного інтелекту, навіть якщо отриманий код легко сприймається на машинному рівні та його можна зберігати в пам'яті компьютера.

*Представлення* (representation) в роботі Уінстона [*Winston,1984*] визначається як «множина синтаксичних і семантичних узгоджень, яка робить можливим опис предмета». В штучному інтелекті під «предметом» розуміється стан у деякій проблемній галузі, наприклад об'єкти в цій галузі, їх властивості, співвідношення, які існують між об'єктами.

*Опис* (description) «дозволяє використати узгодження з представлення для опису віизначених предметів» [Winston, 1992].

Синтаксис представлення специфікує набір правил[1], регламентуючих об'єднання символів для формування виразів мовою представлення. Можна стверджувати те, що вираз *добре чи погано сформовано*, тобто те, наскільки воно відповідає цим правилам. Сенс повинні мати тільки добре сформовані вирази.

## Особливості фреймового представлення знань

У системі фреймів є спроба судити про клас об'єктів, використовуючи представлення знань про прототипи, що добре представляють більшість різновидів об'єктів різних класів конкретної предметної галузі, але повинні бути якимось чином скоректовані, для того щоб представити всю складність, властивої реальному світу.

Власне кажучи, фрейм виявився тим засобом, який допоміг зв'язати декларативні і процедурні знання про деяку сутність в структуру записів, що складається зі *слотів* і *наповнювачів* (filler)[2]. Слоти грають ту ж роль, що і поля в записі, а наповнювачі – це значення, що зберігаються в полях. Фрейм також можна розглядати як складний вузол в особливого виду асоціативної мережі. Як правило, фрейми організовані у вигляді «ослабленої ієрархії», у якій фрейми, що розташовуються нижче мережі, можуть успадковувати значення слотвв *різних* фреймів, розташованих вище.

Фундаментальна ідея полягає в тому, що властивості і процедури, асоційовані з фреймами у вигляді властивостей вузлів, розташованих вище в системі фреймів, є більш-менш фіксованими, оскільки вони представляють ті речі чи поняття, що у більшості випадків є адекватними для нас сутностями, у той час як фрейми більш нижніх рівнів мають слоти, що повинні бути заповнені найбільш динамічною інформацією, що піддається частим змінам. Якщо такого роду динамічна інформація є відсутньою через неповноту наших знань про найбільш ймовірний стан справ, то слоти фреймів більш нижніх рівнів заповнюються даними, успадкованими від фреймів більш верхніх рівнів, що носять глобальний характер. Дані, що передаються в процесі функціонування системи від сторонніх джерел знань у фрейми нижніх рівнів, мають більш високий пріоритет, ніж дані, успадковані від фреймів більш верхніх рівнів[3].

На рис. 1. зображено ієрархію категорій фреймів для пристроїв типу "*датчик*".

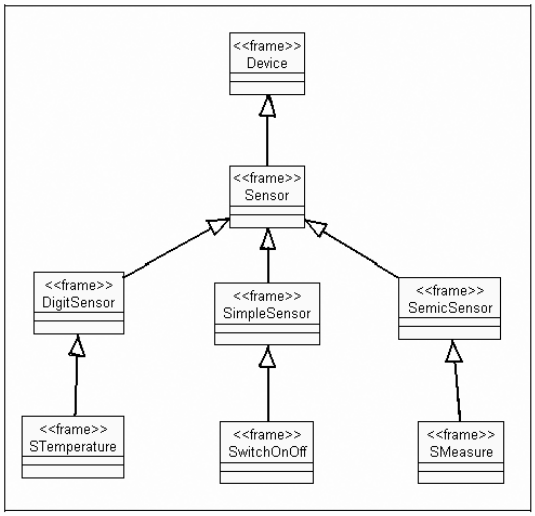


**Рис. 1. Ієрархія категорій фреймів для представлення сенсорних пристроїв**

У більшості останніх досліджень, які стосуються представлення знань, перевага віддається фреймам. Такий підхід дає цілком задовільні відповіді на питання ефективного формування баз знань, а також їхньої наступної обробки для прийняття рішень. Семантика вузлів і зв'язків чітко прослідкується завдяки поділу вузлів на *вузли-типу* і *вузли-лексеми* й обмеженню кількості зв'язків. Ефективність обробки забезпечується підключенням до вузлів специфічних процедур, на які покладається обчислення значень перемінних у відповідь на запити чи при відновленні значень інших властивостей вузла.

Використання фреймів як основної структури даних, що зберігає інформацію про типові об'єкти і події, у даний час широко застосовується в практиці створення компонентів з підтримкою штучного інтелекту. Більшість програмних інструментальних засобів, призначених для побудови експертних систем, забезпечує тим чи іншим засобом організацію бази знань на основі фреймів. У багатьох випадках бажано оцінити, якими можливостями володіє механізм представлення гіпотез за допомогою фреймів у частині використання таких даних, як сукупність симптомів чи результатів спостережень за поведінкою об'єктів. Порівняння цих даних з інформацією, що зберігається в слотах фреймів, дає свідчення на користь гіпотез, представлених фреймом, а також дозволяє формулювати визначені припущення щодо інших даних, наприклад припустити існування додаткових симптомів, присутність чи відсутність яких зможе підтвердити (чи спростувати) аналізовану гіпотезу.

Природно, для того щоб реалізувати систему фреймів у вигляді, придатному для роботи з кінцевим користувачем, потрібно розробити програмну оболонку і засоби інтерфейсу користувача. Хоча до слотів окремих фреймів і можуть бути підключені спеціальні процедури, ці локальні модулі не здатні взяти на себе всі проблеми організації обчислювального процесу в системі. Необхідно мати в тій чи іншій формі спеціальний інтерпретатор, що буде формувати й обробляти запити і приймати рішення, при яких умовах можна вважати досягнутою мету, сформульовану в запиті. Тому найчастіше фрейми використовуються в сполученні з іншими засобами представлення знань, зокрема в сполученні з породжуючими правилами [Yen et al., 1991,a],[Yen et al., 1991, b].

## Онтологічний аналіз представлення знань

Поняття *онтології* припускає визначення і використання взаємозалежної і взаємопогоджуваної сукупності трьох компонентів: таксономії термінів, визначень термінів і правил їхньої обробки. З огляду на це, наведемо наступне визначення поняття моделі онтології. *Онтологією* [Gruber, 1993] називається експліцитна специфікація концептуалізації. Формально онтологія складається з термінів, організованих у таксономію, їх визначень і атрибутів, а також зв'язаних з ними аксіом і правил висновку[4].

Під формальною моделлю онтології **O** будемо розуміти впорядковану трійку виду:

$$O = < X, \Re, \Phi >, \qquad (1)$$

де **X** — кінцева безліч концептів (понять, термінів) предметної області, що представляє онтологія **O**;

$\Re$ — кінцева безліч відносин між концептами (поняттями, термінами) заданої предметної області;

**Ф** — кінцева безліч функцій інтерпретації (аксіоматизація), заданих на концептах і/чи відносинах онтології **O.**

Помітимо, що природним обмеженням, що накладається на безліч **X**, є його кінцевість і непорожнеча. Інакше обстоїть справа з компонентами **Ф** и $\Re$ в визначенні онтології **О.** Зрозуміло, що й у цьому випадку **Ф** и $\Re$ повинні бути кінцевими безлічами.

Розглянемо, однак, граничні випадки, зв'язані з їх порожнечею. Нехай $\Re = \varnothing$ і $\Phi = \varnothing$. Тоді онтологія **О** трансформується в простий словник:

$$\mathbf{O} = \mathbf{V} = < X, \Re, \Phi > \qquad (2)$$

Така онтологія може бути корисна для специфікації, поповнення і підтримки словників програмного забезпечення, але онтології-словники мають обмежене використання, оскільки не вводять експліцитного змісту термінів. Хоча в деяких випадках, коли використовувані терміни належать дуже вузькому (наприклад, технічному) словнику і їхні змісти вже заздалегідь добре погоджені в межах визначеного (наприклад, наукового) співтовариства, такі онтології застосовуються на практиці. Відомими прикладами онтологій цього типу є індекси машин пошуку інформації в мережі Інтернет.

Результати аналізу окремих випадків моделі онтології приведені в таблиці 1.

**Таблиця 1. Класифікація моделей онтології**

| **Компоненти моделі** | $\Re = \varnothing$ $\Phi = \varnothing$ | $\Re = \varnothing$ $\Phi \neq \varnothing$ | $\Re = \varnothing$ $\Phi \neq \varnothing$ | $\Re = \{is\_a\}$ $\Phi = \varnothing$ |
|---|---|---|---|---|
| **Формальне визначення** | $< X, \{\}, \{\} >$ | $< X_1 \cup X_2, \{\}, \Phi >$ | $< X_1 < X_1 \cup (X_2 * \cup \rho_2 *), \{\}, \Phi >$ | $< X_1, is\_a\}, \{\} >$ |
| **Пояснення** | Словник ПЗ[i] | Пасивний словник ПЗ | Активний словник ПЗ | Таксономія понять ПЗ |

Під формальною моделлю онтологічної системи $\sum^{O}$ будемо розуміти триплет виду:

$$\sum^{O} = < O^{meta}, \{O^{d\&t}\}, \Xi^{\inf} >, \qquad (3)$$

де

$O^{meta}$ – онтологія верхнього рівня (метаонтологія);

$\{O^{d\&t}\}$ – безліч предметних онтологій і

$\Xi^{\inf}$ – модель машини висновку, асоційованої з $\sum^{O}$.

Використання системи онтологій і спеціальної машини висновку дозволяє вирішувати в такій моделі різні задачі. Розширюючи систему моделей $\{O^{d\&t}\}$, можна враховувати переваги користувача, а змінюючи модель машини висновку, вводити спеціалізовані критерії релевантності одержуваної в процесі пошуку інформації і формувати спеціальні репозіторії накопичених даних, а також поповнювати при необхідності використовувані онтології.

В моделі $\sum^{O}$ маються три онтологічні компоненти:

- метаонтологія;
- предметна онтологія;
- онтологія задач.

Як було вказане вище, метаонтологія оперує загальними концептами і відносинами, що не залежать від конкретної предметної області. Концептами метарівня є загальні поняття, такі як «об'єкт», «властивість», «значення» і т.д. Тоді на рівні метаонтології ми одержуємо інтенсіональний опис властивостей предметної онтології й онтології задач. Онтологія метарівня є статичною, що дає можливість забезпечити тут ефективний висновок.

Предметна онтологія $O^{domain}$ містить поняття[6], що описують конкретну предметну область, відносини, семантично значимі для даної предметної області, і безліч інтерпретацій цих понять і відносин (декларативних і процедурних). Поняття предметної області специфічні в кожній прикладній онтології, але відносини — більш універсальні. Тому як базис звичайно виділяють такі відносини моделі предметної онтології, як *part_of*, *kind_of*, *contained__in*, *member_of*, *see_also* і деякі інші.

## Проблемно-орієнтовані архітектури агентних систем

Розвиток мережі Інтернет призвів до появи значної кількості розрізнених джерел і сховищ інформації (баз даних і знань, окремих інформаційних ресурсів і т.п.), що характеризуються різними способами представлення, форматами і мовами опису інформації. В результаті виникла необхідність пошуку/формування, обробки і передачі/транспортування адекватних знань з розподілених джерел користувачам, вчасно, в потрібному контексті для рішення актуальних задач.

Для ефективної обробки онтологічних баз знань[7] необхідно обрати потрібну архітектуру мультиагентної системи і мати на увазі два її аспекти:

- архітектуру, що підтримує методи взаємодії агентів у процесі функціонування системи в цілому, і
- архітектуру окремого агента.

Основне призначення компонента архітектури взаємодії системи агентів полягає в тому, щоб забезпечити скоординовану поведінку агентів при вирішенні загальної чи власної задач. Тут можна виділити два основних варіанти архітектур. В одній з них агенти не утворюють ієрархії і вирішують загальну задачу цілком у розподіленому варіанті (*однорівнева архітектура взаємодії агентів*). В іншому варіанті координація розподіленого функціонування тією чи іншою мірою підтримується спеціально виділеним агентом, що при цьому відноситься до мета-рівня стосовно інших агентів (*ієрархічна архітектура взаємодії агентів*).

Класифікація архітектур агентів ґрунтується на парадигмі, що лежить в основі прийнятої архітектури. За цією ознакою розрізняють два основних класи архітектур[5]:

- архітектура, що базується на принципах і методах штучного інтелекту, тобто систем, заснованих на знаннях («deliberative agent architecture», «архітектура розумного агента»)[8];
- архітектура, заснована на поведінці (reactive architecture) чи «реактивна архітектура» (заснована на реакції системи на події зовнішнього світу).

Насправді серед сучасних розроблених архітектур не існує таких, про які можна було б виразно сказати, що вона є чисто поведінкової чи заснована тільки на знаннях. Кожна з розроблених архітектур є, по суті, гібридною, маючи ті чи інші риси від архітектур обох типів.

З іншого боку, незалежно від формалізації парадигми, архітектури агентів класифікуються в залежності від виду структури, накладеної на функціональні компоненти агента, і прийнятих методів організації взаємодії його компонент в процесі роботи.

Протягом розвитку мультиагентні системи стають більш складними, тому потрібно розбивати їхні складні рівні на множину певних рівнів. В результаті створюються багаторівневі чи n-рівневі архітектури. Хоча фундаментальна взаємодія агентів з користувачем може бути в цьому випадку однаковою (представлення, формування і збереження знань), іноді все-таки краще розбивати більш складні рівні на множини рівнів.

На рис. 2 представлена n-рівнева архітектура[9] мультиагентної системи обробки онтологічних баз знань.

У даній статті представлена багаторівнева агентна модель для представлення знань на основі онтологій предметної галузі.

## Рівні мультиагентної системи і розподілення ролей агентів

Отже, розглянута нами мультиагентна система є гібридною, тому що в ній частково об'єднані деліберативна і реактивна архітектури, а також є багаторівневою, що забезпечує визначену гнучкість при розробці та зручну настройку при налагодженні й інтеграції. Однак, багаторівневі системи дуже складні в проектуванні й у побудові, а також у наступній інтеграції і тестуванні. Складність у проектуванні, у першу чергу, представляє самий нижній рівень – керування транзакціями, їх синхронізація і динамічне балансування навантаження[9], що дозволяє у деяких випадках збільшити швидкість і продуктивність системи. На рис. 2 стрілки означають різні за призначеннями транзакції, до того ж наявність транзакції припускає, що мультиагентна система може бути також і розподіленою, тому вирішення проблеми динамічного балансування навантаження її підсистем (рівнів) є однією з головних задач. У таблиці 2 приведена інформація щодо ролей агентів, представлених на рис. 2.

**Таблиця 2. Ролі агентів у системі представлення знань**

| Агент | Роль | Результат |
|---|---|---|
| DataMiner | Аналіз предметної галузі та формування словника простих понять (головні абстракції, сутності тощо). | Словник простих понять зберігається в базі даних. |
| DatabaseService | Сервісні функції - логічна та фізична організація даних. | Впорядкування даних, організація таблиць, зв'язків між ними, запитів тощо. |
| ConceptSelector | Формування списку концептів з бази за визначеним сценарієм задачі. | Логічна організація концептів для подальшого формування асоціативної мережі. |
| RelationshipClassificator | Визначення потрібних зв'язків між концептами. | Створення асоціативної мережі концептів та зв'язків між ними. |
| RelationshipSelector | Онтологічне представлення знань. | Формування та організація онтологічної бази знань. |
| RelationshipInterpretator | Аксіоматизація функцій, інтерпретація онтологій. | Формування мети чи гіпотези. |
| TransactController | Керування транзакціями | Синхронізація потоків |
| Dispatcher | Керування рівнями системи | Динамічне балансування навантаження. |

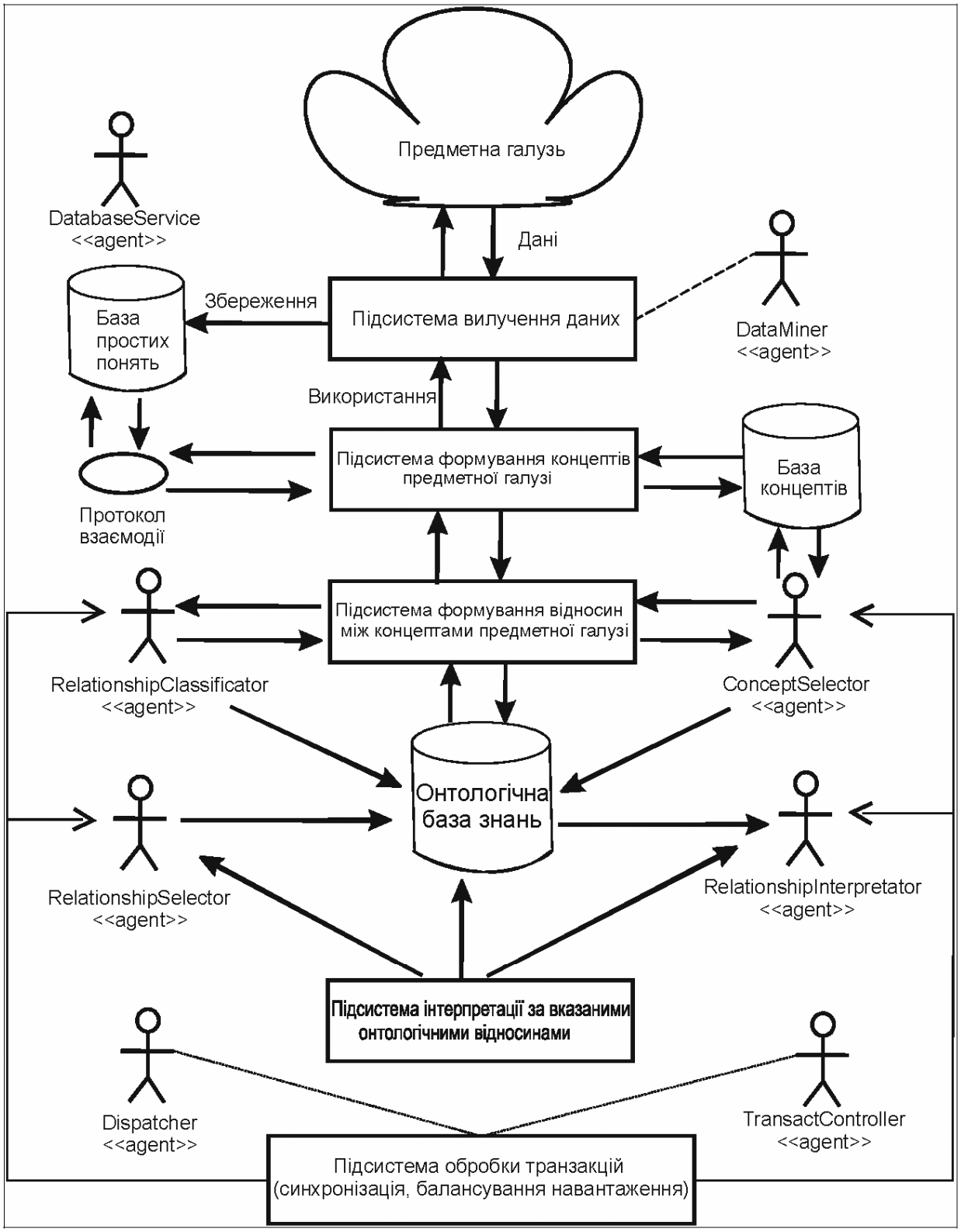


**Рис. 2. Багаторівнева мультиагентна система представлення та обробці знань.**

## Представлення знань

Відповідно до представлення знань приведемо основні етапи формування онтології предметної галузі[10] - охоронна система приміщення (працює в реальному часі). Помітимо, що процес побудови онтології розпадається на серію підпроцесів (етапів, в системі - рівнів) по створенню проміжних представлень. При цьому виконання окремих етапів не послідовне, а визначається повнотою і точністю вже накопичених знань. Отже, як показує досвід [Benjamins et al., 1998], спочатку будується *словник термінів* (глосарій), потім *дерева класифікації концептів* і *діаграми зв'язків між концептами*. І тільки після цього – інші проміжні представлення. Процес виконання вищезазначених етапів наступний[11]:

- *побудова словника термінів*. В даному випадку визначаємо головні абстракції системи, а також головні та другорядні сутності, які забезпечують роботу системи. Для обраної предметної галузі (охоронна система приміщення) сформуємо словник основних термінів (табл. 3).

**Таблиця 3. Фрагмент словника термінів**

| Термін | Опис терміна |
|---|---|
| Сигнал | "Послідовність імпульсів визначеної частоти та через певні інтервали часу" |
| Пристрій | "Система взаємодіючих електронних компонентів виконуючих задану функцію або функції" |
| Датчик | "Пристрій контролю незмінності певного стану" |
| Контроллер | "Пристрій керування датчиками" |
| Сирена | "Пристрій, що генерує звукові хвилі заданої потужності та частоти" |
| Таймер | "Пристрій, що фіксує зміну часу за певним законом" |

- *побудова дерева класифікації концептів*. Виконується тоді, коли глосарій термінів досягає "істотного" обсягу. При цьому використовуються відношення *part_of*, *subclass_of*, *associate_with*, *relation_of*, *member_of* і т.д. На рис. 3 показане дерево класифікації концептів (таксономій).


**Рис. 3. Фрагменти таксономій**

- *побудова діаграм бінарних відносин.* Мета створення цих діаграм (див. рис. 4) – фіксація відносин між концептами однієї чи різних онтологій. Помітимо, що надалі ці діаграми можуть послужити вихідним матеріалом для інтеграції різних онтологій[12].

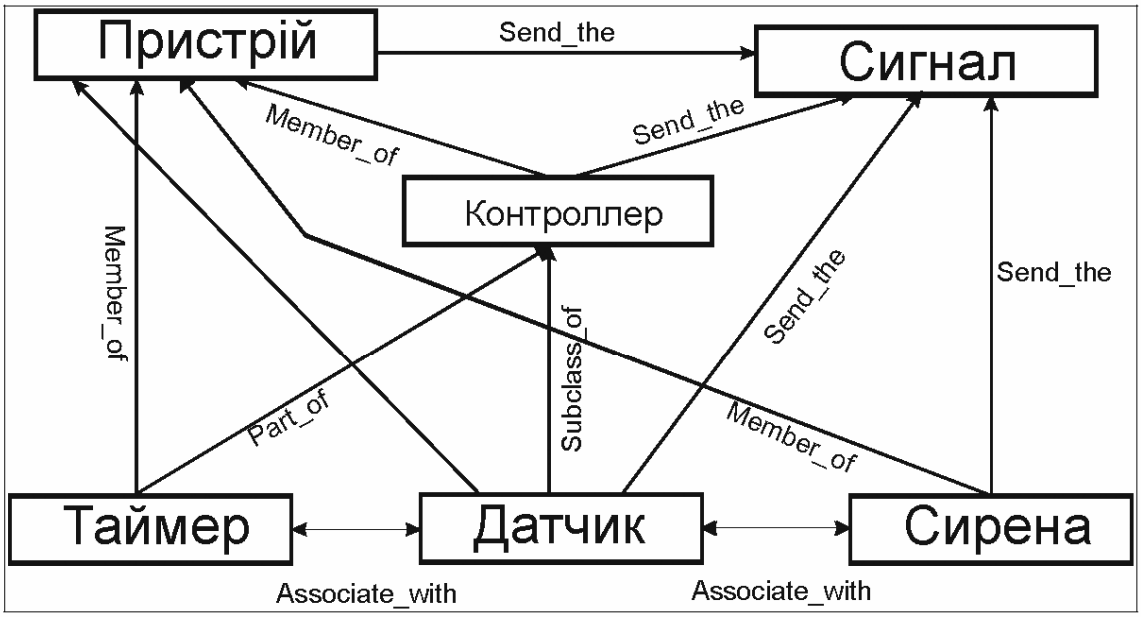


**Рис. 4. Фрагмент діаграми бінарних відносин**

Після побудови представлень, фіксованих вище, для кожного дерева класифікації концептів будуються:

1. ***Словник концептів***, що містить усі концепти предметної області, екземпляри таких концептів, атрибути екземплярів концептів, відносини, джерелом яких є концепт, а також синоніми та акроніми концепта.
2. ***Таблиця бінарних відношень*** для кожної “Ad hoc” відносини, вихідний концепт якої міститься в класифікаційному дереві. Для кожного відношення фіксується його ім'я, імена концепту-джерела і цільового концепту, інверсне відношення й інші характеристики.
3. ***Таблиця атрибутів екземпляра*** для кожного екземпляра зі словника концептів. Основні характеристики наступні: ім'я атрибута, тип значення, одиниця виміру, точність, діапазон зміни, значення «за замовчуванням», атрибути, що можуть бути виведені з використанням даного, формула чи правило для висновку атрибута й ін.
4. ***Таблиця атрибутів класу*** для кожного класу зі словника концептів з аналогічними характеристиками.
5. ***Таблиця логічних аксіом***, у якій даються визначення концептів через логічні вираження, які завжди є *true*. Визначення кожної аксіоми включає її ім'я, природно-мовний опис, концепт, до якого аксіома відноситься, атрибути, використані в аксіомі, логічний вираз, що формально описує аксіому та ін.
6. ***Таблиця констант***, де для кожної константи вказується її ім'я, природно-мовний опис, тип значення, саме значення, одиниця виміру, атрибути, що можуть бути виведені з використанням даної константи, і т.п.

7. ***Таблиця формули*** для кожної формули, включеної в таблицю атрибутів екземпляра. Кожна таблиця цього типу, крім власне формули, повинна специфікувати її ім'я, атрибут, виведений за допомогою цієї формули, природно-мовний опис, точність, обмеження, при яких можливо використовувати формулу та ін.
8. ***Дерева класифікації атрибутів***, що графічно показують відповідні атрибути і константи, використовувані для висновку значення початкового атрибута і формули, застосовувані для цього.
9. ***Таблиця екземплярів*** для кожного входу в словник концептів. Тут специфікується ім'я екземпляра, його атрибути і їхні значення.

## Висновки

Як показує аналіз приведених вище процедур, виконуваних при створенні онтологій у підході METHONTOLOGY[13], усі вони добре корелюють з тими стадіями, що виділені і використовуються при побудові баз знань. І цей не випадковий збіг, а закономірність, зв'язана з тим, що онтологія — це, власне кажучи, база знань спеціального виду. Тому, як і у випадку побудови баз знань, тут використовується концепція швидкого прототипування, а специфіка виявляється в тих конкретних процесах, що реалізують розглянуті вище процедури.

При цьому:

- планування виконується до початку розробки;
- контроль і гарантії якості здійснюються в процесі розробки;
- велика частина операцій по нагромадженню знань і їхній оцінці виконується на стадії концептуалізації для того, щоб запобігти поширенню помилок на фазі реалізації;
- інтеграція не повинна розглядатися на стадії реалізації. Вона виконується в процесі розробки.

## *Література*


1. Питер Джексон Введение в экспертные системы.: Пер. с англ. : Уч. пос.- М.: Издательский дом «Вильямс», 2001. – 624 с.
2. Джордж Ф. Люгер Искусственный интеллект и методы решения сложных проблем, 4-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003.– 864 с.
3. Батищев С.В., Лахин О.И., Минаков И.В., Ржевский Г.А., Скобелев П.О. Разработка инструментальной системы для создания мультиагентных приложений в сети Интернет // Известия Самарского научного центра РАН. 2001. №1.
4. Смирнов А.В., Пашкин М.П., Шилов Н.Г., Левашова Т.В. Онтологии в системах искусственного интеллекта: способы построения и организации. Новости искусственного интеллекта, 2002. № 1. Часть 1. С. 3–13. № 2. Часть 2. С. 3–9.
5. Agent projects. Europe's Network of Exellence for Agent-based Computing. http://www.agentlink.org/resources/-agentprojects-db.html, 2001.
6. Tom Gruber What is an ontology? http://www-ksl.stanford.edu/kst/what-is-an-ontology.html
7. WonderWeb - Ontology Infrastructure for the Semantic Web http://wonderweb.semanticweb.org.
8. Сервер з ресурсами: термінологія агентів, дослідження в галузі агентів, стандарти, публікації і т.д. www.agent.org
9. Ричард Лейнекер COM+. Энциклопедия программиста: Пер. с англ. – СПб.: ООО «ДиаСофтЮП», 2002. – 656 с.
10. Agent Communication with Multiple Ontologies, http://ai-www.aist-nara.ac.jp/doc/people/takeda/doc/icot-paper-v/icot-paper-v.html
11. The On-To-Knowledge project will develop methods and tools and employ the full power of the ontological approach to facilitate knowledge management. It is the home of the Ontology Inference Layer (OIL), ajoint standard for specifying and exchanging ontologies. www.ontoknowledge.org
12. Котенко И.В., Карсаев О.В., Самойлов В.В. Онтология предметной области обучения обнаружению вторжений в компьютерные сети // Международная конференция по мягким вычислениям и измерениям. SMC'2001. Сборник докладов. Том 1. СПб: СПбГЭТУ, 2002. С.255-258.
13. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем, – СПб.: Питер, 2001. – 384 с.