

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя  
(повне найменування вищого навчального закладу)  
Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(назва факультету)  
Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

**Магістра**

(освітній ступінь)

на тему: **Методи та засоби моделювання процесів управління та моніторингу у комп'ютерних мережах**

Виконав: студент (ка) 6 курсу, групи СІМ-61  
спеціальності 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

	(підпис)	<b>Кучма І.М.</b> (прізвище та ініціали)
Керівник	(підпис)	<b>Яцишин В.В.</b> (прізвище та ініціали)
Нормоконтроль	(підпис)	<b>Луцик Н.С.</b> (прізвище та ініціали)
Завідувач кафедри	(підпис)	<b>Осухівська Г.М.</b> (прізвище та ініціали)
Рецензент	(підпис)	<b>Гладько Ю.Б.</b> (прізвище та ініціали)

Тернопіль  
2023

Міністерство освіти і науки України  
 Тернопільський національний технічний університет імені Івана Пулюя  
 (повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
 Кафедра комп'ютерних систем та мереж

### ЗАТВЕРДЖУЮ

Завідувач кафедри Осухівська Г.М.

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр  
 (назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»  
 (шифр і назва спеціальності)

студенту Кучмі Івану Михайловичу  
 (прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) Методи та засоби моделювання процесів управління та моніторингу у комп'ютерних мережах

Керівник проекту (роботи) Яцишин Василь Володимирович, к.т.н., доц.  
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «01» грудня 2023 року №4/7-1132

2. Термін подання студентом завершеної роботи \_\_\_\_\_

3. Вихідні дані до роботи Принципи організації комп'ютерних мереж, модель OSI, моделі комп'ютерних мереж, сутності онтології комп'ютерних мереж

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз особливостей моделювання та створення онтологій при управлінні комп'ютерними мережами 2. Математична модель та архітектура онтології управління та моніторингу процесів в комп'ютерних мережах 3. Результати моделювання процесів управління комп'ютерними мережами 4. Охорона праці та безпека в надзвичайних ситуаціях. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Актуальність і мета дослідження. 2. Задачі дослідження, об'єкт і предмет, наукова новизна і практична цінність дослідження. 3. Класифікація онтологій. 4. Структура моделі онтології 5. Архітектура онтології комп'ютерних мереж. 6. Онтологія потоків і трафіку у комп'ютерних мережах. 7. Події та аномалії у комп'ютерних мережах. 8. Висновки  
8. Висновки

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>Осухівська Г.М., зав. каф. КС</i>		
	<i>Стадник І.Я., проф. каф. ОХ</i>		

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Аналіз особливостей моделювання та створення онтологій при управлінні комп'ютерними мережами</i>	<i>01.12.2023-07.12.2023</i>	<i>виконано</i>
2.	<i>Математична модель та архітектура онтології управління та моніторингу процесів в комп'ютерних мережах</i>	<i>07.12.2023-14.12.2023</i>	<i>виконано</i>
3.	<i>Результати моделювання процесів управління комп'ютерними мережами</i>	<i>14.12.2023-16.12.2023</i>	<i>виконано</i>
4.	<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>17.12.2023</i>	<i>виконано</i>
5.	<i>Оформлення пояснювальної записки</i>	<i>18.12.2023</i>	<i>виконано</i>
6.	<i>Оформлення графічного матеріалу</i>	<i>20.12.2023</i>	<i>виконано</i>
7.	<i>Попередній захист кваліфікаційної роботи магістра</i>	<i>21.12.2023</i>	<i>виконано</i>
8.	<i>Захист кваліфікаційної роботи магістра</i>		

Студент \_\_\_\_\_

(підпис)

*Кучма І.М.*

(прізвище та ініціали)

Керівник проекту (роботи) \_\_\_\_\_

(підпис)

*Яцишин В.В.*

(прізвище та ініціали)

## АНОТАЦІЯ

Методи та засоби моделювання процесів управління та моніторингу у комп'ютерних мережах // Кваліфікаційна робота магістра// Кучма Іван Михайлович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, група СІм-61 // Тернопіль, 2023 // с. – 88 , рис. – 36 , табл. –12 , аркушів А1 –8 , додат. – 1, бібліогр. – 26 .

Ключові слова: метод, засіб, моделювання, комп'ютерна мережа, управління, моніторинг.

У кваліфікаційній роботі магістра проведено аналіз методів моделювання комп'ютерних мереж у результаті якого представлено концепцію онтологій та їх компонентів, що дало змогу визначити способи організації спеціалізованих баз даних для реалізації процесів управління ресурсами та моніторингу стану мережі.

Запропоновано та обгрунтовано застосування математичної моделі для опису онтологій комп'ютерних мереж за допомогою кортежа, який включає в себе п'ять елементів: домен, можливі ситуації або стани системи, операції, функція відображення сутностей домену та поведінка об'єктів. Спроектовано архітектуру онтології для представлення об'єктів, відношень та обмежень в комп'ютерних мережах, що дало змогу забезпечити та змодельовати домен для опису процесів управління та моніторингу.

Розроблено компоненти онтології для представлення та аналізу трафіку і потоків даних у комп'ютерних мережах, що дало змогу забезпечити проведення аналітики завантаженості каналів та її продуктивності.

## ABSTRACT

Methods and tools for modeling processes in the management and monitoring of computer networks /Master's graduation thesis / Kuchma Ivan / Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm -61 // Ternopil, 2023// p. - 88, fig. – 36, table. – 12, Sheets A1 – 8, Add – 1, Ref. – 26.

Keywords: method, tool, modelling, computer network, management, monitoring.

In the master's qualification work, an analysis of computer network modeling methods was carried out, as a result of which the concept of ontologies and their components was presented, which made it possible to determine the methods of organizing specialized databases for the implementation of resource management processes and network state monitoring.

The application of a mathematical model for describing the ontologies of computer networks using a tuple that includes five elements is proposed and justified: the domain, possible situations or states of the system, operations, the function of displaying domain entities, and the behavior of objects.

The ontology architecture was designed to represent objects, relationships and constraints in computer networks, which made it possible to provide and model a domain for describing management and monitoring processes. Ontology components were developed for the presentation and analysis of traffic and data flows in computer networks, which made it possible to provide analysis of channel load and its performance.

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1 АНАЛІЗ ОСОБЛИВОСТЕЙ МОДЕЛЮВАННЯ ТА СТВОРЕННЯ ОНТОЛОГІЙ ПРИ УПРАВЛІННІ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ .....	12
1.1. Аналіз базових понять онтологічного моделювання .....	12
1.2. Формальне представлення онтологій.....	15
1.3. Класифікація онтологій.....	18
1.4. Аналіз принципів проектування та розробки онтології.....	22
1.5. Підходи до оцінювання онтології .....	25
1.6. Висновки до розділу .....	26
РОЗДІЛ 2 МАТЕМАТИЧНА МОДЕЛЬ ТА АРХІТЕКТУРА ОНТОЛОГІЇ УПРАВЛІННЯ ТА МОНІТОРИНГУ ПРОЦЕСІВ В КОМП'ЮТЕРНИХ МЕРЕЖАХ .....	28
2.1. Лексикон .....	28
2.2. Граматика.....	29
2.3. Синтаксичні обмеження утворення термів .....	32
2.4. Структури моделі онтології .....	34
2.5. Архітектура онтології управління комп'ютерними мережами.....	38
2.6. Висновки до розділу .....	44
РОЗДІЛ 3 РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ ПРОЦЕСІВ УПРАВЛІННЯ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ .....	45
3.1. Побудова комунікаційної інфраструктури.....	45
3.2. Моделювання трафіку та його розподілу у комп'ютерній мережі .....	52
3.3. Події та аномалії.....	57
3.4. Інструменти управління та моніторингу комп'ютерної мережі .....	63
3.5. Висновки до розділу .....	67

РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	68
4.1. Охорона праці.....	68
4.2. Оцінка стійкості роботи об'єкту економіки до впливу вражаючих факторів ядерної зброї .....	71
4.3. Забезпечення захисту працівників суб'єктів господарювання та населення від впливу іонізуючих випромінювань .....	75
ВИСНОВКИ.....	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	81
Додаток А Тези конференцій .....	84

## ВСТУП

**Актуальність теми.** Організація комп'ютерних систем з використанням сучасних технологій і сервісів є запорукою конкурентоздатності на IT-ринку та у бізнес-доменах, автоматизацію яких вони забезпечують. Враховуючи тренди щодо інтеграції елементів штучного інтелекту в існуючі системи, необхідність опрацювання великих масивів інформації, а також застосування розумних пристроїв абсолютно завжди виникає задача організації комунікаційної інфраструктури для забезпечення ефективності обміну даними, їх передачі і приймання.

Комп'ютерні мережі представляють собою базис для функціонування більшості комп'ютерних систем, окрім локальних замкнених систем, і можуть забезпечувати комунікацію між окремими пристроями вимірювання даних і подальшого їх опрацювання. Однак при проектуванні комп'ютерних мереж доволі часто виникають проблеми щодо узгодження принципів їх організації, розуміння замовниками основних компонентів системи, представлення топології мережі та ряду інших. У зв'язку з цим важливими та актуальними задачами у сфері організації комп'ютерних мереж є побудова онтології для управління та моніторингу компонентами мережі. Онтології представляють собою формальну та явну специфікацію спільної концептуалізації, забезпечують ефективність структури при організації знань і розуміння зв'язків у межах домену.

У неоднорідному та складному світі комп'ютерних мереж різноманітні пристрої та системи часто використовують різні термінології. Онтології усувають ці семантичні прогалини, забезпечуючи безперебійну комунікацію та співпрацю між різними компонентами. Онтології служать також формальним представленням знань, які фіксують концепти, сутності та їхні взаємозв'язки в межах домену комп'ютерної мережі. Ефективне керування мережею залежить від глибокого розуміння структури та поведінки мережі. Онтології забезпечують структуроване представлення,



яке полегшує точне моделювання мережевих компонентів, конфігурацій і взаємозв'язків, сприяючи ефективному моніторингу та аналізу. В управлінні мережею швидке та обґрунтоване прийняття рішень має вирішальне значення, а онтології дають змогу системам керування автономно аналізувати стан мережі, виявляти аномалії та пропонувати оптимальні конфігурації або відповіді, підвищуючи загальну ефективність мережевих операцій. Дослідженню методів побудови, управління та моніторингу комп'ютерних мереж присвячено багато праць як українських, так і закордонних учених. Серед українських вчених у галузі комп'ютерних мереж, які зробили значний вклад у їх розвиток, потрібно відмітити таких, як Виноградов М., Буров Є., Митник М., Микитишин А. та ін., серед закордонних – Таненбаум Е, Олифер Н., Олифер В., Петерсон Л, Дейві Б. і т.д. Проте важливим та актуальним на сьогодні все ж залишається побудова онтології комп'ютерних мереж в контексті управління та моніторингу їх параметрів.

**Мета кваліфікаційної роботи** полягає у дослідженні та побудові онтології процесів управління та моніторингу комп'ютерних мереж.

**Задачі**, розв'язок яких, необхідно забезпечити у кваліфікаційній роботі магістра, полягають в наступному:

- аналіз особливостей організації, структури, управління та моніторингу комп'ютерних мереж;
- дослідження принципів і правил організації онтологій;
- визначення компонентів і властивостей, необхідних для побудови онтології комп'ютерних мереж;
- обґрунтування математичного забезпечення для представлення онтологій комп'ютерних мереж;
- побудова архітектури онтології для забезпечення процесів управління та моніторингу параметрів комп'ютерних мереж;
- розробка засобами мови OWL онтології управління та моніторингу комп'ютерних мереж;

– оцінювання побудованої онтології щодо відповідності критеріям ефективності організації комп'ютерних мереж.

**Об'єкт дослідження:** процеси управління та моніторингу параметрів комп'ютерних мереж.

**Предмет дослідження:** методи і засоби побудови онтології комп'ютерних мереж.

**Методи дослідження:** Для вирішення поставлених у кваліфікаційній роботі задач використано методи аналізу та узагальнення – при дослідженні методів і засобів моделювання комп'ютерних мереж, теорії онтологій – при формальному описі компонентів домену комп'ютерних мереж; проектування і програмування – при побудові онтології комп'ютерної мережі з використанням мови OWL; експеримент та вимірювання – при моделюванні процесів управління та моніторингу ресурсів комп'ютерної мережі.

**Наукова новизна отриманих результатів.** Наукова новизна, одержаних у роботі результатів полягає в наступному.

– уперше запропоновано та обгрунтовано застосування математичної моделі для опису онтологій комп'ютерних мереж за допомогою кортежа, який включає в себе п'ять елементів: домен, можливі ситуації або стани системи, операції, функція відображення сутностей домену та поведінка об'єктів, що дають змогу забезпечити повноту та ефективність при описі процесів управління та моніторингу ресурсів комп'ютерної мережі.

– уперше визначено лексикон, граматику та синтаксичні обмеження утворення термів при побудові онтології комп'ютерної мережі, що дало змогу формувати складні висловлювання на основі предикатів та забезпечити гнучкість моделювання процесів, які протікають у комп'ютерній мережі.

**Практичне значення одержаних результатів.** Побудовано онтологію комунікаційної інфраструктури комп'ютерної мережі,

розроблено компоненти для представлення та аналізу трафіку і потоків даних у комп'ютерних мережах, що дало змогу забезпечити проведення аналітики завантаженості каналів та її продуктивності.

**Публікації.** Результати кваліфікаційної роботи апробовані на XII Міжнародній науково-практичній конференції молодих учених та студентів (6-7 грудня 2023 р.) та XI науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року) як тези конференцій.

1. Яцишин В.В., Кучма І.М. Побудова онтологій як спосіб ефективного моделювання комп'ютерних систем та мереж. Матеріали XII міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 439.

2. Яцишин В.В., Кучма І.М. Класифікація онтологій в процесі моделювання комп'ютерних мереж. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 162.

**Структура роботи.** Кваліфікаційна робота містить розрахунково-пояснювальну записку та графічний матеріал. До складу записки входить вступу, 4 розділи, загальні висновки, список використаних джерел і додатки. Обсяг роботи: розрахунково-пояснювальна записка – 88 арк. формату А4, графічна частина – 8 аркушів формату А1.

## РОЗДІЛ 1

### АНАЛІЗ ОСОБЛИВОСТЕЙ МОДЕЛЮВАННЯ ТА СТВОРЕННЯ ОНТОЛОГІЙ ПРИ УПРАВЛІННІ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ

Дослідження кваліфікаційної роботи магістра стосуються методів і засобів організації онтологій при управлінні комп'ютерними мережами. Однак перед тим як почати її побудову необхідно проаналізувати особливості процесу та об'єктів, які є важливими при проектуванні комп'ютерних мереж і формують відповідний домен.

#### 1.1. Аналіз базових понять онтологічного моделювання

У сфері інформатики та інформаційних технологій в цілому, онтологія визначається у вигляді структурованого набору термінів і понять, що представляють значення інформаційних доменів за допомогою метаданих простору імен чи елементів домену знань [1].

Онтології використовуються у багатьох сферах, зокрема, в штучному інтелекті, при побудові семантичних мереж, у галузі розробки програмного забезпечення, при виконанні біомедичних обчислень, при організації та проектуванні комп'ютерних систем різного призначення і є формою представлення знань про світ або певну його частину.

Оригінальне означення онтології було визначено Грубером у 1993 як явну специфікацію концептуалізації [1]. Онтології пропонують загальний словник реального світу та визначають значення термінів і зв'язки між ними з різним ступенем формальності.

Онтології, зазвичай, організовуються в таксономії та включають примітиви моделювання, такі як класи, відношення, функції, аксіоми та екземпляри [1]. Управління знаннями, системи, засновані на знаннях, брокери на основі онтологій і взаємодія між системами є яскравими прикладами популярних застосувань онтологій.

Виходячи з цього, онтологія – це ієрархія опису понять, пов'язаних зв'язками підпорядкування; у більш складних моделях додаються відповідні аксіоми, щоб полегшити інші зв'язки між поняттями.

Незважаючи на те, що використання онтологій передбачає конкретний підхід до створення спільних баз знань, воно також викликає низку питань:

- Яке походження онтології?
- Як це має бути розроблено?
- Які інструменти потрібно використати для його побудови?
- Яких принципів проектування необхідно дотримуватися для побудови онтологій?
- Як слід використовувати онтологію?
- Як його слід розробляти та оцінювати?

У контексті штучного інтелекту онтологія ототожнюється з набором формальних термінів, що представляють знання. Грубер [1] визначає онтологію як явну специфікацію концептуалізації. Концептуалізація – це абстрактне уявлення світу, який потрібно зобразити для досягнення деякої поставленої мети.

Представлення знань базується на концептуалізації і містить об'єкти, поняття та зв'язки, які існують між ними. Виходячи з технічної точки зору, визначення [2] онтологію розглядають як специфікацію типів сутностей, які існують або можуть існувати у певному домені чи предметній області.

Більш формально, онтологія може бути представлена набором імен, що проектують концепції та типи відношень, контрольованих у частковому порядку за допомогою відношення типу/підтипу. Наведемо найбільш актуальні означення онтології, які можна використати при побудові онтології управління комп'ютерними мережами.

Означення 1: Згідно з Гуаріно [3], онтологія визначається як інженерний артефакт, утворений спеціальним словником, який

використовується для опису певної реальності, на додаток до набору явних припущень щодо передбачуваного значення словникових слів.

Визначення 2: Відповідно до [4], онтології розглядаються як свого роду «декларативні абстрактні ієрархії верхнього рівня, представлені достатньою інформацією для встановлення основних правил для моделювання домену». Проте онтологія визначає основні терміни та зв'язки, включаючи лексику тематичної області, а також правила поєднання термінів і зв'язків для визначення розширень словника.

Визначення 3: Вілінга та Шрайбер визначили онтологію [5] ШІ, як «теорію того, які сутності можуть існувати в свідомості обізнаного агента». На онтологію можна посилатися як на мета-модель, яка описує структуру бази знань і робить явними відповідності, які використовуються в процесі моделювання, що забезпечує повторне використання та обмін знаннями.

Визначення 4: визначення Грубера [1] було переосмислено Гуаріно та Джареттою [6]. Вони намагалися уточнити значення онтології, беручи до уваги 7 значень термінів, які, можливо, будуть використовуватися в літературі: Онтологію можна розглядати у 7 аспектах:

- як філософська дисципліна;
- як неформальну концептуальну систему;
- як формальну семантичну мережу;
- як специфікацію концептуалізації;
- як представлення концептуальної системи через логічну теорію;
- як словник, який використовується в логічній теорії;
- як конкретизація логічної теорії.

Автори безпосередньо використовують значення від 2 до 7 з додатковою увагою до значення 4, яке вони вважають проблематичним. Тому Гуаріно та Джаретта описують онтологію ШІ як таку, що має два значення:

– Онтологія – це логічна теорія, представлена як «спроектований артефакт, база знань особливого роду, яку можна читати, продавати або фізично ділитися».

– Онтологія – це синонім концептуалізації, що визначається як «навмисна семантична структура, яка кодує неявні правила, що обмежують структуру частини реальності».

Проте, згідно з Гуаріно та Джареттою, та сама онтологічна теорія може належати до різних концептуалізацій, і та сама концептуалізація може лежати в основі різних онтологічних теорій.

## 1.2.Формальне представлення онтологій

Формальне представлення знань в онтології забезпечується п'ятьма основними компонентами до яких входять «класи, їх екземпляри, відношення, функції, аксіоми та екземпляри» [1]. Далі наведено основні характеристики цих компонентів онтології.

Класи або поняття, які також називають типами або універсалами, це група екземплярів, які мають спільні характеристики, що використовуються в широкому сенсі. Поняттям може бути будь-який об'єкт, про який щось відомо. Це може бути опис завдання, дії, функції, стратегії, процес міркування тощо. Більшість онтологічних мов (наприклад, OWL, DAML тощо) дозволяють визначати поняття на основі цих характеристик. Наприклад, усі ссавці мають однакові характеристики, за винятком здатності розмовляти.

Наступний компонент онтології – відношення. В онтологіях відношення описують засоби, за допомогою яких об'єкти (екземпляри або властивості) пов'язані між собою. Іншими словами, відношення представляють собою форму взаємодії між поняттями в одній предметній області.

Формально відношення – це будь-яка підмножина добутку  $n$  множин, визначена таким чином:  $R: C1 \times C2 \times \dots \times Cn$ . Декілька типів зв'язків можна виразити в онтологіях: «підклас у» і «пов'язаний з» є двома прикладами бінарних зв'язків. Речення «Андрій викладає курс штучного інтелекту» виражає прямий зв'язок між окремими особами, але речення «Професор викладає курс» виражає зв'язок між поняттями.

Третій компонент онтології – функції. Функції представляють собою певний тип зв'язків, де  $n$ -й елемент зв'язку є відмінним від  $n-1$  попередніх елементів. Відношення «Author of» і «Price-of-a-new-computer» є двома прикладами функцій.

Наприклад, друга функція має на меті розрахувати ціну нового комп'ютера залежно від типу та швидкості процесора, ємності жорсткого диска та пам'яті.

Четвертий тип онтології – аксіоми. Аксіоми представляють твердження, сформульовані в логічній формі, які разом складають основні знання, які онтологія описує в прикладній області. Іншими словами, аксіоми використовуються для моделювання речень, які завжди істинні. Типи аксіом можна класифікувати відповідно до їх семантичного значення [7].

Екземпляри — це об'єкти конкретних класів, які моделюють конкретні сутності і представляють базові компоненти онтології.

Якщо основні компоненти онтологій представлені, то можна реалізувати онтологію різними мовами: високонформальними, напівнеформальними, напівформальними та строго формальними [8].

### 1.3. Повторне використання онтології

Повторне використання онтології є важливим питанням дослідження у сфері побудови семантичних мереж та онтологій. Його можна визначити як процес, у якому доступні (онтологічні) знання використовуються як



вхідні дані для створення нових онтологій. Далі буде проведено визначення категорій повторного використання онтологій.

Злиття онтологій – це підхід, спрямований на інтеграцію двох або більше онтологій для створення унікальної онтології, яка мала б/включала б всі знання, які мали об'єднані онтології. Потім процес злиття онтологій створює унікальну онтологію – об'єднана версія оригінальних онтологій. Отримана онтологія містить всю інформацію з об'єднаних оригінальних онтологій без вказання їхнього попереднього походження [10]. Процес злиття зазвичай виконується, коли вихідні онтології охоплюють схожі або такі, що перекриваються домени.

Інтеграція онтологій – це процес створення унікальної онтології після агрегації, збірки, розширення, комбінування, спеціалізації або адаптації онтологій на різні теми [10].

Відображення онтологій – це процес визначення відповідності між сутностями онтологій. Відображення використовує кілька онтологій для виконання відображень між ними, але без наявності глобальної онтології. На виході процес відображення онтологій формує набір тверджень відображення, що позначають відношення між цими сутностями [11]. Однак мета цього процесу полягає в тому, щоб дозволити онтологіям обмінюватися та повторно використовувати інформацію між собою.

Узгодження онтологій – це процес визначення відповідності між концепціями, що містяться у двох онтологіях, з метою модифікації однієї з них, щоб зробити її більш узгодженою з іншою [12]. Для реалізації цього процесу розроблено декілька інструментів узгодження, які працюють зі схемами баз даних, схемами XML, таксономіями, моделями зв'язків сутності, формальними мовами, словниками та іншими структурами позначок.

Версії онтології вказують на наявність кількох варіантів її існування. Насправді ці варіанти часто походять від змін до існуючого варіанту онтології і, таким чином, утворюють дерево похідних онтологій. Однак

керування версіями онтології — це зміна опрацювання процесу в різних версіях онтології, що передбачає розпізнавання версій, оновлення онтологій і відстеження зв'язків версій [13].

### 1.3.Класифікація онтологій

Для класифікації онтологій використовується поняття онтологічної прихильності. Якщо є можливість розрізнити типи онтологічної прихильності, то можна ідентифікувати різні типи онтологій, а отже, можна класифікувати онтології.

Для визначення різних типів прихильності загальноприйнятим у сфері штучного інтелекту є розмежування між завданнями, методами та доменами, як показано на рис. 1.1.

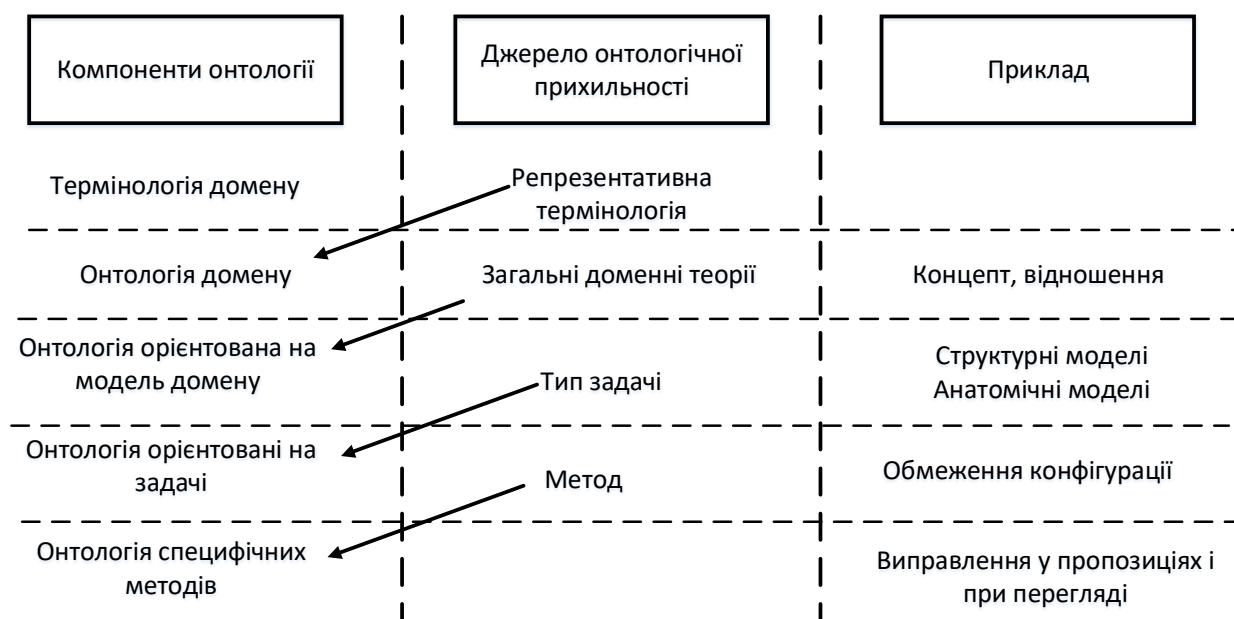


Рис. 1.1. Типи прихильності при класифікації онтологій

Таким чином, можна виділити три різні типи онтологічних прихильності: прихильність у випадку задач, прихильність методу та прихильність домену.

Формулювання прихильності задач можна виразити наступним чином: якщо онтологія визначає сутності та зв'язки, що виражають перспективу конкретного завдання на знання предметної області, тоді вона має прихильність щодо завдань [16]. Задачі можна визначити як специфікацію цілі, включаючи деякі вхідні та необхідні результати. Онтологія для задач діагностики, включає такі сутності, як спостереження, гіпотези та причини і є прикладом онтології, яка має прихильність до задач.

Представлення прихильності методів означають наступним чином: якщо онтологія визначає сутності та зв'язки, які виражають специфічну для методу точку зору на знання предметної області, тоді вона має прихильність методу [16]. Метод можна вказати як специфікацію того, як можна виконати завдання. Онтологія, яка використовує метод «запропонувати та переглянути» (в рамках завдання проектування), який містить такі сутності, як запропоноване рішення, оцінка значення та обмеження, є прикладом онтології, яка має прихильність методу.

Доменна прихильність означається так: якщо онтологія визначає сутності та зв'язки, що стосуються конкретного домену, то вона має зобов'язання і прихильність щодо домену. Домен відноситься до загальноприйнятих фрагментів змодельованого реального світу, таких як медична, судова, фінансова, математична чи соціальна сфери. Онтологія для сфери судової влади, яка містить такі сутності, як норми та акти, є прикладом онтології, яка має зобов'язання щодо сфери. Крім того, онтологія домену надає словники про концепції даної області та їхні взаємозв'язки, про діяльність, яка відбувається в цій області, а також про теорії та основні принципи, які керують сферою

Базуючись на принципах, наведеної вище типології онтологій у [7] запропоновано типологію, яка заснована на наступній ідеї: «З точки зору технології бази знань, знання слід розглядати в певному контексті, тобто в ситуації вирішення проблеми». З цієї точки зору встановлюються такі типології, які проілюстровано на рис. 1.2.

Онтології вмісту також називаються онтологіями для повторного використання знань, вони також включають онтології завдань, робочого місця (онтологія домену) і загальні/загальні онтології (включаючи лексику, пов'язану з речами, подіями, простором, часом, поведінкою, функцією причинності тощо).

Автори у [7] також беруть до уваги концепцію мета-онтології, яку також часто називають як загальною онтологією або основною онтологією як четвертий основний тип для позначення онтологій представлення.

Комунікаційні онтології називаються онтологіями обміну знаннями. Цей тип онтологій призначений для підтримки спільного використання та повторного використання представлених знань у формальний спосіб. Вони є цінними для визначення спільного словника, в якому представлені спільні знання. Ontolingua є прикладом онтологій, які використовують обмін знаннями.

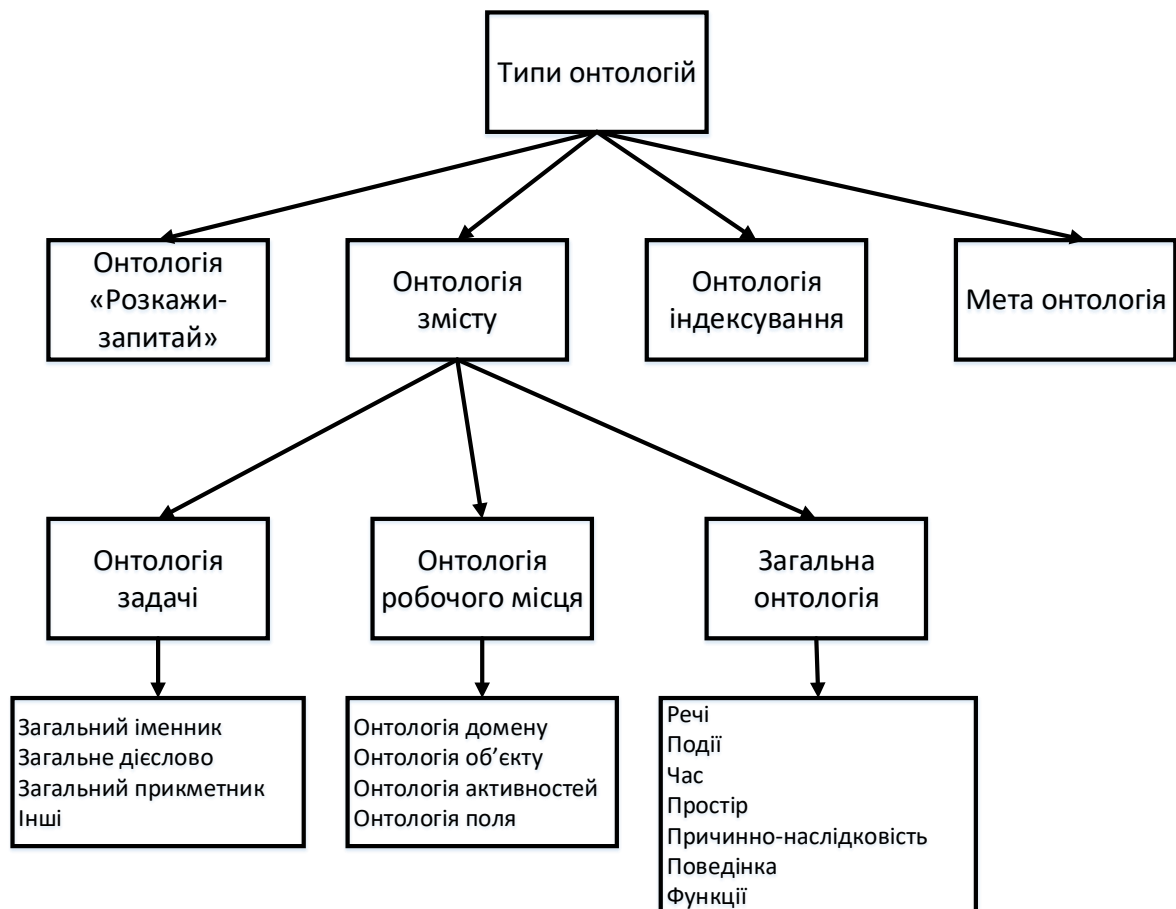


Рис. 1.2. Доповнення класів онтологій

Онтології індексування називаються онтологіями для пошуку інформації. Метою впровадження онтологій у пошук інформації є покращення запам'ятовування та точності [9]. Його основне використання пов'язане з розширенням запиту, що полягає в пошуку термінів в онтології, більш пов'язаних із термінами запиту, і, отже, використання їх як частини запиту.

Можна виділити два типи складності онтології в залежності від широкого кола завдань, до яких включені онтології. Діапазон складності онтологій варіюється від простих таксономій до дуже складних, включаючи обмеження, пов'язані з концепціями та зв'язками.

Спрощені онтології, зазвичай, визначені як більш ієрархічні або класифікаційні характеристики. Вони призначені для представлення субсумпційного відношення (існує єрархія між поняттями) або інших типів зв'язків між поняттями. Такий тип онтології не включає занадто багато або надто складних відношень. Як приклад, ці онтології можна застосовувати в Інтернет пошукових системах, зокрема це стосується онтології Yahoo, що складається з ієрархії тем без урахування точного визначення поняття, різниці між словом і поняттям тощо. Формальне означення спрощеної онтології наведено в [10].

Другий тип онтологій відрізняється від спрощеної онтології. Онтологія з більшою вагою (строго типізована) розроблена з додатковою увагою до строгого значення, принципів організації кожного поняття та семантично строгих зв'язків між поняттями (обмеження кількості, таксономія відносин, аксіом або обмежень тощо). Цільове моделювання світу, яке потребує глобальної концептуалізації світу, щоб гарантувати послідовність і точність моделі, потребує екземплярів моделей, які зазвичай побудовані на даному типі онтології. Іншими словами, у міру того, як додаються зв'язки та зростає складність домену, онтології мігрують від спрощеного до «складнішого» представлення.

#### 1.4. Аналіз принципів проектування та розробки онтології

Онтології можна визначити за допомогою артефактів, які мають структуру (логічну, лінгвістичну, «таксономічну»). Їхня мета полягає в кодуванні опису домену (фактичного, контрфактичного, можливого, неможливого, бажаного тощо) для деяких завдань або проблем (наприклад, область медицини, світ семантичної веб-конференції тощо). Однак онтології повинні відповідати домену та завданню, описаному нижче:

- домен – дозволяє описувати сутності, атрибути та зв'язки які стосуються певної мети (студент як сутність, яка зарахована до університету, під керівництвом академічного персоналу та має назву, адресу тощо);

- завдання – допомагає досягти мети (знайти людей, які працюють над тією самою темою, узгодити теми проекту з компетенціями персоналу тощо).

Проектування онтології є досить складним завданням для складних комп'ютерних систем та мереж і важливою технікою у створенні додатків, заснованих на знаннях [11].

У декількох областях досліджень [12,13] успішно застосовано проектування онтології, зокрема, такі види як перевірка моделі та семантичний аналіз, виявлення неузгодженості в моделюванні складного сценарію.

До основних критеріїв і принципів проектування, які виявилися корисними при розробці онтологій належать:

- об'єктивність – означає, що онтологія повинна надавати значення визначених термінів шляхом надання об'єктивних визначень, а також документації природною мовою;

- повнота – вираження означення базується на необхідних і достатніх умовах, що є кращим перед частковим означенням;

- когерентність – прийнята, щоб формувати висновки, які представляють узгодженість з означеннями;

- максимальна монотонна розширюваність – властивість, яка передбачає, що нові загальні або спеціалізовані терміни повинні бути включені в онтологію таким чином, щоб не вимагати зміни існуючих понять та їх означень;

- мінімальні онтологічні зобов'язання – означає мінімізацію претензій щодо області, яка моделюється, надаючи сторонам свободу спеціалізації та інстанціювання онтології як вимагається;

- диверсифікація ієрархій – корисна для збільшення потужності, що забезпечується кількома механізмами наслідування;

- принцип онтологічного розрізнення – передбачає, що класи в онтологіях повинні бути такими, які не перетинаються;

- мінімізація семантичної дистанції між однотипними поняттями – однакові екземпляри групуються та використовуються для представлення подібних понять.

- модульність – властивість, яка використовується для мінімізації зв'язку між модулями.

- стандартизація назв.

У літературі розроблено велику кількість методів і практик для інженерії онтології. При цьому важливим є те, що всі ці методи вимагають, щоб онтології створювалися або кваліфікованими інженерами, або шляхом співпраці між фахівцями в предметній області та експертами. Лише деякі методи підтримують експертів домену. Розробка вимог відіграє важливу роль у всьому представлені методів.

Грюнінгер і Фокс представили питання компетентності як набір проблем, які є онтологією аксіоми логіки, які потрібно вміти представляти та розв'язувати. Виходячи з їх точки зору таке питання має бути формалізовано у машинно-інтерпретовані та розв'язувані проблеми, а не у формат природної мови. Питання компетенції в онтологіях RDFS і OWL

часто формалізуються в запити SPARQL, і якщо запит SPARQL повертає очікуваний результат після виконання над онтологією, про яку йде мова, вони вважаються задоволеними.

Загальноприйнятим є виконання семи кроків для розробки онтології, які описані нижче.

Визначення домену і області онтології: цей крок має визначити домен, який буде охоплювати онтологія, вказати типи запитань, на які інформація в онтології має надати відповіді (онтологія корисна для тестування), вказати, для кого ця онтологія призначена (користувачі можуть бути клієнтами ресторану, професіоналами (шеф-кухарями). )) і перевірити, чи містить онтологія достатньо інформації, щоб відповісти на ці типи питань.

Повторне використання існуючих онтологій: цей крок має визначити, чи потрібно відповідній системі взаємодіяти з іншими системами, які вже містять певні онтології або словники обмежень (бібліотека Ontolingua, бібліотека DAML тощо). Крім того, цей крок перевіряє, якщо існує вимога до вдосконалення та розширення існуючих джерел для певного домену.

Визначення ключових термінів в онтології: на цьому кроці має бути визначено список термінів, які хотілося б пояснити користувачеві, що описує твердження щодо певної теми. Крім того, в ньому слід вказати властивості, укладені в термінах, і їх тлумачення (різноманітні види їжі, наприклад риба і червоне м'ясо). Список термінів визначається незалежно від категоризації класу (ієрархія, фасет і перекриття).

Визначення класів та ієрархії класів: на цьому етапі потрібно вибрати один із трьох можливих підходів: низхідний процес розробки, який починається з найзагальнішого визначення концепції, висхідний процес розробки, який починається з найбільш конкретного визначення концепції, або комбінований підхід.

Визначення властивостей класу: цей крок починається з визначення класів зі списку, створеного на кроці 3, і їх внутрішньої структури (властивостей). Клас визначається шляхом опису об'єктів, які мають



незалежне існування. Крім того, на цьому кроці потрібно організувати класи в ієрархічну структуру (таксономія). Слот – це слово, яке не є класом, і, крім того, його слід призначити класу.

Визначення аспектів слотів – різні аспекти можуть характеризувати слот, який може містити опис типу значення (типи значень, які дозволено заповнювати в слоті: кількість (наприклад, для ціни)), дозволені значення, потужність (визначає число значень, яке може мати слот), та інші додаткові особливості значень, які може приймати слот (тобто значення імені є рядком).

Створення екземплярів: цей крок визначає окремий екземпляр класу на основі такого порядку: обирається клас, потім створюється окремий екземпляр цього класу і, нарешті, заповнюються значення слота.

### 1.5. Підходи до оцінювання онтології

Процес оцінювання онтології може ґрунтуватися та розглядатися з різних точок зору: наприклад, якість розробленої онтології (технічна точка зору) і її придатність для використання в практичному сенсі (практична точка зору). Крім того, можна оцінити онтологію на основі математичної моделі (математичної точки зору). Для оцінки якості розробленої онтології запропоновано декілька метрик: наприклад, п'ять критеріїв Грубера: зрозумілість, когерентність, розширюваність, мінімальне зміщення кодування та мінімальна онтологічна прихильність.

Онтології включають набори споріднених понять. Загалом, теорія множин використовується для побудови математичних моделей для визначення та аналізу онтології. Наприклад, OntoQA можна розглядати, як підхід, призначений для оцінювання якості онтологій та їхнього потенціалу при представленні знань за допомогою чітко визначеного набору показників. У цьому підході визначено три категорії показників:

Метрика схеми – оцінює дизайн онтології та складається з метрик, які визначають повноту зв'язків, повноту атрибутів і повноту наслідування.

Показники бази знань – описують базу знань у цілому, яка складається з показників повноти класів, середньої кількості елементів та когедійних метрик.

Показники класу – часто називають розподілом екземплярів, оскільки виконується оцінювання розподілу екземплярів за класами. До метрик показників класу входить повнота, повнота наслідування, повнота зв'язків, зв'язність і читабельність.

Інший добре відомий підхід до оцінки онтології, представлений у OntoClean, успадкований від філософського підходу, який оцінює формальні властивості таксономії. OntoClean, з практичної точки зору, пропонує засоби для отримання вимірних невідповідностей таксономії, враховуючи семантику, яку забезпечує зв'язок «існує-існує». Крім того, OntoClean пропонує роз'яснення, чому виникають невідповідності, що згодом покращує таксономічну структуру. Властивості OntoClean застосовуються деякими інструментами, такими як OntoEdit і WebODE.

## 1.6.Висновки до розділу

Основні результати даного розділу полягають в наступному:

1. Проведено аналіз методів моделювання комп'ютерних мереж у результаті якого представлено концепцію онтологій та їх компонентів, що дало змогу визначити способи організації спеціалізованих баз даних для реалізації процесів управління ресурсами та моніторингу стану мережі.

2. Проведено аналіз базових понять при побудові онтологій комп'ютерних мереж, що дало змогу визначити основні сутності та відношення між ними для домену комп'ютерних мереж.

3. Проведено аналітичний огляд принципів проектування онтології та методології їх розробки, а також наведено метрики за якими можна проводити оцінювання якості онтологій, що дало змогу в перспективі забезпечити об'єктивність і достовірність моделювання комп'ютерних мереж.

## РОЗДІЛ 2

### МАТЕМАТИЧНА МОДЕЛЬ ТА АРХІТЕКТУРА ОНТОЛОГІЇ УПРАВЛІННЯ ТА МОНІТОРИНГУ ПРОЦЕСІВ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

Для проведення моделювання процесів управління та моніторингу комп'ютерних мереж на основі онтологій необхідно обґрунтувати математичне забезпечення онтології. Така робота, звісно, якщо вона є правильною, повинна призвести до розробок, у яких мови, які обробляються комп'ютером, є критичними, але на даному етапі можливість обробки не є проблемою.

#### 2.1. Лексикон

Лексикон складається із зліченної множини окремих констант, скінченної множини змінних для кожного  $n \geq 0$ , скінченної множини  $n$ -впорядкованих предикатних констант і зліченної  $n$ -розрядної множини предикатних змінних (їх спільно називають  $n$ -місцевими предикатами), зарезервованих логічних символів  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists, \lambda$  та  $\square$ , а також дужок різних типів..

Змінні формуються з малих літер, як правило,  $x, y, z$ , з можливістю застосування цифрових індексів. Змінні  $n$ -розрядного предикату утворюються з літер верхнього регістру з цифровими верхніми індексами (пригнічуються, якщо контекст служить для вказівки на арність  $n$ -розрядного предиката), як правило,  $F^n, G^n$  і  $H^n$ , можливо, також з числовими індексами.

Константи формуються з алфавітно-цифрових рядків, крім односимвольних, які вже використовуються для змінних і починаються з великої або малої літери. Тире може використовуватися для з'єднання буквено-цифрові рядків.

Як правило, стрічки починаються з малої літери для констант, призначених для позначення окремих об'єктів. Рядки, що починаються з великої літери використовуються для констант, призначених для позначення зв'язків.

## 2.2. Граматика

Формули і терміни визначають за допомогою одночасної рекурсії:

1. Будь-яка константа або змінна (індивід або предикат) є терміном.
2. Якщо  $\pi$  є  $n$ -розрядним предикатом і  $\tau_1, \dots, \tau_n$  є будь-якими термінами,  $n \geq 0$ , тоді  $\pi(\tau_1; \dots; \tau_n)$  є (атомарною) формулою  $\mathcal{L}$ . Кажуть, що  $\pi$  знаходиться в позиції предиката, а кожен  $\tau_i$  в позиції аргументу в  $\pi(\tau_1; \dots; \tau_n)$ . У випадку, коли  $n = 0$ , то можна опустити порожні дужки і сказати, що  $\pi$ , яке стоїть окремо, є атомарною формулою.
3. Якщо  $\varphi, \psi$  формули, то існують  $\neg\varphi, \Box\varphi$  і  $(\varphi \rightarrow \psi)$  також.
4. Якщо  $\varphi$  будь-яка формула і  $v_1, \dots, v_n$  будь-які змінні, тоді  $(\forall v_1 \dots v_n)\varphi$  є формулою.
5. Якщо  $\varphi$  є формулою, яка не містить входжень  $\Box$  жодних залежних змінних, що зустрічаються в позиції предиката, і жодних залежних змінних самого предикату, а  $v_1, \dots, v_n$  є будь-якими змінними, які не зустрічаються вільно в жодному терміні, що зустрічається у  $\varphi$ , тоді  $[\lambda v_1 \dots v_n \varphi]$  є  $n$ -місним предикатом.
6. Ніщо інше не є терміном або формулою  $\mathcal{L}$ .

Далі приймаються традиційні визначення  $\wedge, \vee, \leftrightarrow$  та  $\exists$ .

Є дві особливо відмінні властивості  $\mathcal{L}$ . По-перше, хоча мова  $\mathcal{L}$  містить так звані змінні «вищого порядку», на відміну від стандартних мов вищого порядку, ці змінні та  $n$ -розрядні предикати загалом вважаються термінами; вони можуть бути аргументами для інших предикатів.

Семантично, як буде чітко видно нижче, це означає, що універсум є вільним типом – усе є об’єктом, а квантори мови будуть однакові для всіх. Варто відмітити, що це не означає, що немає різниці між типами об’єктів.

Примітно, як уже зазначалося, базова онтологія включає відношення, а також звичайні об’єкти. Швидше, згідно з аксіомою 4, це просто означає, що всі ці об’єкти є в універсумі, тобто в діапазоні кванторів.

Усі сутності — екземпляри, пропозиції, властивості та відношення представляють першокласні логічні одиниці, які разом складають єдину область кількісного визначення.

Як такі, властивості та відношення можуть самі мати властивості, перебувати у відношеннях і служити потенційними об’єктами-посиланнями. Можливо, найвагомим лінгвістичним доказом свободи типу є явище номіналізації, за допомогою якого будь-яку дієслівну фразу можна трансформувати в іменникову фразу того чи іншого роду, найчастіше – герундій.

Так, наприклад, дієслівна фраза «відомий» вказує на властивість, яка може передаватися особам, як у «Квентін відомий». Однак його герундій «бути відомим» служить для позначення предмета подальшої предикації, як, наприклад, «бути відомим — це все, про що думає Квентін».

Інтуїтивно зрозуміло, що дієслівна фраза, що вказує на предикативну властивість, і герундій, що вказує на об’єкт думок Квентіна (тобто, об’єкт, що володіє властивістю думати про нього Квентіном), є одним і тим же, властивістю «бути знаменитим».

У  $\mathcal{L}$  ця «подвійна роль» властивостей і відношень — предиктивний об’єкт проти об’єкта предикації — відображається в тому факті, що та сама константа може виконувати як традиційні синтаксичні ролі символу предиката, так і індивідуальну константу. Таким чином, у  $\mathcal{L}$  можна записати обидва предикати у формі (2.1) і формі (2.2):

$$Famous(quentin) \tag{2.1}$$

$$(\forall F)(ThinksAbout(quentin, F) \leftrightarrow (F = Famous)) \quad (2.2)$$

У цьому випадку,  $\mathcal{L}$  не зберігає представлення граматичної різниці між дієслівними фразами, наприклад, «знаменитий» та їхніми відповідниками в герундії, наприклад, «бути відомим».

Можна було б додати ще один об'єкт, що досить легко, але оскільки немає семантичної різниці між дієслівними фразами та їхніми герундіями за безтиповою концепцією, будь-яке таке представлення було б семантично безглуздом.

Оскільки всі об'єкти належать до одного логічного типу, з цього випливає, що будь-яка властивість може бути використана і застосована до будь-якої іншої, зокрема, властивість може бути застосована (і, справді, може бути прикладом) до самої себе. Знову ж таки, це відповідає природній мові. Властивість бути властивістю визначає, що вона є екземпляром. Це природно представлено в  $\mathcal{L}$  очевидним чином:

$$Property(Property) \quad (2.3)$$

Необхідно підкреслити той факт, що пропонується кількісно оцінювати властивості, пропозиції та відношення, як правило, само по собі не означає, що  $\mathcal{L}$  є вищого порядку. Для цього семантика має включати квантори вищого порядку, діапазон яких включає конструкцію набору потужностей певного типу в області логічних екземплярів.

У пропонованій семантиці такої конструкції немає. Існує лише одна область, у якій охоплено один тип квантора. Другою відмінною рисою  $\mathcal{L}$ , і, мабуть, найпомітнішою, є наявність складних термінів  $[\lambda v_1 \dots v_n \varphi]$ . Інтуїтивно ці терміни позначають складні відношення. Наприклад, терм

$$[\lambda x Enjoys(x, salomon) \wedge Prefers(x, red_{wine}, white_{wine})] \quad (2.4)$$

Терм (2.4) вказує на властивість насолоджуватися лососем і віддавати перевагу червоному вину білому. Терміни без зв'язаних  $\lambda$ -змінних вказують на нульові відношення, тобто пропозиції. У цьому випадку  $\lambda$  можна опустити. Таким чином

$$[\forall x(Planet(x) \rightarrow Larger(sun, x))] \quad (2.5)$$

Вираз (2.5) вказує на припущення, що Сонце більше за всі планети.

Ця особливість  $\mathcal{L}$  особливо важлива, оскільки онтології в запропонованій теорії будуть характеризуватися приблизно як класи пропозицій, а логічні зв'язки між онтологіями будуть виражені в термінах логічних відношень між пропозиціями.  $\lambda$ -терми дозволяють явно говорити про пропозиції в даній онтології. І, як видно, вони також надзвичайно корисні для визначення різноманітності важливих допоміжних понять, які можуть стосуватися будь-якого домену, зокрема, у сфері управління та моніторингу комп'ютерних мереж.

### 2.3. Синтаксичні обмеження утворення термів

Пункт (5) у граматиці для  $\mathcal{L}$  накладає ряд обмежень на утворення складних термів. Найбільш заслуговує на увагу з них обмеження, яке дозволяє зв'язувати оператором  $\lambda$  тільки окремі змінні в комплексних термінах.

Це обмеження дозволяє уникнути парадоксу Рассела, оскільки без цього обмеження термін  $[\lambda F \neg F (F)]$ , який інтуїтивно вказує на властивість несамоподібності був би законним. Тоді граMATика дозволила б побудувати атомарну формулу  $[\lambda F \neg F (F)] [\lambda F \neg F (F)]$ , яку можна було б довести за логічним принципом  $\lambda$ -перетворення еквівалентно його запереченню.

Однак обмеження, яке запобігає парадоксу, не є випадковим. Його виправдання полягає в тому, що просто немає інтуїтивно зрозумілої



логічної операції, яка формує зв'язки, логічна форма яких відповідає таким термінам, і, отже, немає підстав для їхнього дозволу.

Уникнення парадоксу Рассела впливає як наслідок цього обмеження, і, отже, пояснюється, а не просто уникається – парадокс виникає через теоретично необґрунтоване припущення про структуру складних відношень, подібно до того, як відповідний парадокс приналежності до себе виникає через теоретично невиправдане припущення про природу та структуру множин.

Пункт 5 граматики накладає низку інших обмежень на формування термінів, які, по суті, не є необхідними в тому сенсі, що можна було б фактично забезпечити розумну семантику для них, зокрема:

- Вимога, щоб усі  $\lambda$ -зв'язані змінні зустрічалися самі по собі у  $\varphi$  правилах виключає такі терміни, як  $[\lambda x y P x]$ , які містять порожні  $\lambda$ -зв'язані змінні;

- Обмеження вільного входження  $\lambda$ -зв'язаних змінних у складні терміни, що зустрічаються у правилах  $\varphi$ , виключає такі терміни, як  $[\lambda x y P [\lambda z Q x z] y]$ ;

- Обмеження щодо зв'язаних входжень предикатних змінних у складних термінах, що зустрічаються в  $\varphi$ , виключає такі терміни, як  $[\lambda y (\exists F^1) y = F^1]$ .

Однак терміни, які були б дозволені без цих обмежень, несуттєві для цілей, і, отже, їх дозвіл створив би непотрібну технічну складність. Хоча обмеження на немодальні формули у формуванні термів, як і два, наведених вище, також несуттєві, вони мають певну інтуїтивну підставу.

Оскільки, на відміну від трьох наведених вище обмежень, це обмеження відображає важливу особливість передбачуваної області, яка спрямовує розробку поточної структури. Зокрема, формулювання теорії онтологій першого порядку, тобто онтологій, складові пропозиції яких можна виразити реченнями на немодальній мові першого порядку (отже, на будь-якій її слабшій підмові).

Це, звичайно, не означає, що не існує модальних (або вищих) онтологій. Проте переважна більшість існуючих онтологій є першорядними, і здається малоімовірним, що це зміниться з розвитком семантичної мережі, якщо виразність її базової мови буде в порядку DAML+OIL.

Таким чином, надання можливості для вираження модальних пропозицій на даний момент виглядає невиправданим. Однак теорія онтологій потребує цієї виразності. Зокрема, модальність корисна для характеристики природи онтологій та їхніх логічних зв'язків. Найпомітніше, можливо, як буде чітко показано далі, модальний компонент мови пропонованої теорії дозволяє визначити надійне поняття втілення, яке, у свою чергу, може бути використане для формулювання відповідного стійкого поняття онтологічного змісту.

#### 2.4. Структури моделі онтології

Структура моделі  $\mathcal{M}$  для  $\mathcal{L}$  представляє собою кортеж з п'ятьма елементами

$$\langle D, W, dom, Op, ext \rangle \quad (2.6)$$

де  $D = \cup\{R, A\}$  – є доменом моделі  $\mathcal{M}$ , що складається з об'єднання двох взаємно неперетинних множин  $A$  і  $R$ . Множина  $A$  представляє собою множину екземплярів  $D$  і  $R$  – це набір відношень, які можуть бути одним з видів пропозицій.  $R$  сама по собі може бути розділена двома способами.

По-перше,  $R$  є об'єднанням двох взаємно неперетинних непорожніх множин  $R^p$  і  $R^c$  інтуїтивно зрозумілої сукупності логічно примітивних і логічно складних відношень відповідно.

Крім того,  $R$  є об'єднанням скінченної кількості непорожніх множин  $R_0, R_1, \dots$ , кожен  $R_n$  є інтуїтивним класом  $n$ -місних відношень. Припустимо, що  $R_n^p$  і  $R_n^c$  такі, що  $R^p \cap R_n$  і  $R^c \cap R_n$  відповідно.

$W$  є непорожньою множиною «можливих універсумів» або «можливих ситуацій». Якщо говорити більш формально,  $W$  надає моделі модальності, яка дозволяє представити реалізацію та інші логічні відношення між онтологіями. Відповідно,  $dom$  – це функція, яка відображає кожен елемент  $w$  з  $W$  на підмножину  $dom(w)$  з  $D$ , що представляє набір об'єктів, які «існують» у можливому універсумі  $w$ .

Наступний елемент структури моделі  $Op$  – це множина із п'яти підмножин логічних операцій:

- набір операцій предикатів  $\{Pred_{i_1 \dots i_k}^n, 0 < i_1 < \dots < i_k \leq n; k \geq 0\}$ ;
- набір логічних операцій  $\{Neg; Impl\}$ ;
- набір операцій універсуму  $\{Univ_{i_1 \dots i_k}, 1 \leq i_1 < \dots < i_k \leq n; \}$ ;
- набір операцій перетворення,  $\{Conv_j^i : 1 \leq i < j < \omega\}$ ;
- набір операцій відображення  $\{Refl_j^i : 1 \leq i < j < \omega\}$ .

Ці операції «конструюють» логічно складні відношення з екземплярів і менш складні відношення у домені  $D$ . Зокрема, для всіх  $n$ :

- $Pred_{i_1 \dots i_k}^n : R_n \times D^k \rightarrow R_{n-k}^c (1 \leq i_1 < \dots < i_k \leq n; k \geq 1)$ ;
- $Neg : R_n \rightarrow R_n^c$ ;
- $Impl : R_n \times R_m \rightarrow R_{n+m}^c$ ;
- $Univ_{i_1 \dots i_k} : R_n \rightarrow R_{n-k}^c$ , для  $k \leq n$ ;
- $Conv_j^i : R_n \rightarrow R_n^c (1 \leq i < j \leq n)$ ;
- $Refl_j^i : R_n \rightarrow R_{n-1}^c (1 \leq i < j \leq n)$ ;

Обумовлюється, що  $Pred^n$  (тобто  $Pred_{i_1 \dots i_k}^n$  для  $k = 0$  є просто відношенням тотожності на  $R_n$  і що  $R_c$  є просто об'єднанням діапазонів логічних операцій, тобто  $R_c = \{\cup Range(f) : f \in \cup Op\}$ .

Щоб забезпечити цілісність, передбачається, що всі ці операції є типу «один-до-одного», а діапазони всіх операцій попарно не перетинаються — подібно до їхніх синтаксичних аналогів, «логічних форм» зв'язків, утворених із цих операцій всі відрізняються один від одного.

Нехай  $D_n$  буде множиною всіх  $n$ -кортежів над  $D$  і  $D^* = \bigcup_{0 < n < \omega} D^n$ . Параметр  $ext$  в кортежі представляє собою таку функцію на  $R \times W$ , що для всіх  $r \in R_n, w \in W, ext(r; w) \subseteq D^n$ . Потрібно зауважити, що для  $r \in R^0$  можливі тільки два розширення:  $\{\langle \rangle\}$ , тобто сам  $D^0$ , і порожня множина  $\emptyset$ . У цьому випадку корисно розглядати перше як значення істини, а друге як значення хибності.

Поведінка  $ext$  додатково обмежується логічними операціями в  $\cup Op$ . Деякі умовні позначення будуть корисними для формулювання цих обмежень. Для  $A \subseteq D^n$  нехай  $\bar{A}$  – це  $D^n - A$ ,  $s \frown s'$  – конкатенація двох послідовностей (кортежів)  $s, s'$ , для підмножин  $A, B$  з  $D_n$  і  $D_m$  відповідно. Нехай  $A \frown B = \{a \frown b : a \in A; b \in B\}$ , де  $1 \leq i_1 < \dots < i_o \leq n$ . Припустимо, що  $\langle b_1; \dots; b_n \rangle_{a_1 \dots a_j}^{i_1 \dots i_j}$  є результатом заміни кожного  $b_{i_k}$  на  $a_k$ , і нехай  $\langle b_1; \dots; b_n \rangle^{i_1 \dots i_j}$  буде результатом видалення кожного  $b_{i_k}$  з  $\langle b_1; \dots; b_n \rangle$ . Зважаючи на це, нехай  $r \in R_n; q \in R_m$ , тоді:

- $ext(Pred_{i_1 \dots i_k}^n(r, a_1, \dots, a_k), \omega) = \left\{ \langle b_1; \dots; b_n \rangle_{a_1 \dots a_k}^{i_1 \dots i_k}; \langle b_1; \dots; b_n \rangle_{a_1 \dots a_k}^{i_1 \dots i_k} \right\} \in ext(r, \omega);$
- $ext(Neg(r), \omega) = \overline{ext(r, \omega)};$
- $ext(Impl(q, r), \omega) = \overline{ext(q, \omega)} \frown D^n \cup D^m \frown ext(r, \omega);$
- $ext(Univ_{i_1 \dots i_j}(r), \omega) = \{ \langle a_1; \dots; a_n \rangle_{b_1 \dots b_j}^{i_1 \dots i_j} : \forall b_1; \dots; b_j \in dom(\omega), \langle a_1; \dots; a_n \rangle_{b_1 \dots b_j}^{i_1 \dots i_j} \in ext(r, \omega) \};$
- $ext(Refl_j^i(r), \omega) = \{ \langle a_1, \dots, a_{i-1}, a_j, \dots, a_{j-1}, a_i, \dots, a_n \rangle : \langle a_1; \dots; a_n \rangle \in ext(r, \omega) \text{ та } a_i = a_j \};$
- $ext(Conv_j^i(r), \omega) = \{ \langle a_1, \dots, a_{i-1}, a_j, \dots, a_{j-1}, a_i, \dots, a_n \rangle : \langle a_1; \dots; a_n \rangle \in ext(r, \omega) \};$

Логічно прості відношення розглядаються як значення примітивних предикатів в онтології. Функції предикатів, які застосовуються до примітивних відношень та екземплярів, формують базові атомарні

відношення, зокрема, базові атомарні пропозиції, а решта логічних операцій, які застосовуються до них, дають логічно складні відношення.

Вони, у свою чергу, можуть бути аргументами для подальшого застосування логічних операцій, створюючи «ітераційну ієрархію» зв'язків дедалі більшої складності. Отже, інтуїтивно зрозуміло, що зв'язки в  $R$  є або примітивними, або «побудовані» з атомарних та інших зв'язків за допомогою логічних операцій, і спосіб, у який такий зв'язок будується, можна розглядати як його логічну форму.

Так, приклад пропозиції (2.5)  $[\forall x(Planet(x) \rightarrow Larger(sun; x))]$  свідчить про те, що Сонце більше за будь-яку планету, буде побудовано з властивості бути планетою, 2-місного відношення «бути більшим, ніж», і сонце.  $Pred_1^2$  застосовано до відношення «більшого, ніж сонце», що формує властивість

$$[\lambda y Larger(sun; y)] \quad (2.6)$$

Логічний оператор  $Impl$  застосовано до властивості бути планетою та (2.6) дає співвідношення (7)

$$[\lambda xy Planet(x) \rightarrow Larger(sun; y)] \quad (2.7)$$

Це означає, що  $a$  формує  $b$  на випадок, якщо  $a$  не є планетою або сонце більше за  $b$ . Операція відображення  $Refl_1^2$ , застосована до (2.7), «згортає» свої два аргументи в один, щоб отримати властивість

$$[\lambda x Планета(x) \rightarrow Larger(sun; x)] \quad (2.8)$$

Це означає, що об'єкт має бути чимось таким, що якщо це планета, то Сонце більше за неї. Нарешті, застосування оператора «кількісної оцінки»  $Univ_1$  дає бажане твердження

$$Univ_1(Refl_1^2(Impl(Planet; Pred_1^2(Larger; Sun)))) \quad (2.9)$$

Отже, в одному рівнянні використано усі елементи моделі та визначено у який спосіб будується зв'язок з екземплярів та інших сутностей, які можна розглядати як його логічну форму. Можна трансформувати цю ідею наступним чином. Наприклад, структурне дерево для елемента  $r \in R$  є будь-яким позначеним впорядкованим деревом  $T$ , вузли якого знаходяться в  $D$ , а кореневим вузлом є  $r$ , таким чином, що для кожного вузла  $e$  з  $T$  існують дочірні вузли  $e_1; \dots; e_j$  з  $e$  такі, що для деякої операції  $F \in \cup Op$ ,  $F(e_1, \dots, e_j) = e$ .

Дерево конститутивності  $T$  для  $r$  є повним, якщо кожен листовий вузол  $o$  у  $T$  є індивідуальним або примітивним відношенням, тобто якщо  $o$  належить  $A \cup R^p$ .

Враховуючи обмеження на логічні операції, легко показати, що кожне  $r \in R$  має рівно одне повне дерево складу, яке можна ідентифікувати з логічною формою  $r$ . Тут визначається об'єкт  $o \in D$  як складова відношення  $e$ , що є вузлом у повному дереві для  $r$ . У цьому випадку,  $o$  є примітивною складовою  $r$ , якщо  $o$  є складовою  $r$  і  $o \in A \cup R^p$ . Поняття конституентності є важливим для визначення поняття онтологічного змісту.

## 2.5. Архітектура онтології управління комп'ютерними мережами

Онтологія, яка проектується у кваліфікаційній роботі, не охоплює всі аспекти в області побудови комп'ютерних мереж, а представляє знання предметної області з точки зору управління, намагаючись охопити ті аспекти мереж, які є важливими для цілей моніторингу та контролю.

Описані знання потім можуть використовуватися програмно-апаратними засобами керування мережею для діагностики та відновлення несправностей, аналізу та оцінки продуктивності, планування дій,

спрямованих на покращення якості обслуговування. Для забезпечення виконання цих завдань, онтологія повинна:

- представляти всі відношення, які існують між поняттями домену;
- визначати явні причинно-наслідкові зв'язки, за допомогою яких різні події, що відбуваються в різні часові моменти та просторові місця, можуть бути пов'язані один з одним;
- представляти інтегрований аналіз поведінки мережі;
- фіксувати, як різні елементи мережі можуть впливати один на одного для встановлення їх глобального статусу і їх динамічної еволюції у часі.

Такий опис онтологічних знань забезпечує глибоке розуміння комп'ютерних мереж в цілому, представляючи високорівневу точку зору на їх функціонування.

Більш конкретно, концепти та зв'язки, представлені в онтології, описують основні компоненти, особливості та поведінку мережевого рівня, деяких елементів каналу передачі даних мережі інтернет та фізичного рівнів.

Крім цих аспектів, пропонована онтологія повинна забезпечувати опис концепції трафіку, тип задіяних ресурсів та їх розподіл по мережі. Онтологія повинна забезпечувати опис інструментів і послуг, які необхідні для управління мережею при виконанні завдань моніторингу та контролю, а також механізми, за допомогою яких фахівець може їх використовувати. Оскільки сама система управління належить до домену представлення, остання частина онтології повинна характеризувати розподілену структуру, інструменти і послуги, які пропонуються додаткам керування.

При побудові онтології необхідно реалізувати систему, засновану на знаннях, яка діє як керуючий об'єкт у запропонованій в подальшому архітектурі – Logical Reasoner. Це дозволить чітко показати, як використовуються знання та розподілена підтримка при виконанні завдань. Включення всіх цих концепцій в онтологію дозволяє зробити її більш

повною, забезпечувати підтримку діяльності з розробки системи, яка становить одну з основних цілей онтології.

Для формалізації онтології управління комп'ютерними мережами використовується експресивна мова веб-онтології (OWL). У зв'язку з цим, онтологія, яка пропонується у кваліфікаційній роботі представляє собою OWL DL, тобто вона визначена за допомогою версії OWL DL серед трьох підмов OWL.

OWL DL містить обмеження у порівнянні з версією OWL Full, яка є найбільш повною (але з цієї причини невизначеною) підмовою. Однак OWL DL є оптимальним рішенням, оскільки інша версія OWL Lite відноситься до найпростіших підмов OWL.

Перед детальним описом запропонованої онтології варто коротко представити, які примітиви моделювання OWL DL використовуються для представлення знань. Як і кожна онтологія OWL, онтологія управління комп'ютерними мережами структурована відповідно до ієрархічної таксономії класів, що представляють основні поняття домену.

Коренем кожної онтологічної концепції є загальний клас «*Thing*», який не має спеціальних ознак; всі інші поняття походять від нього. Кожне поняття описується та специфікується набором властивостей відповідного класу. Властивості можуть бути використані для представлення як особливостей, що характеризують саму концепцію, так і асоціацій, що пов'язують концепцію з іншими концепціями домену: перші називаються «*DatatypeProperties*», а другі – визначеними «*ObjectProperties*».

«*DatatypeProperties*» може приймати різні типи значень, наприклад ціле число, число з плаваючою точкою, логічні значення, рядок тощо. Клас, до якого властивість відноситься, представляє «домен» властивості, тоді як клас або комбінація класів, над якими властивість може приймати значення, це її «діапазон». Крім того, властивості можуть бути структуровані в ієрархії підвластивостей, можуть мати певні особливості (вони можуть бути



симетричними, транзитивними, функціональними) і можуть бути пов'язані один з одним зворотними зв'язками.

Властивості, приєднані до найзагальніших класів, успадковуються їхніми підкласами та мають той самий домен і діапазон. Обмеження використовуються для подальшого визначення властивості в межах певного класу, змушуючи властивість приймати деякі або всі значення в обмеженій підмножині свого загального діапазону або приймати конкретне значення. Кількість окремих значень, які може приймати властивість, визначається обмеженнями кількості, які також мають область класу. Іншою особливістю мови є можливість визначати класи як довільні булеві комбінації (об'єднання, перетин, доповнення, диз'юнкція) інших класів і оголошувати еквівалентність класів і еквівалентність властивостей. Використовуючи описані вище екземпляри та переваги OWL, розроблена онтологія організована в ієрархічну структуру. Кореневі класи представляють найбільш загальні та, отже, найбільш багаторазово використовувані концепції предметної області та описують знання на найвищому рівні абстрагування.

Досліджуючи ієрархію від кореня до листків, концепції втрачають загальність, щоб найбільш відповідно адаптуватись до конкретних середовищ і архітектур. На рис. 2.1 показано повний вигляд ієрархічного дерева онтології «is-a». Тут показано лише перші рівні ієрархії, включаючи всі прямі підкласи «*Thing*» і деякі їх підкласи. Чорні стрілки всередині класу вказують на те, що низхідне піддерево приховано.

Структурні компоненти мережі згруповані в класі «*NetEntity*». Ці компоненти включають фізичні елементи, що утворюють комунікаційну інфраструктуру (маршрутизатори, хости, канали зв'язку та інтерфейси), а також деякі програмні компоненти, такі як таблиці маршрутизації та черги, пов'язані з інтерфейсами вузлів.

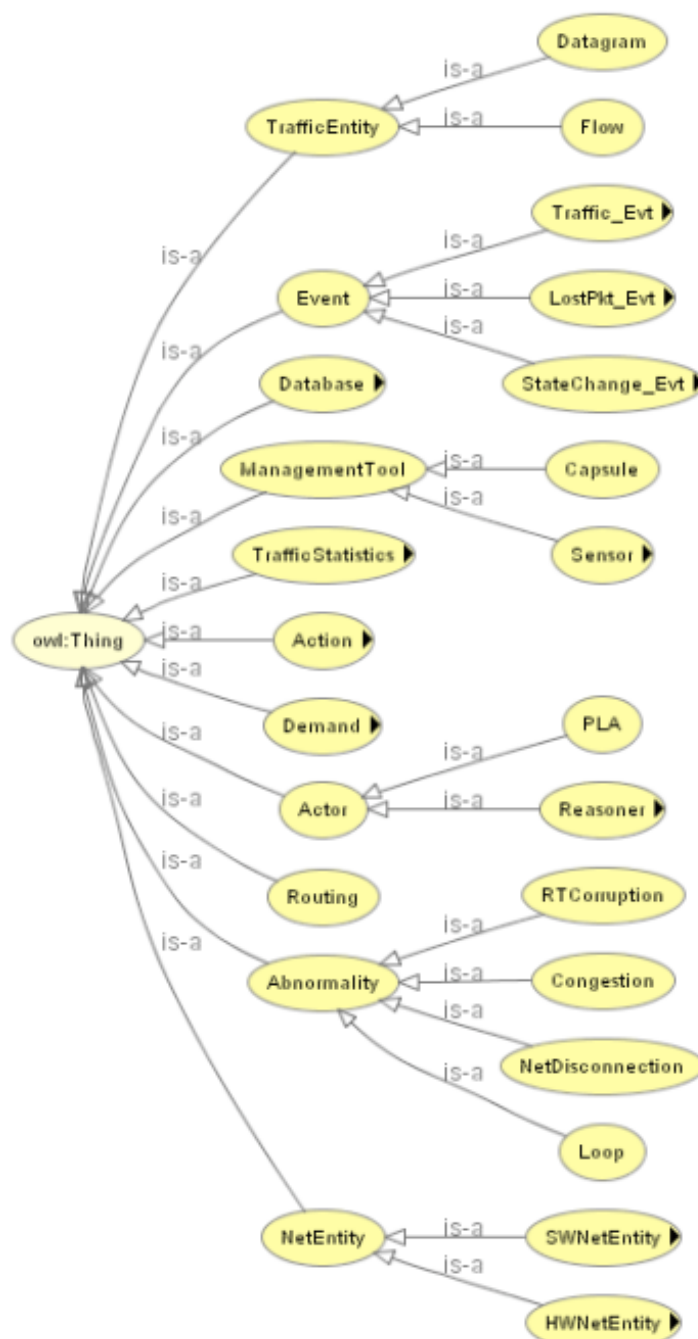


Рис. 2.1. Ієрархічне дерево онтології «is-a»

Трафік, тобто дані, що проходять по мережі та передаються мережевими об'єктами, представлений класом «*TrafficEntity*». Він розглядається на різних рівнях абстракції та має прямі посилання на ресурси, які він використовує. Функціонування елементів мережі та використання трафіку мережеских ресурсів фіксуються властивостями стану, значення яких залежать від часу.

Глобальний статус мережі також визначається загальними та абстрактними поняттями, такими як «*Routing*» та «*Demand*», які окреслюють набір взаємозалежностей, які сприяють визначенню загальної поведінки мережі.

Динамічна еволюція стану мережі та розподілу трафіку враховується концептом «*Events*», який має точне часове та просторове розташування та пов'язана з функціями, на які вона впливає.

«*Events*» (локальні «*Sensors*», спеціальні «*ManagementTools*») можуть бути частиною природної поведінки мережі або симптомом несправностей і аномалій; тому вони необхідні програмі керування, щоб підтримувати актуальне представлення мережі під час її експлуатації, а також для виявлення несправностей і визначення їх першопричин.

Для представлення загальних причинно-наслідкових зв'язків, які пов'язують події *Events* з їх причинами, було введено поняття аномалій «*Abnormality*»; представлення цих асоціацій робить процес логічного міркування здатним до злиття вхідних події з представленням мережі та високим рівнем знань, щоб зробити висновок про реальну присутність аномалії «*Abnormality*».

Решта концепцій онтології в основному пов'язані з елементами інфраструктури управління: вони представляються, як «*Reasoner*», що розглядається як актор, який може взаємодіяти з мережею за допомогою інших «*Actors*» – програмовані локальні агенти (клас PLA). Ці взаємодії описують механізми, за допомогою яких «*Reasoner*» може отримувати дані з мережі, налаштовувати систему моніторингу, активно змінювати статус і поведінку мережі, і представлені концепцією дії «*Action*». Усі ці дії дозволяють «*Reasoner*» діагностувати «*Abnormality*» та виконувати статистичні оцінки ефективності за допомогою «*TrafficStatistics*».

## 2.6. Висновки до розділу

Основні результати даного розділу полягають в наступному:

1. Запропоновано та обгрунтовано застосування математичної моделі для опису онтологій комп'ютерних мереж за допомогою кортежа, який включає в себе п'ять елементів: домен, можливі ситуації або стани системи, операції, функція відображення сутностей домену та поведінка об'єктів, що дають змогу забезпечити повноту та ефективність при описі процесів управління та моніторингу ресурсів комп'ютерної мережі.

2. Визначено лексикон, граматику та синтаксичні обмеження утворення термів при побудові онтології комп'ютерної мережі, що дало змогу формувати складні висловлювання на основі предикатів та забезпечити гнучкість моделювання процесів, які протікають у комп'ютерній мережі.

3. Запропоновано архітектуру онтології для представлення об'єктів, відношень та обмежень в комп'ютерних мережах, що дало змогу забезпечити та змодельовати домен для опису процесів управління та моніторингу.

## РОЗДІЛ 3

### РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ ПРОЦЕСІВ УПРАВЛІННЯ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ

#### 3.1. Побудова комунікаційної інфраструктури

Запропонована онтологія містить всі ті елементи комунікаційної інфраструктури, які необхідні для визначення структури та функціонування рівня мережі Інтернет; крім того, він включає деякі структурні аспекти нижніх рівнів.

Представлені концепти дозволяють визначити топологію мережі та описати основні аспекти функціональних можливостей передачі даних, таких як маршрутизація та розподіл ресурсів для управління трафіком.

Усі ці поняття визначають «керовані об'єкти», характеристики та параметри функціонування яких можна відстежувати та контролювати. Усі апаратні та програмні елементи, що утворюють комунікаційну інфраструктуру, моделюються як «*NetEntities*».

Концепт «*NetEntity*» визначає кожен елемент, який бере участь у функціонуванні мережі, забезпечуючи виконання певних завдань. Загальний клас спеціалізується на двох основних підкласах: апаратні сутності (представлені класом «*HWNetEntity*») і програмні сутності (представлені класом «*SWNetEntity*»). Перші є фізичними компонентами мережі, тоді як другі є тими нефізичними елементами, які пояснюють поведінку мережі, як, наприклад, розподіл ресурсів.

Повна ієрархія концепту «*NetEntit*» у показана на рис. 3.1.

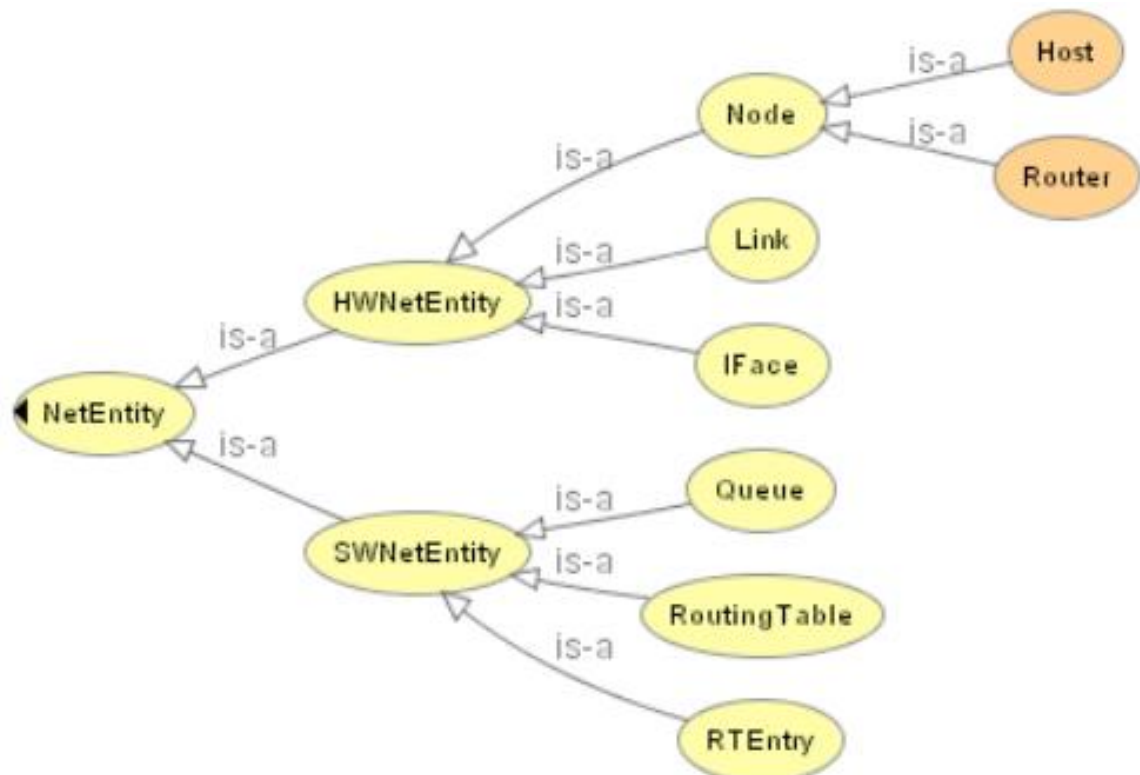


Рис. 3.1. Ієрархія концепту *NetEntity*

Як видно з рис. 3.2 клас підклас «*HWNetEntity*» містить наступні класи:

- *Node*, який в свою чергу поділений на два підкласи «*Router*» і «*Host*»;
- «*IFace*», який представляє мережеві плати, які є інтерфейсними вузлами для решти компонентів мережі;
- «*Link*» – представляє фізичні двонаправлені зв'язки, які з'єднують вузли мережі за допомогою їх інтерфейсів.

Усі ці сутності мають «*HWStatus*», тобто «*DatatypeProperty*», що визначає їхній статус функціонування. Ця властивість може приймати одне з трьох різних значень: «ON», «OFF» і «ABN».

Інформація про статус «ON» вказує на те, що об'єкт активний і функціонує належним чином. Статус «OFF» означає, що сутність навіть якщо вимкнено, то її можна контролювати та активувати за потреби. Статус

«ABN» вказує на те, що об'єкт вийшов з ладу або працює несправно, і ним неможливо керувати безпосередньо.

Наприклад, якщо стан зв'язку має значення «ABN», це означає, що зв'язку не існує, його не можна використовувати для передачі даних і неможливо віддалено повторно активувати до того, як дія прямого відновлення відновить його нормальний статус.

Для визначення топології мережі, «*HWNetEntities*» пов'язані між собою за допомогою «*ObjectProperties*», що представляє їхні фізичні асоціації та взаємозв'язки, як показано на рис. 3.2.

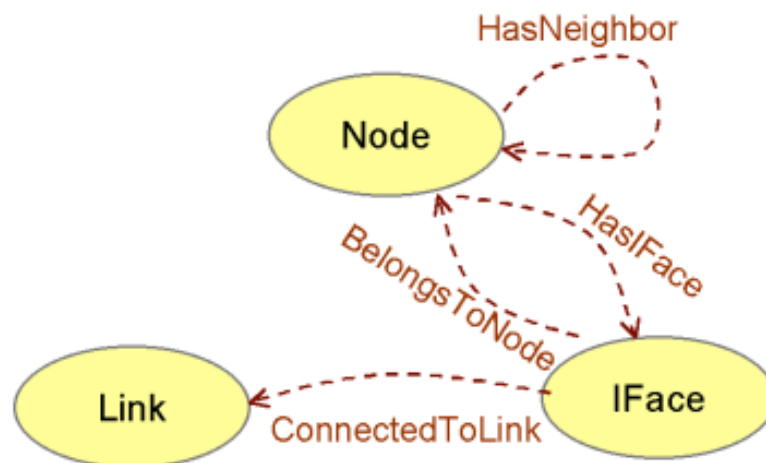


Рис. 3.2. Асоціації між класами на основі *ObjectProperties*

Наприклад, щоб представити зв'язок між інтерфейсом і вузлом, на якому він встановлений, використовуються дві властивості: властивість «*HasIFace*» і властивість «*BelongsToNode*». Перша властивість оголошена у класі «*Node*» і має кілька значень у класі «*IFace*»; друга властивість, навпаки, оголошується в класі «*IFace*» і може приймати одне значення в класі «*Node*». Кожна з цих двох властивостей є зворотною до іншої, тому вони створюють двонаправлене посилання між двома класами. Подібним чином «*IFace*» пов'язані з «*Link*» через властивість «*ConnectedToLink*», яка дозволяє визначити, як вузли, до яких вони належать, пов'язані між собою за допомогою фізичних зв'язків. Властивість «*ConnectedToLink*», зазвичай,

може приймати кілька значень. Незважаючи на те, що дані переміщуються з одного інтерфейсу до іншого за допомогою одного зв'язку, запропонована модель також дозволяє відображати резервні зв'язки. Резервні зв'язки (посилання) можна вчасно активувати, щоб відновити підключення після розриву з'єднання; багатофункціональність властивості «*ConnectedToLink*» забезпечує заміну «звичайного» зв'язку резервним і не впливає на логічне з'єднання між двома вузлами. У такий спосіб змінюється лише значення «*ConnectedToLink*», тоді як два вузли повторно з'єднуються за допомогою тих самих інтерфейсів (та IP-адрес). Щоб більш чітко обмежити ці концепти, до класу «*Links*» додано логічну властивість «*Backup*» для представлення кожного типу зв'язку.

Логічний зв'язок між двома сусідніми вузлами було представлено явно за допомогою властивості, яка була додана до класу «*Node*»; «*HasNeighbor*» є симетричною властивістю (тобто якщо А є сусідом В, це істинним є і зворотне твердження) і він пов'язує члени класу «*Node*» з іншими членами того самого класу. Внутрішні обмеження OWL не дозволяють зв'язати властивість «*HasNeighbor*» з іншою інформацією про топологію, таким чином обмежуючи представлення знань.

На рис. 3.3 показано приклад «*HWNetEntity*» екземплярів, які пов'язані між собою вище описаними властивостями, і визначають з'єднання між двома вузлами.

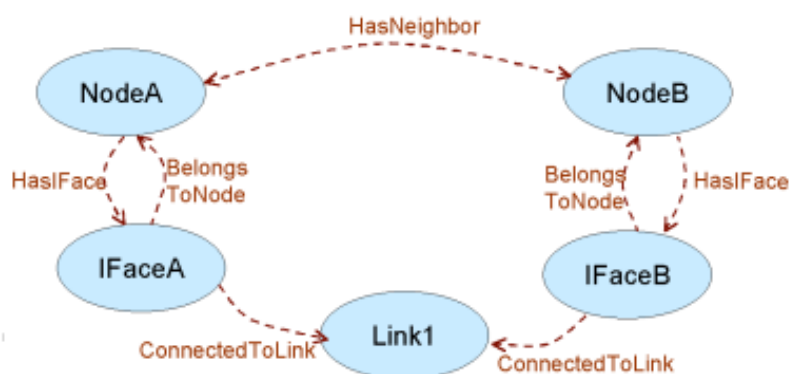


Рис. 3.3. Приклад взаємодії екземплярів класу «*HWNetEntity*»



Як видно з рис. 3.2 та рис. 3.2, «*HWNetEntities*» показує, що кожен клас володіє кількома властивостями, які більш точно визначають їх природу та функціонування. Наприклад, «*IFaces*» унікально ідентифікуються своєю адресою та містять посилання на пов'язану з ними чергу, де зберігаються пакети, що очікують на обробку або передачу.

«*Links*» моделюються дуже детально, оскільки вони важливі для моніторингу розподілу ресурсів і трафіку, а також для покращення продуктивності мережі. Вони мають властивості (такі як «*Technology*», «*BandWidth*» тощо), що описують їхні фізичні характеристики, і вартість *Cost*, яка представляє інформацію про вартість маршрутизації усією мережею у конкретний момент часу.

Інші важливі властивості представляють використання зв'язку, пов'язуючи канали з потоками трафіку, які вони передають. Більш детально ці властивості будуть описані в наступних пунктах кваліфікаційної роботи, коли концепти трафіку стануть більш зрозумілими, а подання розподілу трафіку буде вичерпно досліджено.

Вузли мережі формують каталоги у вигляді хостів («*Hosts*») та маршрутизаторів («*Routers*») (рис. 2.1), що відрізняються за кількістю інтерфейсів: перші – це всі ті вузли, які мають більше одного «*IFace*» (тобто де властивість «*HasIFace*» має мінімальну потужність, рівну двом), тоді як хости мають тільки один. Усі вузли, маршрутизатори або хости виконують функції маршрутизації та потребують інформації про переміщення для їх виконання. Це найважливіший аспект функціонування з точки зору управління, і він формалізується властивістю «*HasRoutingTable*», яка пов'язує кожен вузол із власною «*RoutingTable*» (рис. 3.4).

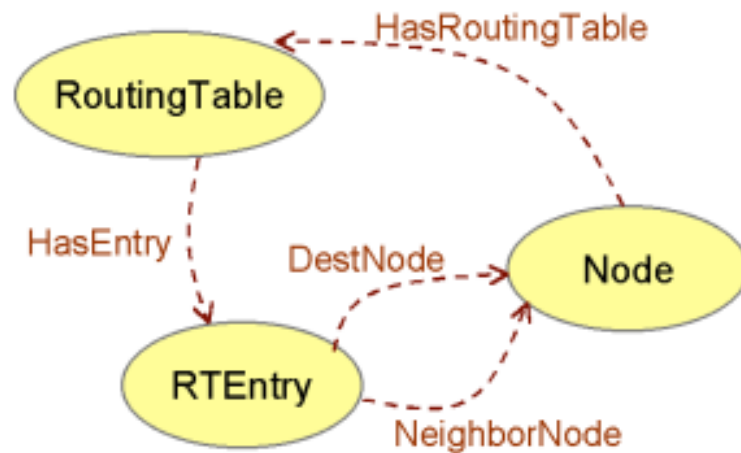


Рис. 3.4. Властивості об'єктів, якими володіють класи «*Node*», «*RoutingTable*», «*RTEntry*»

Клас *RoutingTable* є підкласом більш загального класу «*SWNetEntity*» (рис. 2.1). Концепт «*SWNetEntity*» моделює компоненти програмного забезпечення, які відіграють центральну роль у функціонуванні мережі та важливі для моніторингу.

Клас «*SWNetEntity*» поділяється на три підкласи: окрім класу «*RoutingTable*», є клас «*RTEntry*» і клас «*Queue*». Перші два класи, «*RoutingTable*» і «*RTEntry*», використовуються для опису інформації маршрутизації, наявної на вузлі. «*RoutingTables*» визначаються як агрегації екземплярів класу «*RTEntry*» через властивість об'єкта «*HasEntry*». Члени класу «*HasEntry*» представляють рядки таблиці маршрутизації та містять інформацію, необхідну для пересилання пакетів до кожного адресата; вони мають явні посилання на вузол призначення, якого вони стосуються (властивість «*DestNode*»), і на сусідній вузол (властивість «*NeighborNode*»), до якого пересилаються пакети, щоб досягти пункту призначення. Цей конкретний вибір представлення, використовуючи клас для представлення одного запису таблиці маршрутизації, зробив можливим виразити онтологічний зв'язок, що існує між інформацією маршрутизації та фізичними вузлами мережі, на які вона посилається (рис. 3.3).

Вартість шляху маршрутизації до пункту призначення також представлено за допомогою властивості «*Cost*» у «*DatatypeProperty*». Щоб надати інформацію про послідовність і правильність інформації маршрутизації вузла, статус «*DatatypeProperty*» – «*RTStatus*», приєднується до класу «*RoutingTable*». Ця властивість може приймати одне з двох значень: «*Normal*» і «*Corrupted*», де друге означає відсутність запису або ненормальний стан.

Нестандартні записи можуть спричинити пошкодження бітів або нескінченну вартість, і обидва випадки унеможливають маршрутизацію для пов'язаних пунктів призначення.

Третім підкласом «*SWNetEntity*» є клас «*Queue*» (рис. 2.1). Цей концепт використовується для представлення буферів, де тимчасово зберігаються пакети, що очікують на обробку або передачу. Для окремого моделювання двох випадків було введено властивість «*QueueType*» з одним із двох значень «*IN*» або «*OUT*». Черги пов'язані з «*IFace*» властивістю «*AssociatedToIFace*», тоді як у класі «*IFace*» є дві різні функціональні властивості, «*HasInQueue*» і «*HasOutQueue*», що означає, що кожен «*IFace*» має рівно одну чергу типу «*IN*» і рівно одну чергу типу «*OUT*».

Черги разом із посиланнями представляють мережеві ресурси, зайнятість яких є важливим параметром у вимірюванні навантаження трафіку та продуктивності мережі. Зокрема, високі або середні рівні зайнятості в чергах можуть бути симптомом нестачі обчислювальних ресурсів вузла щодо фактичного навантаження трафіку, тоді як високі рівні зайнятості поза чергою можуть вказувати на перевантаження каналу. Різні значення переповнень у черзі та поза чергою є причиною їх точного моделювання.

Через свою важливість вони представляють властивості, що визначають їхній загальний розмір і відсоток використання, а також чіткі посилання на дані трафіку, які вони використовують.

### 3.2. Моделювання трафіку та його розподілу у комп'ютерній мережі

Трафік є важливим компонентом представлення мережі, оскільки основна функція, яку мережа пропонує своїм користувачам, полягає в переміщенні даних з одного місця в інше.

Управління мережею має справу з моніторингом та оцінкою параметрів функціонування для того, щоб зрозуміти, чи відповідає мережа вимогам ефективності та надійності та наскільки добре, виконуючи вищезазначене завдання.

Таким чином, неможливо побудувати онтологію мережі без концепції трафіку. Кореневим класом усіх концепцій трафіку є клас «*TrafficEntity*» (рис. 3.5), який представляє всі дані, що проходять через усю мережу. Хоча дані можна розглядати як «програмні» елементи, вони відрізняються від концептів, представлених класом «*SWNetEntity*»: а саме, тоді як «*SWNetEntities*» є частиною мережевої комунікаційної інфраструктури, яка активно сприяє її функціонуванню та бере участь в управлінні трафіком, «*TrafficEntities*» представляють різницю трафіку у мережі та використання її інфраструктури, ресурсів і здатності переміщувати дані.

Загальна концепція «*TrafficEntity*» спеціалізується за двома підкласами, «*Datagram*» і «*Flow*», що представляють дані на різних рівнях абстракції (рис. 3.5).

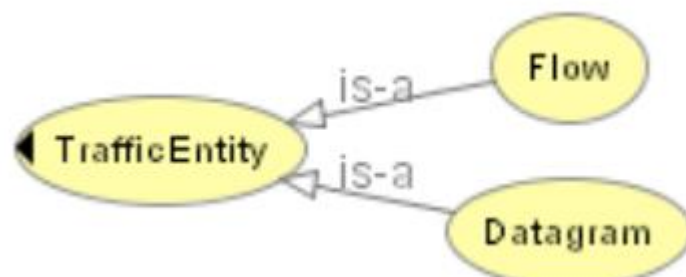


Рис. 3.5. Клас опису трафіку комп'ютерної мережі

Перший ідентифікує одну IP-датаграму, яка є атомарною одиницею трафіку, керованою мережевим рівнем, і має властивості, що представляють її типові характеристики, такі як розмір, параметри джерела та призначення (властивості «*Length*», «*SourceNode*» і «*DestNode*»).

Клас «*Flow*» визначає трафік на вищому рівні, агрегуючи датаграми, що проходять уздовж того самого шляху маршрутизації та, таким чином, спільно використовують ті самі параметри джерела та призначення.

Клас «*Flow*» володіє властивістю «*HasDatagram*», він приймає кілька значень класу «*Datagram*» і визначає зв'язок із набором одиниць, що утворюють потік. Крім того, клас «*Flow*» має «*SourceNode*» і властивості «*DestNode*», але, через обмеження OWL, обмеження між їхніми значеннями та відповідними значеннями в датаграмах, що складають потік, має залишатися неявним.

Щоб розташувати концепт потоку у часовому вимірі, дві властивості типу даних «*StartTime*» і «*EndTime*», були додані до класу, що визначає інтервал часу, протягом якого існує потік.

Потік описується як безперервний потік даних, хоча він складається з окремих елементів (тобто датаграм); дві датаграми, що використовують один і той самий шлях маршрутизації, можна вважати частиною одного потоку, якщо вони виникають у два досить близькі моменти часу.

Слід зауважити, що змодельована тут концепція «*Flow*» є лише пом'якшеною версією типового «мережевого з'єднання», ідентифікованого п'ятьма параметрами: «*hSourceNode*», «*DestNode*», «*SourcePort*», «*DestPort*», «*TransportProtocol*»; цей вибір був мотивований завданнями моніторингу та контролю, для яких визначена онтологія, і добре підходить для представлення лише трьох нижніх рівнів.

Грубо кажучи, це уявлення про потік фіксує лише ті параметри, які стосуються функції маршрутизації, і ті характеристики, достатні для розуміння розподілу трафіку. Щоб пояснити, як мережеві ресурси розподіляються для передачі даних, було введено спеціальні властивості, які

явно прив'язують дані трафіку до ресурсів, тобто черги та посилення, які вони використовують.

Наприклад, потоки мають двонаправлені посилення на лінки, через які вони проходять, за допомогою властивості «*OnLink*» у класі «*Flow*» і «*UsedByFlow*» у класі «*Link*». Так само черги посиляються на дейтаграми, які вони містять у певний момент через властивість «*HasDatagram*». Через важливість вимірювання використання ресурсів додаткові властивості, такі як «*UsedBandwidth*» і «*FlowBandwidth*» у класі «*Link*» або «*UsedSpace*» у класі «*Queue*», точно кількісно визначають навантаження даних. Властивість «*UsedBandwidth*» визначає загальний відсоток використаної пропускну здатності посилення, тоді як властивість «*FlowBandwidth*» визначає пропускну здатність, яку використовує один потік у бітах на секунду.

Це неявний зв'язок між тими потоками, на які посиляється властивість «*UsedByFlow*», і відповідною смугою пропускання, вираженою у властивості «*FlowBandwidth*»; прив'язка між сумою цих смуг пропускання потоку та загальною «*UsedBandwidth*» також неявна.

На рис. 3.6 показано асоціації та властивості, які використовуються для представлення розподілу трафіку.

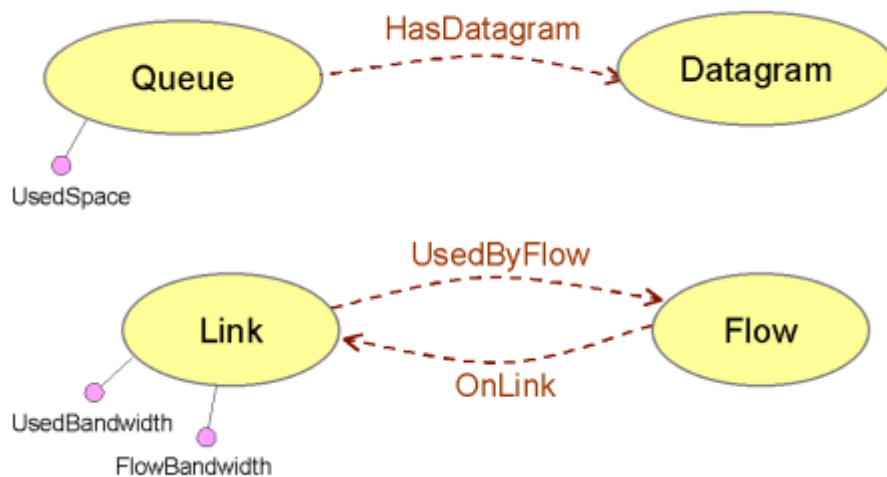


Рис. 3.6. Асоціації і властивості представлення розподілу трафіку

Класи та властивості, описані вище, визначають структурні аспекти функціонування мережі та трафіку. Тим не менш, деякі взаємозалежності не можуть бути охоплені через детальний і низькорівневий опис окремих елементів і потребують більш загального уявлення про стан глобальної мережі та розподіл трафіку. Наприклад, зв'язки, що зберігають потік, просто визначаються таблицями маршрутизації проміжних вузлів на шляху від джерела до пункту призначення, а швидкість потоку залежить від обсягу даних, які джерело намагається передати.

Крім того, потік залежить лише від цих таблиць маршрутизації або визначається лише вимогою джерела. Зокрема, швидкість потоку (і використана пропускна здатність каналів) також залежить від наявності та швидкості інших потоків, які спільно використовують ті самі канали та обчислювальні ресурси на тих самих вузлах.

Для вираження такого роду знань було введено ще два загальні класи, які формалізують абстрактні поняття маршрутизації та вимог відповідно. Хоча ці класи не мають прямого зв'язку з будь-яким мережевим елементом, вони корисні для опису навантаження та продуктивності мережі. Обидва класи є прямими підкласами «*Thing*» (рис 2.1).

Концепція маршрутизації використовується для представлення того, як маршрутизація в усій мережі визначає розподіл і використання ресурсів, таким чином впливаючи на продуктивність. Ця концепція має два основних зв'язки з іншими класами онтології: властивість «*RelatedTo*», яка зв'язує клас «*Routing*» з «*RoutingTable*», і властивість «*DependsOnRouting*», що пов'язує клас «*Flow*» з класом «*Routing*» (рис. 3.5).

Перша властивість показує, що таблиці маршрутизації вузлів конкретно представляють глобальну маршрутизацію, тоді як друга властивість представляє залежність між ресурсами, використовуваними потоком уздовж його шляху, та діяльністю маршрутизації, що виконується всіма вузлами мережі, що змушує інші потоки спільно використовувати те саме шлях.

Концепція попиту (вимог) представляє гіпотетичне навантаження трафіку, яке буде передано через мережу без обмежень ресурсів. Клас «*Demand*» володіє лише двома властивостями: «*StartTime*» і «*EndTime*», що представляють розглянутий часовий інтервал.

Клас «*Demand*» далі поділяється на два підкласи: «*DemandMatrix*» і «*DMEEntry*» (рис. 3.7), у яких кількісно визначається попит кожного джерела.

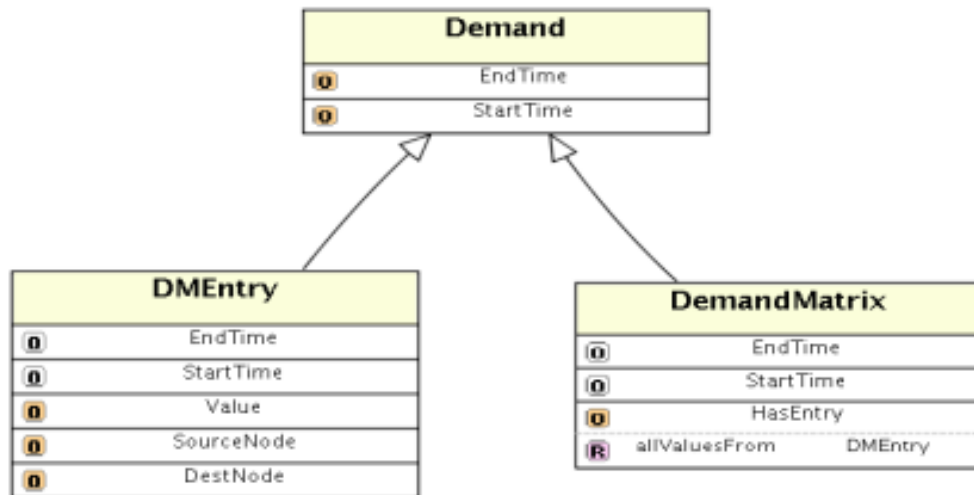


Рис. 3.7. Структура класу «*Demand*»

Клас «*DemandMatrix*» є сукупністю «*DMEEntries*», кожна з яких представляє обсяг даних (властивість *Value*), які кожне джерело хоче передати всім можливим адресатам (властивості «*SourceNode*» і «*DestNode*»). Очевидно, попит є абстрактним поняттям, але його представлення корисно, щоб чітко вказати, що він визначає реальне навантаження трафіку. Цей зв'язок виражається властивістю «*DependsOnDemand*», яка пов'язує члени класу «*Flow*» із класом «*Demand*» (рис. 3.8).





Рис. 3.8. Зв'язок між класами потоку і попиту

Побудувавши модель для аналізу трафіку і попиту вузлів, наступний етап полягає у моделюванні подій та аномалій, які можливі у комп'ютерній мережі.

### 3.3. Події та аномалії

Як уже було сказано, «*NetEntities*» визначають структуру мережі, описуючи її статус, «*TrafficEntities*» описують мережевий трафік та його розподіл за посиланнями та чергами, маршрутизація та попит описують внутрішні механізми, які генерують певний глобальний статус мережі.

Зовнішнє представлення динамічної еволюції мережі від одного стану до наступного можна зробити, ввівши концепт події.

Клас «*Event*» представляє загальну концепцію, яка використовується для визначення того, що може статися в мережі. Він має єдині властивості «*Time*» та «*OnEntity*», що вказують на момент і місце, де це відбувається.

Цей загальний концепт потім специфікується в більш конкретних класах, які моделюють різні типи подій. Ієрархія подій показана на рис. 3.9.

Клас «*Event*» має три основні підкласи, які об'єднують події схожої природи та подібним чином впливають на стан мережі: «*LostPktEvt*», «*StateChangeEvt*» і «*TrafficEvt*».

«*LostPkt Evt*s» використовуються для вказівки на те, що дейтаграму було відхилено маршрутизатором, і вона поділяється відповідно до локальної причини, яка спричинила втрату: «*RTLostPkts*» вказують на

втрата дейтаграми через неможливість пересилання під час виконання функції маршрутизації, тоді як «*TTLLostPkts*» вказують на відкидання дейтаграми через закінчення терміну дії «*TimeToLive*».

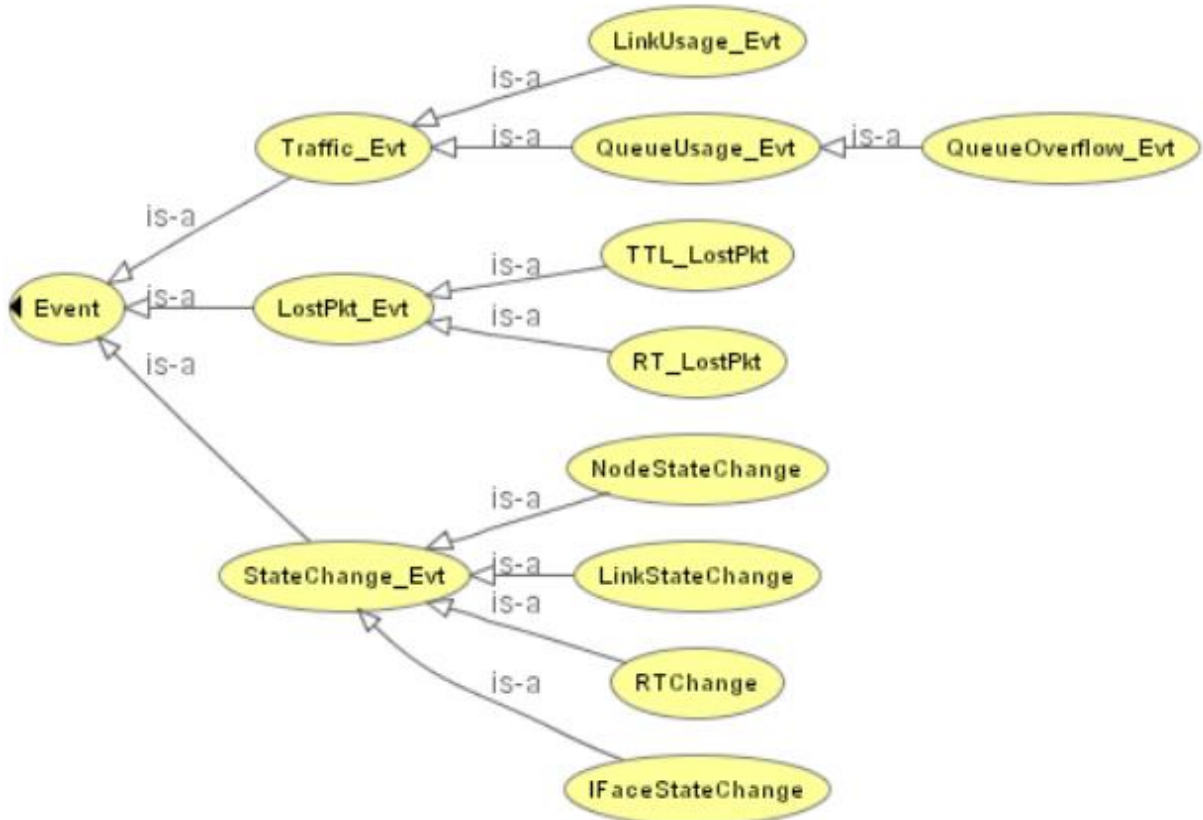


Рис. 3.9. Ієрархія класу подій

«*StateChangeEvs*» вказують на зміни статусу функціонування на апаратних або програмних об'єктах (наприклад, підвищення або зниження зв'язку або оновлення таблиці маршрутизації) і далі поділяються на «*NodeStateChange*», «*LinkStateChange*», «*IFaceStateChange*» та «*RTChange*».

«*TrafficEvt*» описує динамічний потік даних трафіку через мережеві ресурси, і вони розрізняються на «*LinkUsageEvt*» і «*QueueUsageEvt*».

Ця остання подія додатково спеціалізується на більш специфічну «*QueueOverflowEvt*». Події в зміні мережі відіграють певну роль, яка виражається властивостями, що пов'язують їх з об'єктами, де вони відбуваються, і з функціональними аспектами, на які вони впливають.

Клас «*Event*» володіє властивістю «*OnEntity*», пов'язуючи його з «*NetEntity*», де відбувається подія. Підкласи подій мають обмеження щодо типу сутності в діапазоні «*OnEntity*». Наприклад, у класі «*StateChangeEvt*» значення властивості «*OnEntity*» визначає апаратне або програмне забезпечення, стан якого змінився. У своєму підкласі «*LinkStateChange*» ця властивість обмежена значеннями лише над класом «*Link*».

Три підкласи «*StateChangeEvt*», класи «*LinkStateChange*», «*IFaceStateChange*» і «*NodeStateChange*», також мають властивість «*ChangeType*», яка може мати значення «ON» і «OFF», що вказує на те, чи вказана «*HWEntity*» зараз активна чи ні. Натомість клас «*RTStateChange*» має властивість «*ChangedEntry*», яка приймає значення над класом «*RTEntry*» і явно вказує записи, які були замінені, видалені або додані.

Події, що виходять із «*TrafficEvt*», представляють ті самі властивості, що використовувалися раніше для вказання використовуваності ресурсу, це означає, що вони оновлюють значення цієї властивості відповідно до фактичного розподілу трафіку.

Таким чином, «*LinkUsageEvt*» має властивості «*UsedByFlow*» і «*FlowBandwidth*», сповіщаючи про зміну пропускну здатності каналу, що використовується одним потоком (таким чином ми також зробили явний зв'язок між потоком і відповідною пропускну здатністю, яку він приймає на певному каналі зв'язку, відношення які не можуть бути виражені в класі «*Link*»). Подібним чином «*QueueUsageEvt*» має властивість «*UsedSpace*», яка також належить класу «*Queue*». Як приклад, рис. 3.10 показує, як зв'язок між «*QueueUsageEvt*» і функціями, на які він впливає, представлений в онтології.

Події втілюють знання, необхідні системі, заснованій на знаннях управління, щоб підтримувати в актуальному стані своє представлення мережі та виконувати завдання діагностики. Вони можуть бути частиною нормальної динаміки мережі або розглядатися як симптоми несправності.

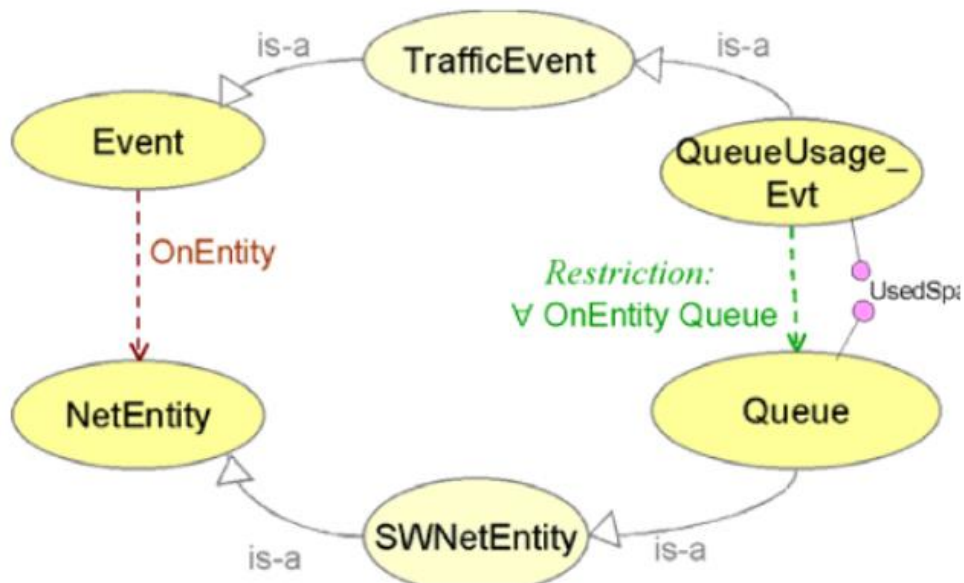


Рис. 3.10. Асоціація між класом подій і сутностей

У другому випадку основна причина таких симптомів формалізована в онтології в класі аномалій і пов'язана з подіями за загальними причинно-наслідковими зв'язками.

Аномалія пов'язана з набором подій, які вона викликає, за допомогою властивості «*Causes*», зворотної до властивості «*CausedBy*» у класі «*Event*».

Події та аномалії мають дуже різну онтологічну природу: у той час як події є миттєвими проявами зміни або події, мають точне часове та просторове розташування та можуть бути локально вловлені «спеціальними» датчиками, аномалії представляють більш глобальний тип інформації та визначають несправний стан, який може включати багато мережевих елементів, розміщених у різних місцях, може зберігатися з часом і може виявлятися в широкому діапазоні зовнішніх симптомів.

З цієї причини аномалії і події не мають зв'язків наслідування у своєму онтологічному представленні, і вони є прямими підкласами загального класу «*Thing*». Прояв множинних симптомів відображається множинною потужністю властивості «*Causes*».

Аномалії, представлені в онтології, виявлені як показано на рис. 3.11. Існує чотири підкласи аномалій: петля, відключення, «*RTCcorruption*» і

перевантаження. Кожен із цих класів має відповідні властивості, що визначають, які об'єкти беруть участь у ненормальній ситуації та яке місце розташування аномалії (наприклад, параметри джерела та призначення циклічних або роз'єднаних шляхів, вузли та зв'язки, залучені до ситуації перевантаження тощо).

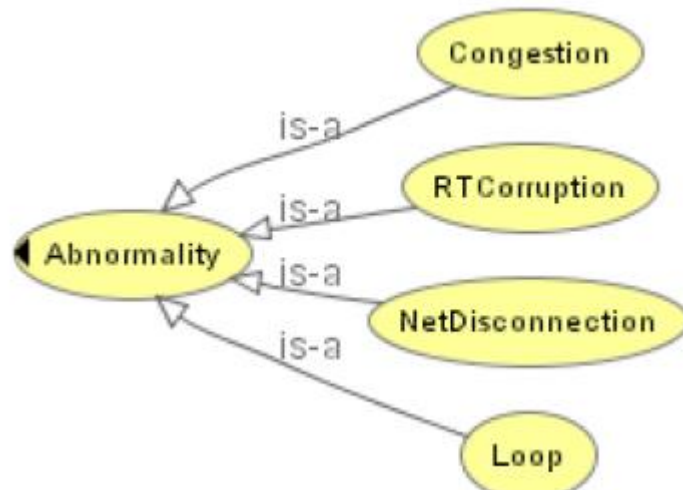


Рис. 3.11. Клас аномалій комп'ютерних мереж

Крім того, кожна аномалія явно пов'язана з її можливими симптомами через обмеження властивості «*Causes*». Також були накладені обмеження на властивість «*CausedBy*» класу «*Event*». Наприклад, через обмеження властивість у класі «*Disconnection*» змушена приймати значення лише для класу «*RTLostPkt*», тоді як подія такого роду, у свою чергу, може бути спричинена аномаліями як «*Disconnection*», так і «*RTCorruption*».

Усі обмеження, що виражають причинно-наслідкові зв'язки між підкласами подій та аномалій, показані на рис. 3.12.

Варто зауважити, що в онтологічному представленні події класифікуються не на основі їхньої нормальної чи ненормальної природи, а з огляду на зміну мережі, яку вони викликають. Ця форма представлення походить від активної інтерпретації, яка виконується процесом аналізу, який ідентифікує аномальну природу події та виявляє її першопричину аномалії. Зокрема, якщо події можна виявити локально, для виявлення

аномалій потрібен процес аналізу, який поєднує різні події з більш загальними знаннями та глобальною інформацією про стан. Наприклад, аномалію відключення, яка зазвичай викликає події «*RTLostPkt*» на залучених маршрутизаторах (де встановлено певні датчики), неможливо визначити локально; необхідний комплексний огляд стану мережі та високого рівня та високий рівень знань про можливі причини «*RTLostPkt*», щоб визначити її присутність.

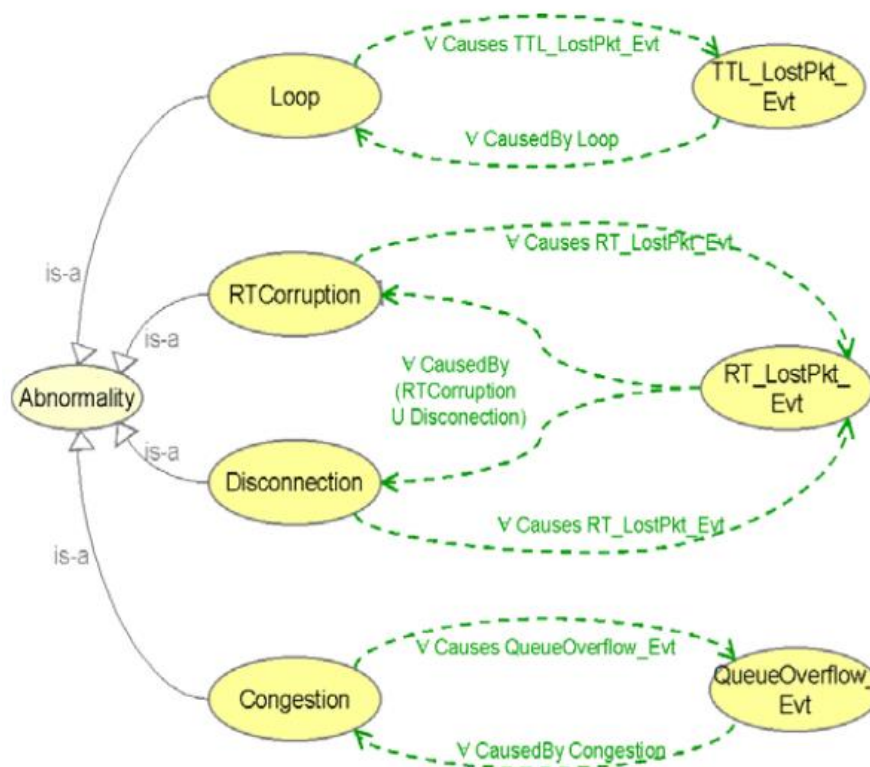


Рис. 3.12. Властивості аномалій у комп'ютерних мережах

Оскільки події є дуже важливими для виконання завдань керування, вони являють собою центральну концепцію цієї онтології та виникають як посилення в широкому наборі властивостей, що описують, як вони локально формуються та зберігаються, сповіщають програми центрального керування та використовуються для виконання діагностики несправностей і статистична оцінки продуктивності.

### 3.4. Інструменти управління та моніторингу комп'ютерної мережі

Раніше описана онтологія охоплює концептуальне визначення мережі як самодостатньої системи. Вона описує структуру, функціонування та динамічну еволюцію, охоплюючи знання високого рівня, необхідні для розуміння різних факторів, взаємозалежностей і причинно-наслідкових зв'язків, які сприяють визначенню поведінки мережі. Цей облік знань робить програму управління здатною виконувати завдання управління високого рівня.

Незважаючи на те, що «операційні знання», які така програма використовує для вирішення проблем і здійснення певних стратегій, не представлені (відповідно до цілей онтології), онтологія моделює найбільш репрезентативні елементи інфраструктури управління та найбільш репрезентативні взаємодії, які забезпечують управління. Тому онтологія не представляє, наприклад, логічний шлях, який використовує дослідник для класифікації подій і виявлення аномалій, ані критеріїв, які вона використовує для оцінки продуктивності, або міркувань і обчислень, які вона виконує, щоб прийняти рішення про відповідні втручання.

Онтологія описує базову інфраструктуру та знання, необхідні для виконання цих завдань, роблячи явним зв'язок між інструментами моніторингу та контролю та параметрами функціонування мережі, які вони відповідно спостерігають і впливають, таким чином надаючи високорівневе уявлення про те, як їх використовувати. Однак перша частина онтології залишається найбільш загальною і, отже, найбільш багаторазовою.

Далі коротко представлено архітектуру системи та опис її основних елементів, які наявні в онтології.

Система управління базується на наявності механізму логічного висновку і розподілених локальних агентів. Перший компонент діє як програма централізованого керування, тоді як другий надає підтримку для виконання завдань управління високого рівня.

Архітектурна схема базується на парадигмі активної мережі. Reasoner і PLA представляють кінцеві точки управління. Зв'язок стає можливим завдяки службі шлюзу, що надається одним із вузлів мережі.

Reasoner видає команду та отримує інформацію від ANGateway, що має завдання отримувати та надсилати активні пакети з мережі. Ці пакети можуть містити код для виконання на вузлах мережі. Рис. 3.13 демонструє архітектуру керування мережею та її основними елементами, а на рис. 3.14 показано основні класи, які використовуються онтологією для їх представлення.

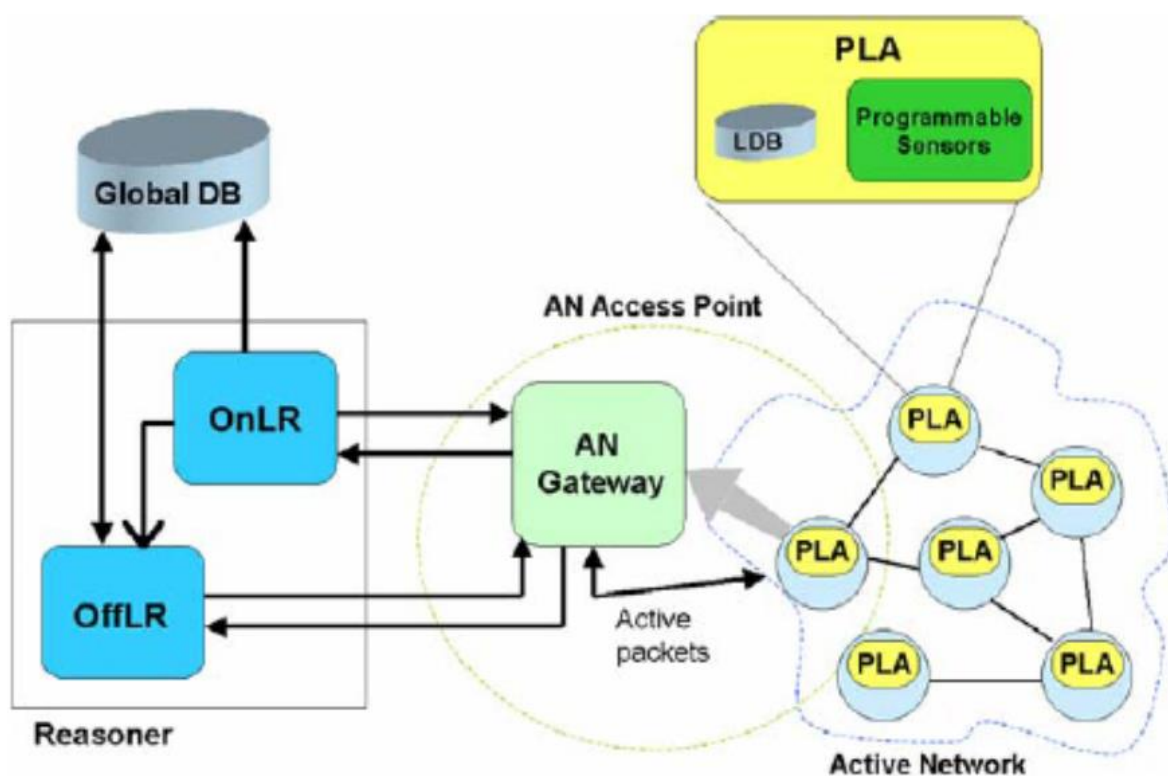


Рис. 3.13. Архітектура системи управління мережею

PLA оснащені набором датчиків, які здатні контролювати конкретні аспекти вузла, тобто конкретні параметри функціонування. Завдяки можливості мережевого програмування, наданій Active Network, Reasoner може динамічно розгортати датчики по всій мережі. Крім того, їх можна вчасно налаштувати, використовуючи служби PLA.





Рис. 3.14. Класи фреймворку управління мережею

Датчики в онтології представлені класом *Sensor*, підкласом класу *ManagementTool* (рис. 3.15), який об'єднує деякі інструменти, які керуючі об'єкти (*Reasoner* і *PLA*) можуть використовувати для виконання завдань управління.

Окрім класу *Sensor*, іншою спеціалізацією інструменту управління є клас *Capsule*, який представляє активні пакети, що містять деякий код, який буде виконано на вузлах мережі. Здатність датчика фіксувати локальні події виражається властивістю *CatchesEvent*, яка пов'язує клас *Sensor* з класом *Event*.

Клас датчиків далі поділяється на більш конкретні класи, кожен з яких представляє окремий тип датчика, який фіксує певний вид події. Крім того, деякі датчики мають додаткові властивості, які представляють параметри моніторингу, які може налаштувати. Н

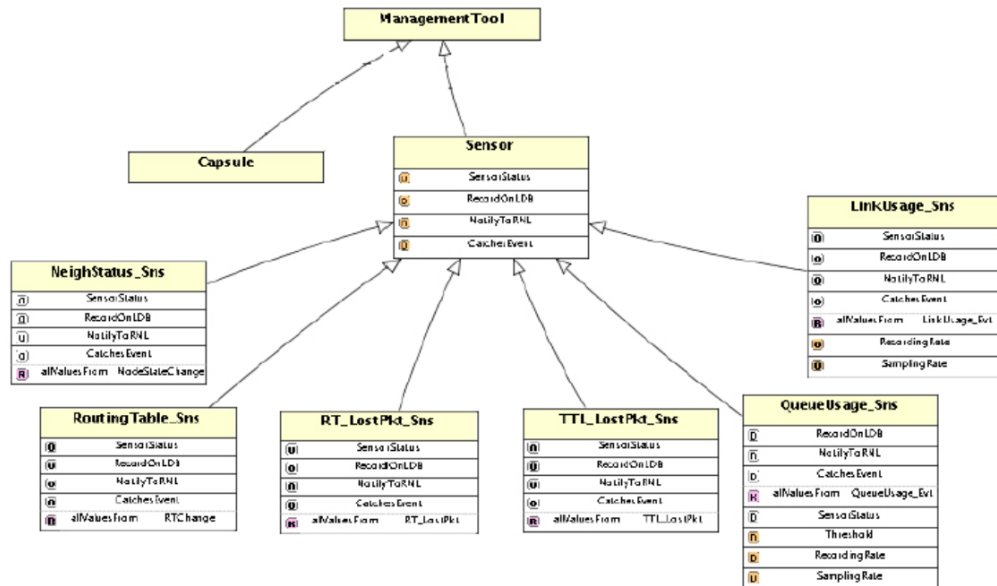


Рис. 3.15. Структура класів інструменту управління комп'ютерною мережею

Налаштування датчика полягає у зміні цих параметрів, таких як частота дискретизації, на якій відбуваються спостереження, або умови, за яких виникають події.

Наприклад, клас QueueUsage Sns має властивість SamplingRate, яка представляє частоту вибірки при яких спостерігаються черги. Натомість параметр Threshold представляє значення нагадування, вище якого має бути викликано «*QueueOverflowEvt*».

Reasoner виконує два типи процесів аналізу: реактивний онлайн-аналіз та офлайн-аналіз історії. Оскільки ці дві управлінські дії були розроблені як незалежні завдання, можна виділити два різних агенти аналізу.

OnLineReasoner (OnLR) виконує динамічне обґрунтування: він збирає інформацію в режимі реального часу про мережу, її причини, щоб зробити висновок про аномалії, і, на основі своїх висновків, своєчасно реагує, щоб налаштувати систему моніторингу та вжити заходів для усунення несправностей.

Комплексний статистичний аналіз і оцінка продуктивності є складнішими завданнями, які включають знання про історію функціонування мережі та вимагають складних міркувань. Вони виконуються OffLineReasoner (OffLR), який, у свою чергу, може запропонувати відповідні дії, спрямовані на покращення якості обслуговування. Пропозиції OffLR потім перетворюються на конкретні втручання OnLR; а саме, це єдиний агент, який ефективно виконує завдання керування та видає команди для локального виконання розподіленими агентами.

### 3.5. Висновки до розділу

Основні результати даного розділу полягають в наступному:

1. Побудовано онтологію комунікаційної інфраструктури комп'ютерної мережі, що дало змогу описати вузли мережі, пристрої передачі даних, таблиці маршрутизації та інші необхідні сутності, які використовуються в процесі управління та моніторингу ресурсів.
2. Розроблено компоненти онтології для представлення та аналізу трафіку і потоків даних у комп'ютерних мережах, що дало змогу забезпечити проведення аналітики завантаженості каналів та її продуктивності.
3. Визначено онтологічні компоненти для відображення подій у комп'ютерній мережі, що дало змогу забезпечити ефективність моніторингу та управління мережевими ресурсами при виникненні аномалій та аналізу причин їх виникнення.

## РОЗДІЛ 4

### ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

#### 4.1. Охорона праці

Метою кваліфікаційної роботи магістра є дослідження методів та засобів моделювання процесів управління та моніторингу у комп'ютерних мережах на основі підходу онтологій. Оскільки, усі етапи такого дослідження передбачають використання комп'ютерної та оргтехніки, то актуальним є аналіз та дотримання вимог з охорони праці і техніки безпеки при роботі з ПК.

Для забезпечення ефективності та оптимізації роботи фахівців з управління та моніторингу параметрів комп'ютерних мереж необхідно організувати безпечні умови праці. При цьому безпосередню відповідальність за порушення нормативно-правових актів з охорони праці несуть як інженери, так і їх керівники [23, 24].

Окрім цього, на робочих місцях осіб, які проводять моделювання процесів управління та моніторингу ресурсів комп'ютерної мережі, необхідно забезпечити дотримання вимог НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Згідно НПАОП 0.00-7.15-18 приміщення, де розміщені робочі місця фахівців з налаштування комп'ютерних мереж, крім серверних, мають бути оснащені системою автоматичної пожежної сигналізації відповідно до вимог:

- переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації;
- Державних будівельних норм "Інженерне обладнання будинків і споруд.

В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі. Приміщення, де розміщені робочі місця інженерів, мають бути оснащені вогнегасниками, кількість яких визначається згідно з вимогами Типових норм належності вогнегасників, затверджених наказом Міністерства України з питань надзвичайних ситуацій.

Приміщення, в яких розміщуються робочі місця операторів серверів мережі загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог ДБН В.1.1-7-2016, ДСТУ Б.В.1.1-36:2016, НАПБ А.01.001-2014 і вимог нормативно-технічної та експлуатаційної документації виробника. Проходи до засобів пожежогасіння мають бути вільними.

Лінія електромережі для живлення комп'ютерів та периферійних пристроїв повинні бути виконаними як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники повинні відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам ДСТУ Б В.2.5-82:2016.

У приміщенні, де одночасно експлуатуються понад п'ять комп'ютерів, на помітному та доступному місці встановлюється аварійний

резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

Комп'ютери повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним.

Не допускається підключати комп'ютери до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.

Електромережі штепсельних з'єднань та електророзеток для живлення комп'ютерної техніки повинні бути виконаними за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі.

Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

При побудові онтологій комп'ютерних мереж, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у ДБН В.2.5-28 : 2018. „Природне і штучне освітлення”.

Організація робочого місця оператора ПК повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам.

Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами НПАОП 0.00-7.15-18.

Розміщення принтера на робочому місці має забезпечувати добру видимість екрана комп'ютера, зручність керування пристроєм введення-виведення інформації в зоні досяжності моторного поля.

Таким чином, у результаті аналізу вимог щодо охорони праці користувачів комп'ютерів, визначено особливості організації робочих місць, вимог з електробезпеки, природного та штучного освітлення для ефективної і безпечної роботи фахівців з моделювання процесів управління та моніторингу комп'ютерної мережі.

#### 4.2. Оцінка стійкості роботи об'єкту економіки до впливу вражаючих факторів ядерної зброї

Вихідні дані: об'єкт розташовується на відстані  $R = 7 \text{ км}$  км від ймовірного ядерного вибуху. Очікувана потужність ядерного боєзапасу  $q = 800 \text{ кт}$ , вибух наземний. Елементи системи, схильні до дії ЕМІ:

1 Живлення щитових: напруга 380 В по наземних неекраниваних кабелях  $L_1 = 60 \text{ м}$ . Кабелі мають вертикальне відхилення заввишки  $l_1 = 1 \text{ м}$ . Допустимі коливання напруги мережі  $\pm 5\%$ , коефіцієнт екранування кабелю  $\eta = 2$ .

2 Система організаційного обліку та керування підприємства складається з 28 ПК та активного і периферійного обладнання. Згадані пристрої виконані на мікросхемах, що мають струмопровідні елементи заввишки  $l_3 = 0,03 \text{ м}$ . Робоча напруга мікросхем 5 В. Живлення від загальної мережі напруги 220 В через трансформатор. Допустимі коливання напруги мережі  $\pm 5\%$ . Комп'ютерна мережа управління має горизонтальну лінію  $L_2 = 610 \text{ м}$  і вертикальні відгалуження заввишки  $l_2 = 2,5 \text{ м}$ . Робоча мінімальна напруга живлення 6В. Допустимі коливання напруги мережі

$\pm 5\%$ , коефіцієнт екранування комп'ютерної мережі  $\eta = 2$ .

Таблиця 4.1

**Вхідні дані для оцінки дії ЕМІ на стійкість об'єкту**

Відстань, км.	Потужність, кТ	Довжина, м		Допуск
		$L_1$	$L_2$	%
7	800	60	610	5

Розрахуємо очікувані на об'єкті максимальні значення вертикальної (ЕВ) і горизонтальної (ЕГ) складових напруженості електричного поля [22]:

$$E_B = 5 \cdot 10^3 \cdot \frac{(1 + 2 \cdot R)}{R^3} \cdot \lg(14,5 \cdot q), \text{ В/м} \quad (4.1)$$

$$E_r = 10 \cdot \frac{(1 + 2 \cdot R)}{R^3} \cdot \lg(14,5 \cdot q), \text{ В/м} \quad (4.2)$$

де  $R$  — відстань об'єкту від вірогідного ядерного вибуху;

$q$  — очікувана потужність ядерного боєзапасу.

$$E_B = 5 \cdot 10^3 \cdot \frac{(1 + 2 \cdot 7)}{10^3} \cdot \lg(14,5 \cdot 800) = 888,73 \text{ В/м}$$

$$E_r = 10 \cdot \frac{(1 + 2 \cdot 7)}{10^3} \cdot \lg(14,5 \cdot 800) = 1,78 \text{ В/м.}$$

Визначимо максимальну очікувану напругу наводок [22]:

а) у системі електроживлення:

$$U_B = \frac{E_B \cdot l_1}{\eta}, \quad (4.3)$$

$$U_r = \frac{E_r \cdot L_1}{\eta}, \quad (4.4)$$



де  $l_1$  – висота вертикального відхилення кабелю до верстатів;

$L_1$  – довжина екранованого кабелю;

$\eta$  – коефіцієнт екранування кабелю.

$$U_B = \frac{888,73 \cdot 1}{2} = 444,36 \text{ В}$$

$$U_r = \frac{1,78 \cdot 60}{2} = 53,32 \text{ В}$$

б) у комп'ютерній мережі управління:

$$U_B = \frac{E_B \cdot l_2}{\eta}, \quad (4.5)$$

$$U_r = \frac{E_r \cdot L_2}{\eta}, \quad (4.6)$$

де  $l_2$  – висота вертикального відгалуження комп'ютерної мережі;

$L_2$  – довжина горизонтальної лінії комп'ютерної мережі управління;

$\eta$  – коефіцієнт екранування кабелю.

$$U_B = \frac{888,73 \cdot 2,5}{2} = 1110,91 \text{ В}$$

$$U_r = \frac{1,78 \cdot 610}{2} = 542,13 \text{ В}$$

в) в комп'ютерах та периферійних пристроях:

$$U_B = \frac{E_B \cdot l_3}{\eta}, \quad (4.7)$$

де  $l_3$  – висота струмопровідних елементів;

$\eta$  – коефіцієнт екранування кабелю.

$$U_B = \frac{888,73 \cdot 0,03}{2} = 13,33 \text{ В}$$

Визначимо допустиму максимальну напругу наводок [22]:

а) у мережі живлення:

$$U_d = U + U(5\%), \quad (4.8)$$

де  $U$  — напруга живлення щитових;

$$U_{d1} = 380 + \frac{380 \cdot 5}{100} = 399 \text{ В}$$

б) у комп'ютерній мережі управління:

$$U_{d2} = 6 + \frac{6 \cdot 5}{100} = 6,3 \text{ В}$$

в) у комп'ютерах та периферійних пристроях:

$$U_{d4} = 5 + \frac{5 \cdot 5}{100} = 5,25 \text{ В}$$

Розрахуємо коефіцієнт безпеки [22]:

$$K = 20 \cdot \lg\left(\frac{U_d}{U_e}\right) \quad (4.9)$$

де  $U_d$  — допустима максимальна напруга наведень в пристрої введення, ЕОМ, блоці управління

$U_e$  — очікувана максимальна напруга наведень в комп'ютерах та периферійних пристроях.

$$K = 20 \cdot \lg\left(\frac{5,25}{13,33}\right) = -8,09 \text{ дБ}$$

Зведемо отримані дані в таблицю (табл. 4.2).

Таблиця 4.2

## Результати оцінки стійкості об'єкту до дії ЕМІ

Елементи системи	Допустима напруга мережі, В	Напруженість електричного поля, В/м		Наводки напруги в струмопровідних елементах, В	
		Е <sub>В</sub>	Е <sub>Г</sub>	U <sub>В</sub>	U <sub>Г</sub>
Електропостачання щитових	399	888,73	1,78	444,36	53,32
Комп'ютери, периферійні пристрої	5,25	888,73	1,78	13,33	—
Комп'ютерна мережа, активне обладнання	6,3	888,73	1,78	1110,91	542,13
Коефіцієнт безпеки $K = -8,09 \text{ дБ} \ll 40 \text{ дБ}$ .					

Локальна комп'ютерна мережа та обладнання може опинитися в зоні дії ЕМІ наземного ядерного вибуху. Нестійкими елементами об'єкту є: комп'ютери та периферійне обладнання, а також локальна комп'ютерна мережа. Об'єкт не стійкий до дії ЕМІ, оскільки коефіцієнт безпеки значно менше задовільного значення, що становить до 40 дБ.

Для підвищення стійкості роботи об'єкту до дії ЕМІ ядерного вибуху необхідно провести наступні заходи:

- кабель живлення щитових екранувати, помістивши в сталеві труби, а на входах встановити швидкодіючі відключаючі пристрої;
- комп'ютери і периферійні пристрої закрити заземленими екранами;
- для комп'ютерів встановити швидкодіючі відключаючі пристрої.

4.3. Забезпечення захисту працівників суб'єктів господарювання та населення від впливу іонізуючих випромінювань

Іонізуюче випромінювання поділяється на корпускулярне (потіки альфа-, бета-частинок, протонів) і електромагнітне (гамма-випромінювання,

рентгенівське). Перше має велику іонізуючу і малу проникну властивість, друге - меншу іонізуючу і велику проникну здатність.

У промисловості використовують радіоактивні ізотопи для вимірювання густини і вологості сировини і готових виробів, гамма-дефектоскопії, дозування сипких матеріалів і контролю їх рівня та в інших потребах.

Робота електровакуумних приладів часто супроводжується утворенням побічних ефектів; які шкідливо діють на обслуговуючий персонал. Зокрема, будь-який електровакуумний прилад, який працює з високими напругами на електродах, є джерелом рентгенівського випромінювання, потужним генератором важких та легких іонів обох полярностей, озону і оксидів азоту, а також підвищення температури повітря. В радіоелектронній апаратурі рентгенівське випромінювання виникає внаслідок електронного бомбардування електродів та інших поверхонь. Це потужні генераторні, модуляторні і посилювальні лампи, високовольтні тиратрони, кенотрони, електронно-променеві трубки, кінескопи, магнетрони та інші електровакуумні прилади, що працюють за прискорювальних напруг вище 5 кВ.

Потужність дози рентгенівського випромінювання побутової апаратури у будь-якій точці на відстані 5 см від її зовнішньої поверхні не повинна перевищувати 7,2 рА/кг; апаратів, що застосовуються для промислової дефектоскопії і медичних діагностичних досліджень - 1,44 рА/кг; відеоконтрольного пристрою телевізійної системи - 36 рА/кг на відстані 5 см від корпусу апарата на боці, зверненому до оператора.

Дія іонізуючих випромінювань на людину може бути місцевою і загальною. При місцевому опромінюванні може утворитись променева виразка, ракове захворювання. При загальному - може виникнути гостра або хронічна променева хвороба, яка супроводжується порушенням обмінних процесів у клітинах організму, змінами в центральній нервовій системі, крові, кровотворних органах. Крім зовнішнього, може бути внутрішнє

опромінення організму, яке виникає при потраплянні радіоактивних речовин всередину організму з повітрям, їжею. Дія іонізуючих променів, як і ЕМІТ, не сприймається органами чуттів людини.

Біологічна дія іонізуючих променів залежить від типу випромінювання і поглинутої дози. Поглинута доза  $D$  - це середня енергія, яка передана одиниці маси речовини. Одиницею її є Грей (Гр), який відповідає енергії в 1 Дж, що передана масі в 1 кг.

Враховуючи, що біологічна дія опромінення людини різними видами іонізуючих випромінювань не однакова, введено поняття еквівалентної дози  $H$ , яка визначається як добуток дози поглинання на коефіцієнт якості  $K$ :  
$$H = K \cdot D.$$

Одиницею еквівалентної дози є Зіверт (Зв), позасистемною - бер (1 бер = 0,01 Зв). Коефіцієнт якості для рентгенівського та гамма-випромінювання беруть за 1, нейтронів - 10, альфа-частинок - 20.

Для характеристики іонізуючої здатності випромінювань введено поняття експозиційної дози, яка являє собою повний заряд іонів одного знаку, що виникає в одиниці маси сухого атмосферного повітря. Одиниця експозиційної дози - кулон на кілограм, позасистемна - рентген (Р). Поглинена, еквівалентна і експозиційна доза, віднесені до одиниці часу, називаються потужністю дози. Потужність експозиційної дози називають також рівнем радіації.

Для захисту населення від впливу іонізуючого випромінювання, розроблено ряд законів та актів, що повинні гарантувати охорону та збереження здоров'я людини. Зокрема, до повноважень Кабінету Міністрів України, міністерств та інших центральних органів виконавчої влади щодо забезпечення захисту людини від впливу іонізуючого випромінювання належать.

Рівень випромінювання на робочих місцях та ефективність радіаційного захисту контролює служба радіаційної безпеки. Для дозиметричного контролю застосовують комплекти індивідуальних

дозиметрів, дозиметри типу ДРГ, рентгенометри ДП та ін.

Захист від іонізуючого випромінювання забезпечується такими методами і засобами:

- ізоляцією або огороженням його джерела за допомогою спеціальних камер, екранів;
- "захистом часом";
- "захистом відстанню";
- застосуванням дистанційного управління, сигналізації і засобів контролю;
- використанням засобів індивідуального захисту.

Вибір матеріалу для загород і екранів залежить від проникаючої здатності випромінювання. Альфа-частинки затримує навіть аркуш паперу, для захисту від бета-частинок необхідні матеріали більшої густини, а захист від гамма-променів здійснюється матеріалами з великою атомною масою.

Висновки.

Проведено розрахунок стійкості об'єкту економіки до уражаючих факторів ядерного вибуху і за заданими умовами встановлено, що для підвищення його стійкості потрібно екранувати кабель живлення щитових, помістивши його в сталеві труби, а на входах встановити швидкодіючі відключаючі пристрої. Комп'ютери і периферійні пристрої закрити заземленими екранами і встановити швидкодіючі відключаючі пристрої. Визначено фактори негативного впливу іонізуючого випромінювання на організм людини і наведено способи захисту працівників суб'єктів господарювання та населення від їх дії.

## ВИСНОВКИ

Основні наукові та практичні результати полягають в наступному.

1. Проведено аналіз методів моделювання комп'ютерних мереж у результаті якого представлено концепцію онтологій та їх компонентів, що дало змогу визначити способи організації спеціалізованих баз даних для реалізації процесів управління ресурсами та моніторингу стану мережі.

2. Проведено аналіз базових понять при побудові онтологій комп'ютерних мереж, що дало змогу визначити основні сутності та відношення між ними для домену комп'ютерних мереж.

3. Проведено аналітичний огляд принципів проектування онтології та методології їх розробки, а також наведено метрики за якими можна проводити оцінювання якості онтологій, що дало змогу в перспективі забезпечити об'єктивність і достовірність моделювання комп'ютерних мереж.

4. Запропоновано та обгрунтовано застосування математичної моделі для опису онтологій комп'ютерних мереж за допомогою кортежа, який включає в себе п'ять елементів: домен, можливі ситуації або стани системи, операції, функція відображення сутностей домену та поведінка об'єктів, що дають змогу забезпечити повноту та ефективність при описі процесів управління та моніторингу ресурсів комп'ютерної мережі.

5. Визначено лексикон, граматику та синтаксичні обмеження утворення термів при побудові онтології комп'ютерної мережі, що дало змогу формувати складні висловлювання на основі предикатів та забезпечити гнучкість моделювання процесів, які протікають у комп'ютерній мережі.

6. Запропоновано архітектуру онтології для представлення об'єктів, відношень та обмежень в комп'ютерних мережах, що дало змогу забезпечити та змодельовати домен для опису процесів управління та моніторингу.

7. Побудовано онтологію комунікаційної інфраструктури комп'ютерної мережі, що дало змогу описати вузли мережі, пристрої передачі даних, таблиці маршрутизації та інші необхідні сутності, які використовуються в процесі управління та моніторингу ресурсів.

8. Розроблено компоненти онтології для представлення та аналізу трафіку і потоків даних у комп'ютерних мережах, що дало змогу забезпечити проведення аналітики завантаженості каналів та її продуктивності.

9. Визначено онтологічні компоненти для відображення подій у комп'ютерній мережі, що дало змогу забезпечити ефективність моніторингу та управління мережевими ресурсами при виникненні аномалій та аналізу причин їх виникнення.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Городецька О., Гикавий В., Онищук О. Комп'ютерні мережі. Навчальний посібник. Вінниця: ВНТУ. 2015. 128 с.
2. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі : навчальний посібник. Львів : «Магнолія 2006». 2013. 256 с.
3. Мельник І., Лунтовський А. Проектування та дослідження комп'ютерних мереж. К. : Університет «Україна». 2010. 362 с.
4. Ткаченко В., Касілов О., Рябик В. Комп'ютерні мережі та телекомунікації: навч. посіб. Харків: НТУ «КПІ». 2011. 224 с.
5. Басюк Т.М., Литвин В.В. Мови опису онтологій. Навчальний посібник. Львів: Видавництво Львівської політехніки, 2020. 276 с.
6. Литвин В.В., Угрин Д.І. Метод автоматичної розбудови адаптивної онтології. Вісник Національного технічного університету «ХПІ». Харків. 2011. № 10. С. 75–82.
7. Литвин В.В. Автоматизація процесу розвитку базової онтології на основі аналізу текстових ресурсів. Вісн. Нац. ун-ту «Львівська політехніка». Серія: Інформаційні системи та мережі. 2010. № 673. С. 319–325.
8. Web Ontology Language (OWL). URL: [www.w3.org/2004/OWL/](http://www.w3.org/2004/OWL/) (дата звернення: 10.09.2023).
9. Resource Description Framework (RDF). URL: [www.w3.org/RDF/](http://www.w3.org/RDF/) (дата звернення: 15.09.2023).
10. KA2 - Knowledge Acquisition Community Ontology. URL: <http://ontobroker.semanticweb.org/ontos/ka2.html> (дата звернення: 12.09.2023).
11. Литвин В.В., Басюк Т.М., Досин Д.Г. Онтологічний інжиніринг. Львів: Видавництво Львівської політехніки, 2017. 224 с.

12. Liu Y., Xu C., Zhang Q., Pan Y. The smart architect: Scalable ontology based modeling of ancient chinese architectures. *IEEE Intelligent Systems*, 23(1). 2008. pp. 49–56.
13. DiPietro I., Pagliarecci F., Spalazzi L. Model checking semantically annotated services. *IEEE Transactions on Software Engineering*, 38(3). 2012. pp. 592–608.
14. Liu Y., Zhang M., Jiang Y., Zhao H. Improving procedural modeling with semantics in digital architectural heritage. *Computers and Graphics*, 36(3). 2012. pp. 178–184.
15. Liu Y., Zhang M., Tang F., Jiang Y., Pan Z., Liu G. et al. Constructing the virtual jing-hang grand canal with ontodraw. *Expert Systems With Applications*. 39(15). 2012. pp. 12071–12084.
16. Онтологія в комп'ютерних системах. URL: <https://rsdn.org/article/philosophy/what-is-onto.xml> (дата звернення: 16.09.2023).
17. Bechhofer S., Horrocks I., Goble C., Stevens R. OilEd: A Reason-able Ontology Editor for the Semantic Web // Joint German/Austrian conf. on Artificial Intelligence (KI'01). *Lecture Notes in Artificial Intelligence LNAI 2174*, Springer-Verlag, Berlin. 2001. pages.396-408.
18. Литвин В.В. Технології менеджменту знань: навч. посібник /; за заг. ред. В.В. Пасічника. Львів: Видавництво Львівської політехніки. 2010. 260 с.
19. Smith M., Veltz K., OWL, MacGinness D. OWL language of web-ontolgy. Direction. Recommended W3C. URL: <http://www.w3.org/TR/2004/REC-owl-guide-20040210/> (дата звернення: 15.09.2023).
20. Довгий С.О., Величко В.Ю., Глоба Л.С., Стрижак О.Є. та ін. Комп'ютерні онтології та їх використання у навчальному процесі. Теорія і практика. Монографія. К. : Інститут обдарованої дитини. 2013. 310 с.

21. Haag F., Lohmann S., Siek S., Ertl T. QueryVOWL: Visual Composition of SPARQL Queries. Proceedings of ESWC 2015 Satellite Events. Springer. 2015. pp. 62- 66.

22. Яцишин В.В., Кучма І.М. Побудова онтологій як спосіб ефективного моделювання комп'ютерних систем та мереж. Матеріали XII міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2022. С. 439.

23. Яцишин В.В., Кучма І.М. Класифікація онтологій в процесі моделювання комп'ютерних мереж. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2022. С. 162.

24. Жидецький В.Ц. Охорона праці користувачів комп'ютерів. Львів: Афіша, 2011. 176 с.

25. Желібо Е.Н. Безпека життєдіяльності: Навчальний посібник/ За редакцією Е.П. Желібо, В.М. Львів: «Новий світ - 2000», 2011. 320с.

26. Стадник І.Я., Зварич Н.М. Оцінка хімічної обстановки при аваріях на хімічно небезпечних об'єктах викидом (виливом) небезпечних хімічних речовин та застосуванні хімічної зброї. ТНТУ. 2020. 36 С.

Додаток А  
Тези конференцій

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
Тернопільський національний технічний університет імені Івана Пулюя (Україна)  
Університет імені П'єра і Марії Кюрі (Франція)  
Маріборський університет (Словенія)  
Технічний університет у Кошице (Словаччина)  
Вільнюський технічний університет ім. Гедимінаса (Литва)  
Міжнародний університет цивільної авіації (Марокко)  
Наукове товариство ім. Т.Шевченка

# **АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ**

**Збірник**  
тез доповідей

**ХІІ Міжнародної науково-практичної  
конференції молодих учених та студентів**  
6-7 грудня 2023 року



**УКРАЇНА**  
**ТЕРНОПІЛЬ – 2023**

38.	<b>Т. Крамар</b> ДЕЦЕНТРАЛІЗОВАНЕ АВТОМАТИЧНЕ ПІДКЛЮЧЕННЯ ПУНКТИВ НЕЗЛАМНОСТІ ПІД ЧАС ВІДКЛЮЧЕНЬ У ЗИМП 2023 В ПРИФРОНТОВИХ ЗОНАХ УКРАЇНИ	415
39.	<b>Б. Б. Млинко, О. П. Стефанюк</b> АНАЛІЗ ВИКОРИСТАННЯ ІГРОВИХ РУШІВ ДЛЯ СТВОРЕННЯ ЦИФРОВИХ ДВІЙНИКІВ НА ОСНОВІ СИСТЕМНОГО ПІДХОДУ	417
40.	<b>Н. М. Коцюк, В. Д. Тимощук, Ю. О. Момоток, Н. С. Луцки</b> СИСТЕМА РЕЗЕРВУВАННЯ ТРАФІКУ НА ОСНОВІ МІКРОТІК	419
41.	<b>В. В. Василюшин, В. Д. Тимощук, Н. Ю. Кігчак, Н. С. Луцки</b> АНАЛІЗ ХАРАКТЕРИСТИК ТА ЗАСТОСУВАННЯ МІКРОКОНТРОЛЕРІВ ATTINY85, ATMEGA8, RP2040	420
42.	<b>А. М. Ковтко, Н. В. Лещук, І. Р. Козбур, І. В. Коноваленко</b> АНАЛІЗ ЕФЕКТИВНОСТІ СИСТЕМ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНИХ ПРОДУКТІВ	421
43.	<b>О. Ю. Замора, А. В. Немеришин, І. Р. Козбур, О. Р. Дмитрів</b> АНАЛІЗ МЕРЕЖЕВИХ СИСТЕМ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ З ВИКОРИСТАННЯМ ПРОТОКОЛІВ МНОЖИННОГО ДОСТУПУ	423
44.	<b>М. В. Дрогобицький, Н. С. Луцки, А. М. Паламар</b> КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО КОНТРОЛЮ РІВНЯ ШУМУ НАВКОЛИШНЬОГО СЕРЕДОВИЩА	425
45.	<b>І. В. Лялик, А. М. Паламар</b> КОМП'ЮТЕРНА СИСТЕМА ДИСТАНЦІЙНОГО КОНТРОЛЮ ІНТЕНСИВНОСТІ УЛЬТРАФІОЛЕТОВОГО ВИПРОМІНЮВАННЯ	426
46.	<b>А. М. Паламар, Д. С. Сома, В. П. Волоський</b> КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ВІДДАЛЕНОГО СПОСТЕРЕЖЕННЯ ЗА РІВНЕМ НАСИЧЕННЯ КИСНЕМ КРОВІ ЛЮДИНИ	427
47.	<b>М. В. Криховецький</b> МЕТОДИ ВІЯВЛЕННЯ ДРОНІВ НА БАЗІ НЕЙРОННИХ МЕРЕЖ	428
48.	<b>Д. І. Муштин</b> МОБІЛЬНА МЕТЕОСТАНЦІЯ ДЛЯ ОБПРИСКУВАЧА	431
49.	<b>Л. Є. Мосій, І. В. Струтинська, Г. В. Козбур</b> РОЛЬ КОМП'ЮТЕРНО-ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У ЦИФРОВІЙ ТРАНСФОРМАЦІЇ ЕКОНОМІКИ.	432
50.	<b>О. Є. Подвисоцький, Н. Б. Стадник</b> МЕТОДИ БІОМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ В РОЗУМНОМУ БУДИНКУ	435
51.	<b>А. М. Паламар, Р. О. Романчук</b> КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ВІДДАЛЕНОГО КОНТРОЛЮ РІВНЯ ЗАБРУДНЕННЯ ПОВІТРЯ ПИЛОМ	436
52.	<b>Є. В. Тиш, Р. І. Шалапай</b> ТИПИ ВИМОГ ДО КОМП'ЮТЕРНИХ СИСТЕМ І МЕТОДИ ЇХ ВІЯВЛЕННЯ	437
53.	<b>А. М. Луцків, С. В. Макогон</b> НЕЙРОМЕРЕЖЕВІ ПІДХОДИ ДО ПЕРЕТВОРЕННЯ ТЕКСТОВИХ ПОВІДОМЛЕНЬ В АУДИОПОТІК	438
54.	<b>В. В. Яцишин канд. І. М. Кучма</b> ПОБУДОВА ОНТОЛОГІЙ ЯК СПОСІБ ЕФЕКТИВНОГО	439

УДК 004.94

**В. В. Яцишин канд. техн. наук, доцент, І. М. Кучма**

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

### **ПОБУДОВА ОНТОЛОГІЙ ЯК СПОСІБ ЕФЕКТИВНОГО МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ**

**V. V. Yatsyshyn PhD., Assoc. Prof., I. M. Kuchma**

### **ONTOLOGY BUILDING AS A METHOD OF EFFICIENT MODELING OF COMPUTER SYSTEMS AND NETWORKS**

Онтології можна визначити за допомогою артефактів, які мають структуру (логічну, лінгвістичну, «таксономічну»). Їхня мета полягає в кодуванні опису домену (фактичного, контрфактичного, можливого, неможливого, бажаного тощо) для деяких завдань або проблем (наприклад, область медицини, світ семантичної веб-конференції тощо). Однак онтології повинні відповідати домену та завданню, описаному нижче:

- домен – дозволяє описувати сутності, атрибути та зв'язки які стосуються певної мети (студент як сутність, яка зарахована до університету, під керівництвом академічного персоналу та має назву, адресу тощо);

- завдання – допомагає досягти мети (знайти людей, які працюють над тією самою темою, узгодити теми проекту з компетенціями персоналу тощо).

Проектування онтології є досить складним завданням для складних комп'ютерних систем та мереж і важливою технікою у створенні додатків, заснованих на знаннях. У декількох областях досліджень успішно застосовано проектування онтології, зокрема, такі види як перевірка моделі та семантичний аналіз, виявлення неузгодженості в моделюванні складного сценарію. До основних критеріїв і принципів проектування, які виявилися корисними при розробці онтологій належать:

- об'єктивність – означає, що онтологія повинна надавати значення визначених термінів шляхом надання об'єктивних визначень, а також документації природною мовою;

- повнота – вираження означення базується на необхідних і достатніх умовах, що є кращим перед частковим означенням;

- когерентність – прийнята, щоб формувати висновки, які представляють узгодженість з означеннями;

- максимальна монотонна розширюваність – властивість, яка передбачає, що нові загальні або спеціалізовані терміни повинні бути включені в онтологію таким чином, щоб не вимагати зміни існуючих понять та їх означень;

- мінімальні онтологічні зобов'язання – означає мінімізацію претензій щодо області, яка моделюється, надаючи сторонам свободу спеціалізації та інстанціювання онтології як вимагається;

- диверсифікація ієрархій – корисна для збільшення потужності, що забезпечується кількома механізмами наслідування;

- принцип онтологічного розрізнення – передбачає, що класи в онтологіях повинні бути такими, які не перетинаються;

- мінімізація семантичної дистанції між однотипними поняттями – однакові екземпляри групуються та використовуються для представлення подібних понять.

- модульність – властивість, яка використовується для мінімізації зв'язку між модулями.

<b>Ясків О.П., Крисюк І.В.</b> <b>ФАКТОРИ ВПЛИВУ НА НАДІЙНІСТЬ КОМП'ЮТЕРНИХ СИСТЕМ В ПРОЦЕСІ ЇХ РОЗРОБКИ</b> <b>Yasnyy O.P., Krysiuk I.V.</b> <b>EFFECTS RELIABILITY FACTORS OF COMPUTER SYSTEMS IN THE PROCESS OF THEIR DEVELOPMENT</b>	161
<b>Василь Яцишин, Іван Кучма</b> <b>КЛАСИФІКАЦІЯ ОНТОЛОГІЙ В ПРОЦЕСІ МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ МЕРЕЖ</b> <b>Vasyl Yatsyshyn, Ivan Kuchma</b> <b>CLASSIFICATION OF ONTOLOGIES IN THE PROCESS OF COMPUTER NETWORK MODELING</b>	162
<b>І.В. Лилік, А.М. Паламар</b> <b>КОМП'ЮТЕРИЗОВАНА СИСТЕМА МОНІТОРИНГУ РІВНЯ УЛЬТРАФІОЛЕТОВОГО ВИПРОМІНЮВАННЯ НА ОСНОВІ ІНТЕРНЕТУ РЕЧЕЙ</b> <b>I.V. Lylyk, A.M. Palamar</b> <b>COMPUTERIZED ULTRAVIOLET RADIATION LEVEL MONITORING SYSTEM BASED ON THE INTERNET OF THINGS</b>	163
<b>Андрій Луцків, Сергій Макогон</b> <b>ТИПИ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ДЛЯ ПЕРЕТВОРЕННЯ ТЕКСТОВИХ ПОВІДОМЛЕНЬ У ЗВУКОВИЙ ПОТІК</b> <b>Andriy Lutskiv, Serhii Makohon</b> <b>TYPES OF NEURAL NETWORK ARCHITECTURES FOR TEXT TO SPEECH</b>	164
<b>Андрій Луцків, Юрій Мельничук</b> <b>МУЛЬТИАГЕНТНА ОРГАНІЗАЦІЯ СЕРВЕРА ОНЛАЙН АУКЦІОНІВ</b> <b>Andriy Lutskiv, Yuriy Melnychuk</b> <b>MULTI-AGENCY ONLINE AUCTION SERVER ORGANIZATION</b>	165
<b>Галина Осухівська, Денис Муштин</b> <b>КОМП'ЮТЕРИЗОВАНА СИСТЕМА КОНТРОЛЮ ЗА МЕТЕОДАНИМИ ДЛЯ ОБПРИСКУВАЧА</b> <b>Halyna Osukhivska, Denys Mushryn</b> <b>COMPUTERIZED METEODATA CONTROL SYSTEM FOR SPRAYER</b>	166
<b>Т.А. Озарків; Р.О. Жаровський</b> <b>МЕТОД ОПТИМІЗАЦІЇ EIGRP ПРОТОКОЛУ ДЛЯ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ПЕРЕДАЧІ ДАНИХ В КОМП'ЮТЕРНИХ МЕРЕЖАХ</b> <b>T. A. Ozarkiv; R.O. Zharovskiy</b> <b>THE METHOD OF OPTIMIZING THE EIGRP PROTOCOL TO INCREASE THE PRODUCTIVITY OF DATA TRANSMISSION IN COMPUTER NETWORKS</b>	167
<b>Андрій Луцків, Андрій Островський</b> <b>ОРГАНІЗАЦІЯ ДОСТУПУ ДО МОДЕЛІ GPT-3 ЗАСОБАМИ МОВИ PYTHON</b> <b>Andriy Lutskiv, Andriy Ostrovskiy</b> <b>ORGANIZING ACCESS TO THE GPT-3 MODEL USING PYTHON</b>	168
<b>А.М. Паламар, Р.О. Романчук, М.В. Дрогобицький</b> <b>КОМП'ЮТЕРИЗОВАНА СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО КОНТРОЛЮ РІВНЯ КОНЦЕНТРАЦІЇ ПИЛУ НА ОСНОВІ ІНТЕРНЕТУ РЕЧЕЙ</b> <b>A.M. Palamar, R.O. Romanchuk, M.V. Drohobytckiy</b> <b>COMPUTERIZED SYSTEM FOR REMOTE MONITORING OF DUST CONCENTRATION LEVEL BASED ON THE INTERNET OF THINGS</b>	169
<b>Ярослав Панчишин</b> <b>СТРУКТУРА СИСТЕМИ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ МІНІ-ТЕПЛИЦІ</b> <b>Yaroslav Panchyshyn</b> <b>STRUCTURE OF THE MINI-GREENHOUSE MICROCLIMATE PARAMETER CONTROL SYSTEM</b>	170

УДК 004.94

Василь Яцишин канд. техн. наук, доцент, Іван Кучма

Тернопільський національний технічний університет імені Івана Пулюя

### КЛАСИФІКАЦІЯ ОНТОЛОГІЙ В ПРОЦЕСІ МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ МЕРЕЖ

Vasyl Yatsyshyn PhD., Assoc. Prof., Ivan Kuchma

### CLASSIFICATION OF ONTOLOGIES IN THE PROCESS OF COMPUTER NETWORK MODELING

Для класифікації онтологій використовується поняття онтологічної прихильності. Якщо є можливість розрізнити типи онтологічної прихильності, то можна ідентифікувати різні типи онтологій, а отже, можна класифікувати онтології.

Для визначення різних типів прихильності загальноприйнятим у сфері штучного інтелекту є розмежування між завданнями, методами та доменами, я показано на рис. 1.

Компоненти онтології	Джерело онтологічної прихильності	Приклад
Термінологія домену	Репрезентативна термінологія	
Онтологія домену	Загальні доменні теорії	Концепт, відношення
Онтологія орієнтована на модель домену	Тип задачі	Структурні моделі Анатомічні моделі
Онтологія орієнтовані на задачі	Метод	Обмеження конфігурації
Онтологія специфічних методів		Виправлення у пропозиціях і при перегляді

Рис. 1. Типи прихильності при класифікації онтологій

Таким чином, можна виділити три різні типи онтологічних прихильності: прихильність у випадку задач, прихильність методу та прихильність домену.

Формулювання прихильності задач можна виразити наступним чином: якщо онтологія визначає сутності та зв'язки, що виражають перспективу конкретного завдання на знання предметної області, тоді вона має прихильність щодо завдань. Задачі можна визначити як специфікацію цілі, включаючи деякі вхідні та необхідні результати. Онтологія для задач діагностики, включає такі сутності, як спостереження, гіпотези та причини і є прикладом онтології, яка має прихильність до задач.