НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ОДЕСЬКА ПОЛІТЕХНІКА”

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кафедра комп'ютерних інтелектуальних систем та мереж

БРОШОВЯН Олександр Анатолійович

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

ЗАСТОСУВАННЯ БАГАТОРІВНЕВИХ ОНТОЛОГІЙ У СЕНСОРНИХ МЕРЕЖАХ

Спеціальність 123 – Комп`ютерна інженерія

Спеціалізація – Комп'ютерні системи та мережі

Керівник: Тішин Петро Метталинович

Одеса – 2022

**З А В Д А Н Н Я**

**НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Брошовян\_Олександр\_Анатолійович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(прізвище, ім’я, по батькові)**

1. Тема проекту (роботи)\_Застосування багаторівневих онтологій у сенсорних мережах\_\_\_\_\_ \_\_\_\_\_\_

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_**­

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** **\_**

керівник проекту (роботи)\_Тішин\_Петро\_Метталинович\_доцент\_ \_ \_\_\_\_\_\_\_\_,

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ректора ОНПУ від “\_06\_”\_\_\_\_\_06\_\_\_\_\_2022 року №\_187-в \_

2. Строк подання студентом проекту (роботи)\_13.06.2022\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_\_\_\_\_

3. Вихідні дані до проекту (роботи)\_Завдання\_на\_розробку\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)\_1\_Аналіз\_існуючих\_рішень\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

\_2\_Завдання\_на\_розробку\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_3\_Проектування\_багаторівневої\_онтології\_сенсорної\_мережі\_системи\_опалення\_\_4\_Використання\_багаторівневої\_онтології\_сенсорної\_мережі\_системи\_опалення\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_\_\_\_\_\_ \_\_\_\_\_

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

\_Методи\_та\_онтології\_використовувані\_в\_роботі;\_Задачі\_проектування\_\_\_\_\_\_\_\_\_\_\_\_багаторівневої\_онтології;\_Складові\_багаторівневої\_онтології;\_Прикладна\_логіка\_\_онтології\_модуля\_сенсорної\_мережі;\_Прикладна\_логіка\_та\_візуалізація\_\_\_\_\_\_\_\_\_\_\_онтології\_модуля\_оцінки\_ризиків;\_Прикладна\_логіка\_та\_візуалізація\_онтології\_\_\_модуля\_системи\_опалення;\_Демонстрація\_схильних\_до\_зношення\_компонентів\_\_\_обладнання\_системи\_опалення;\_Описання\_моделі\_оцінки\_ризиків;\_Діаграмма\_\_\_\_загроз\_та\_метод\_її\_опису\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Консультанти розділів проекту (роботи)**

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада  консультанта | Підпис, дата | |
| завдання видав | завдання  прийняв |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

7. Дата видачі завдання\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***КАЛЕНДАРНИЙ ПЛАН***

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів дипломного  проекту (роботи) | Строк виконання етапів проекту  ( роботи ) | Примітка |
| 1 | Аналіз існуючих рішень |  |  |
| 2 | Завдання на розробку |  |  |
| 3 | Проектування багаторівневої онтології сенсорної мережі системи опалення |  |  |
| 4 | Використання багаторівневої онтології сенсорної мережі системи опалення |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Студент \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

( підпис ) (прізвище та ініціали)

**Керівник проекту (роботи) \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

( підпис ) (прізвище та ініціали)

|  |  |  |  |
| --- | --- | --- | --- |
| Відомість кваліфікаційної роботи бакалавра | | | |
| № рядка | Найменування | Кільк. | Примітка |
| 1 | Пояснювальна записка | 57 |  |
| 2 | Методи та онтології використовувані в роботі | 1 |  |
| 3 | Задачі проектування багаторівневої онтології | 1 |  |
| 4 | Складові багаторівневої онтології | 1 |  |
| 5 | Прикладна логіка онтології модуля сенсорної мережі | 1 |  |
| 6 | Прикладна логіка та візуалізація онтології модуля оцінки ризиків | 1 |  |
| 7 | Прикладна логіка та візуалізація онтології модуля системи опалення | 1 |  |
| 8 | Демонстрація схильних до зношення компонентів обладнання системи опалення | 1 |  |
| 9 | Описання моделі оцінки ризиків | 1 |  |
| 10 | Діаграма загроз та метод її опису | 1 |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |
| 18 |  |  |  |
| 19 |  |  |  |
| 20 |  |  |  |
| 21 |  |  |  |
| 22 |  |  |  |
| 23 |  |  |  |
| 24 |  |  |  |

ОП ІКС

КІСМ АМ181

1

Листів

Літ.

Затвердив

Н. Контр.

Реценз.

Перевірив

Тішин П.М.

Розробив

Брошовян О.А.

**АМДП.АМ181.0108**

1

Лист

Дата

Підпис

№ докум.

Лист

Зм.

Застосування багаторівневих онтологій в сенсорних мережах

**АНОТАЦІЯ**

### **Брошовян О.А. Застосування багаторівневих онтологій у сенсорних мережах** – кваліфікаційна робота бакалавра. Одеса, 2022: 57 с., 40 рис., 2 табл., 6 джерел.

У роботі виконано аналіз технологій сенсорних мереж, які часто використовуються при проектуванні промислових або автоматизованих жилих мереж та при організації побутового господарства, розглянуті можливі вразливості у безпеці мережі та загрози фізичного характеру. Визначені основні методи та підходи щодо оцінки ризиків загроз безпеки сенсорної мережи за допомогою створення додаткових онтологій і необхідної моделі даної оцінки для аналізу можливих варіантів ефективного розпізнавання і класифікації ризиків.

Визначений необхідний центр комутації, апаратне та програмне забезпечення, датчики температури, додаткові модулі управління та кінцеві пристрої.

Розроблені структури онтологій сенсорної мережі та оцінки ризиків за допомогою мови прикладної логіки, створена модель оцінки ризиків за допомогою онтології та таблиць компонентів пристроїв мережі, що можуть підлягати поломці або зношенню.

Проведено тестування створених баз знань на основі онтологій та аналізу можливих ризиків системи опалення.

**СЕНСОРНІ МЕРЕЖІ, ЗАГРОЗИ БЕЗПЕКИ МЕРЕЖІ, ВРАЗЛИВОСТІ МЕРЕЖІ, ЦЕНТРАЛЬНИЙ КОМУТАЦІЙНИЙ ЦЕНТР, АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ДАТЧИКИ ТЕМПЕРАТУРИ.**

**ABSTRACT**

### **Broshovian O.A.** **Application of multilevel ontologies in sensor networks** **-**bachelor's thesis. Odessa, 2021: 57 p., 40 pic., 2 tablets, 6 sources.

The work analyzes sensor network technologies that are often used in the design of industrial or automated residential networks and in the organization of household services, considers possible network security vulnerabilities and physical threats. The main methods and approaches for assessing the risks of sensor network security threats are defined by creating additional ontologies and the necessary model of this assessment for analyzing possible options for effective risk recognition and classification.

The required switching center, hardware and software, temperature sensors, additional control modules, and end devices have been identified.

Structures of sensor network ontologies and risk assessment were developed using the applied logic language, a risk assessment model was created using ontologies and tables of components of network devices that may be subject to breakage or wear.

The created knowledge bases were tested on the basis of ontologies and analysis of possible risks of the heating system.

**SENSOR NETWORKS, NETWORK SECURITY THREATS, NETWORK VULNERABILITIES, CENTRAL SWITCHING CENTER, HARDWARE, SOFTWARE, TEMPERATURE SENSORS.**

**ЗМІСТ**

[1 Аналіз існуючих рішень 5](#_Toc106276321)

[1.1 Використання OWL-онтологій в сенсорних мережах 5](#_Toc106276322)

[1.2 Багатосортні мови прикладної логіки 7](#_Toc106276323)

[1.3 Методологія Coras 9](#_Toc106276324)

[1.4 Онтологія HSSN 11](#_Toc106276325)

[1.5 Застосування онтологій для опису середовищ предметних областей 13](#_Toc106276326)

[2 Завдання на розробку 15](#_Toc106276327)

[2.1 Визначення використовуваного обладнання та основних цілей сенсорних мереж систем опалення 15](#_Toc106276328)

[2.2 Багаторівнева модель онтології 17](#_Toc106276329)

[3 Проектування багаторівневої онтології сенсорної мережі системи опалення 19](#_Toc106276330)

[3.1 Прикладна логіка модуля системи опалення 19](#_Toc106276331)

[3.2 Прикладна логіка модулю онтології оцінки ризиків 28](#_Toc106276332)

[3.3 Прикладна логіка модулю онтології сенсорної мережі 29](#_Toc106276333)

[4 Використання багаторівневої онтології сенсорної мережі системи опалення 31](#_Toc106276334)

[4.1 Онтологія найнижчого рівня на основі прикладної логіки системи опалення (SHSO) 31](#_Toc106276335)

[4.2 Онтологія другого рівня на основі прикладної логіки оцінки ризиків 46](#_Toc106276336)

[4.3 Визначення оцінки ризиків сенсорної мережі системи опалення 50](#_Toc106276337)

[Висновок 57](#_Toc106276338)

[Перелік літературних джерел 58](#_Toc106276339)

**ВСТУП**

Сенсорні мережі - це фундамент закладений для основи розвитку технологій сфери Інтернет речей, що на поточний момент часу є вкрай перспективною екосистемою автономних пристроїв. Протягом десятиліть безліч інженерів і вчених вивчають і займаються побудовою автономних мереж, що працюють автоматизовано від людини і дозволяють за допомогою сенсорних пристроїв обробляти і виокремлювати з уже осмислених даних мереж корисні знання для їх використання в знаходженні рішень складних комплексних завдань.

Наприклад, ризик виникнення збою обладнання у певних вузлах мережі, відмова роботи відразу декількох різних пристроїв мережі у результаті стрибка напруги і можливість перевизначення магістралі потоку даних на інші робочі пристрої.

Онтології сенсорних мереж є підходом до характеризації концептуальної моделі предметної області, що використовує логічні конструкції відповідності класів та властивостей об’єктів класів для цілісної комплектації знань обраної предметної області. Онтологічний підхід до формалізації знань предметної області дозволяє визначити необхідні знання та використати їх для таких ситуацій:

* для процесів вирішення завдань, що стоять в даній інтелектуальної мережі;
* для опису взаємодії різних вхідних і вихідних даних пристроїв в поле зору однієї системи;
* для можливості розширення побудованої концепції за допомогою перевикористання знань в інших схожих областях

Мета кваліфікаційної роботи – створення багаторівневої онтологій сенсорної мережі системи опалення дому для дослідження особливостей проектування такої системи для одноповерхового будинку та розробки загальної моделі оцінки ризиків на основі знань отриманих з онтологій сенсорної мережі та оцінки ризиків.

# АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

## Використання OWL-онтологій в сенсорних мережах

Мова веб-онтологій OWL, є ключовою ланкою, для опису предметної області: "Система опалення приватного будинку", тому що дозволяє описувати елементи за методом "екземпляр класу - властивість" при цьому у кожного елемента онтології є власний ідентифікатор - "URI" (уніфікований ідентифікатор ресурсу), необхідний для правильного ототожнення і організації ресурсів онтології.

Мова OWL за офіційним посібником [1] є одним із специфікаційних стандартів опису семантичних мереж, які представляють із себе модель інформаційних схем певної предметної області у вигляді графа, де:

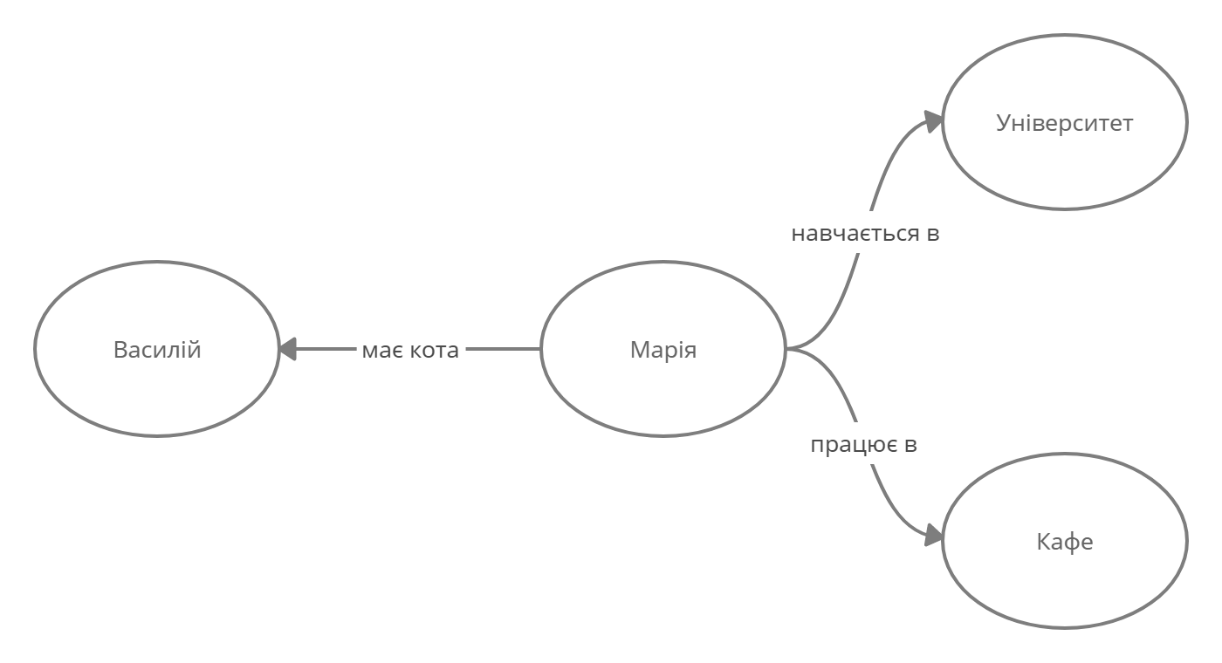


Рисунок 1.1 – Приклад семантичної мережі

* Вершини - описувані об'єкти (властивості, поняття, назви процесів, події)
* Дуги - відносини між об'єктами (вимірювати, мати, використовувати, бути інструментом для, бути властивістю об'єкта і т. п.)

Онтології в якості однієї з основних технологій для сенсорних мереж необхідні для поліпшення семантичної сумісності, інтеграції нових понять, які розширюють поточну побудовану концепцію мережі, а також спрощують міркування і класифікацію вже наявних уявлень в області бази знань.

Семантична сенсорна мережа (SSW) являє собою поєднання сенсорів та технології семантичної павутини. Поточні технології в області семантичних сенсорних мереж дають широкі можливості для організації автоматизованої керованої мережі, що дозволяє контролювати сенсори і отримані ними дані за допомогою різних високорівневих наборів правил поведінки мережі.

Однією з таких технологій, які задають такі правила стала SWE (Sensor Web Enablement) – це пакет розроблених стандартів, який підтримується відкритим геопросторовим консорціумом (OGC). Даний пакет включає в себе велику кількість стандартів SWE, які дозволяють розробникам відкрито використовувати і розробляти під свої потреби будь-які типи датчиків, перетворювачів і репозиторіїв даних датчиків для комфортної доступності виявлення і використання їх через Інтернет.

Всі ці мови опису моделі датчиків, служби спостереження, планування і вимірювання мають власні значні мінуси у вигляді:

* обмеження методів передачі та обміну даних на базі XML
* відсутність нормальної семантичної сумісності

Дані недоліки були вирішені після створення Sensor Model Language (SensorML) на базі SWE, розробкою якої зайнявся Консорціум Всесвітньої Павутини (W3C).

SensorML надає семантичні засоби, що надійно пов'язують між собою процеси та компоненти обробки, які виникають під час вимірювання та подальшого перетворення досліджуваних даних. Рішення включає в себе датчики і виконавчі механізми, процеси спостереження використовувані до будь-яких вимірювань і після них.

Основна суть технології полягає в тому, що є 2 рівня: синтаксичний і семантичний.

Синтаксичний рівень - складається з опису виконуючих механізмів, датчиків і їх мережі, що організовується через API, які в свою чергу визначаються відкритими стандартами OGC.

Семантичний рівень - складається зі стандартів семантичної мережі (RDF, OWL, SPARQL), які представляють одержувані дані з синтаксичного рівня через API як центральний елемент всієй архітектури.

Такий підхід дозволяє використовувати семантичні сенсорні мережі в багатьох сферах діяльності людини, де є можливість організації автоматизованої мережі для збору даних та аналізу для покращення виробництва, зменшення витрат на промислові процеси або створення комфортних умов для вирощування рослин з допомогою моніторингу складових ґрунту, температури, вологості повітря, швидкості вітру, вимірювання тиску, ймовірності опадів.

## Багатосортні мови прикладної логіки

Мова прикладної логіки - мова обчислення предикатів, перш за все необхідна для двох речей: по-перше, є відмінним засобом для використання в логічному програмуванні, так як маючи модель опису предметної області на основі логічно-орієнтованої мови, можна набагато простіше використовувати бази знань і механізми вилучення даних знань для вирішення поставлених завдань, по-друге, як можливість об'єднати всі класифікації, міркування та принципи логічної дедукції в єдину формальну систему.

Семантичний базис класичної мови обчислення предикатів описаний у статті [2] (множина символів мови, семантика яких не залежить від інтерпретації символів сигнатури), необхідний для функціонування мови опису станів породжуючого процесу (або множина формул) повинен володіти властивістю повноти обчислювальних предикатів, які містять кінцеву множину правил виведення. Також до семантичного базису мови входять такі множини, як:

* Області допустимих значень змінних (ОДЗ);
* Області визначення функцій і предикатів, що являются декартовими похідними;
* Множини функціональних відображень (тлумачення функціональних та предикатних символів).

Даний семантичний базис формує пропозиційні зв'язки і квантори.

Пропозиційні зв’язки – задіюються у якості функції союзів та союзних слів у логіці висловлювань (заперечення, кон'юнкція, диз'юнкція), а квантори – використовуються у якості “кількісних” слів, як у звичайній мові (“усе”, “будь-який”, “единий”).

В результаті можна зробити висновок, що існує велика кількість різних логічних мов, які можуть відрізнятися один від одного, типом семантичного базису: обмежений для опису формул і розширений для формалізації ідей. Мови з обмеженим синтаксичним базисом використовуються в межах сфери математичної логіки, а з розширеним у прикладній математиці.

Відповідно для написання будь-якої логічної моделі опису онтології предметної області необхідно вибрати один з двох можливих шляхів використання логічної мови прикладної логіки:

1. Детермінувати і закріпити широкий спектр семантичного базису мови або іншими словами розширити додаток мови, однак це може привести до необхідності створення версій логічної мови, де відбувається постійне розширення семантичного базису і додавання нових понять для попередніх версій.
2. Максимально наблизити визначення семантичного базису формалізованої мови до семантичного базису звичайної мови, а також допустити можливості деякого розширення семантичних категорій для конкретних частин формалізованої системи.

Допустимі два класи елементів для розширення семантичного базису:

* Елементи першого класу – елементи, що краще не використовувати для вже побудованих розширень ядра семантичного базису, описуються у стандартних та спеціалізованих розширеннях мови як і форма для опису ядра мови.
* Елементи другого класу – елементи, що можно і потрібно використовувати у вже розширеному семантичному базисі, так як вони природно визначені цими розширеннями.

Кожна конкретна мова прикладної логіки складається з речень логічної теорії, які зіставляють іменам елементів їх значення або ж сорту, також можлива установка обмежень для порівняння імен з їх інтерпритацією якщо на них впливають значення інших імен логіки.

## Методологія Coras

Важливість методів аналізу ризиків в нашому цифровому просторі висока як ніколи, адже знаходження ризиків в будь-якій технічній системі або мережі дозволяє поліпшити її для безпечного використання насамперед для користувачів. Превентивні заходи оцінки ризиків дозволяють ефективно ними управляти і є однією з обов'язкових умов роботи складних систем.

Віповідно до статті [3] мова моделювання ризику безпеки CORAS є графічною мовою, налаштованою для аналізу загроз і сценаріїв ризиків. Програмне забезпечення CORAS використовує мову UML - мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення.

Сама мова складається із сукупності методів проведення аналізу ризиків:

* Event-Tree-Analysis (Аналіз дерева подій) - метод, який створений для визначення можливих наслідків і ймовірностей (середньорічних частот) виникнення деякої відправної події;
* HazOp - метод визначення небезпечних факторів при експлуатації існуючих пристроїв системи і проектуванні нових об'єктів для неї. В основі дослідження лежить штучне створення "відхилень" виробничих характеристик від зазначених у попередньому проектуванні системи;
* FMECA - (Аналіз видів, наслідків і критичності відмов) це невід'ємна частина аналізу надійності системи дозволяє ідентифікувати різні види можливих відмов, їх наслідки і можливі причини виникнення.

CORAS використовується як підтримка мозкового штурму при визначенні та оцінки ризиків безпеки. Тобто дозволяє фахівцям з різних областей брати участь у проектуванні одного середовища за допомогою уявлень їх професійних термінів через інструменти мови моделювання.

Для аналізу конкретних аспектів досліджуваної системи методологія має комплекс діаграм, що описують події, загрози та можливі уразливості безпеки системи.

Серед таких інструментів є діаграми виду:

* Діаграми активів;
* Діаграми загроз;
* Діаграми ризиків;
* Діаграми рішень.

Відповідно для доповнення опису діаграм існують такі елементи:

* Майно (Asset) - те, що має значення для зацікавлених сторін (учасників) і потребує захисту;
* Учасник (Party) - люди або організації, безпека інформації яких може бути порушена або які можуть вплинути на оцінку об'єкта аналізу;
* Випадкова загроза (Accidental threat) - загроза людського походження, яка здійснена абсолютно випадково і не мала на меті нашкодити системі;
* Навмисна загроза (Deliberate thread) - загроза людського походження, яка здійснена навмисно і мала на меті нашкодити системі або отримати конфеденційну інформацію учасників системи;
* Нелюдська загроза (Non-human threat) - загроза нелюдського походження;
* Вразливість (Vulnerability) - уразливість, яка може бути використана загрозою;
* Сценарій загрози (Threat scenario) - те, що може статися через загрозу використання вразливостей;
* Небажаний інцидент (Unwanted incident) - те, що необхідно запобігти, оскільки воно може безпосередньо завдати шкоди активам учасників;
* Сценарій виправлення (Treatment scenario) - спосіб боротьби з ризиком;
* Ризик (risk) - ймовірність того, що якась подія вплине на активи учасників.

## Онтологія HSSN

Онтологія HSSN – онтологія, що розширює онтологію семантичної сенсорної мережі (SSN). Використання HSSN допомогає збагатити поняття датчиків, їх досліджуваних даних та середовищ розгортання.

Основною перевагою HSSN, яка описана статтею [4] є можливість створення гібридних сенсорних мереж, тобто мереж з мобільними або статичними датчиками, що мають в своєму арсеналі різний комплекс властивостей: від типу вимірюваної величини до визначення чутливості і точності вимірювань.

Опис різновидів датчиків:

* Мобільний сенсор (Mobile Sensor) – відповідно до назви, це датчик, що має можливість змінювати місце свого розташування у інфраструктурі;
* Статичний сенсор (Static Sensor) – не може змінювати місце свого розташування у інфраструктурі.

Опис прикладів властивостей розташування датчиків та пристроїв мережі:

* “В даний час знаходиться у” (isCurrentlyLocatedAt) - властивість застосовується як до статичних, так і до мобільних датчиків і визначає зв'язок датчика з поточним місцем його установки;
* “Раніше знаходився у” (hasPreviouslyLocated) – властивіть застосовується тільки до мобільних датчиків і визначає відносини між попередньою локацією розміщення мобільного датчика та самим датчиком;
* “Має місце розміщення” (has deployment) – властивість застосовується до усіх девайсів і необхідна для визначення точного місця розгортання пристрою (прилад розташован на стіні конкретной кімнати або платформою розгортання служить інший девайс).

Датчики мають властивості для описання процесів вимірювання певной величини на яку вони налаштовані.

Опис прикладів властивостей вимірювань датчиків:

* “В даний час охоплює” (currentlyCovers) – властивість, що являє собою зв’язок між охопленим сектором вимірювань та самим датчиком (властивість обов’язкова так як допомогає чітко уявити масштаби можливостей збору інформації датчиків і додатково обґрунтувати попередні властивості місця розташування);
* “Дальність вимірювань” (Sensing Range) – властивість, що безпосередньо пов’язана з попередньою властивістю і додатково розширює уявлення про зону покриття датчика шляхом вказівки можливої для датчика дальності вимірювань.

Додатково хочеться відзначити також важливі класи представлення платформного розмаїття - середовища розміщення пристроїв, тобто визначення їх фізичного оточення як цілісної інфраструктури або навіть опис однієї інфраструктури в іншій більш великій (комплексні інфраструктури).

У кожного датчика крім його фізичних властивостей є властивості визначення величини вимірювання - це може бути скалярна величина, за типом температури або вологості повітря, а також різні мультимедійні властивості досліджуваного об'єкта і онтологія HSSN надає широкий спектр для опису вимірюваних даних:

* Аудіо;
* Відео;
* Зображення;
* Скалярні величини та їх властивості.

За підсумком можна сказати, що онтологія HSSN дозволяє описувати величезну кількість варіантів предметних областей для різних сенсорних мереж, від невеликих систем приватного будинку до великих комплексів досліджень, що робить її універсальною при роботі з гібридними сенсорними мережами. Розширення семантичного представлення даних, платформ розміщення, датчиків, їх експлуатаційних можливостей і властивостей тільки поліпшило концепцію семантичних сенсорних мереж і дало додатковий поштовх до розвитку оптимізаційних можливостей для таких мереж і ефективності продуктивного використання в безлічі сфер діяльності людини.

## Застосування онтологій для опису середовищ предметних областей

На сьогоднішній день розробка якісної семантичної сенсорної мережі під певну предметну область використовуючи онтології є дуже трудомістким і складним процесом особливо якщо намагатися проектувати таку мережу з нуля. Тому підхід до використання вже заздалегідь створених онтологій, які тим чи іншим чином допоможуть розширити класифікацію і спростити процес формалізації семантики для власної предметної області, є ефективним і рекомендованим.

SAREF (Smart Appliances Reference) – онтологія-еталон для пристроїв у сферах IoT і M2M (Machine-to-Machine - це провідні і бездротові системи датчиків, які передають інформацію від одного пристрою іншому) та відносяться до класу пристроїв, що використовуються звичайними споживачами в побутовій електроніці.

SAREF поділена на три основні розділи:

1. Пристрої;
2. Функції пристроїв;
3. Можливе споживання/виробництво енергії.

Device – клас SAREF описуючий пристрої як матеріальні об'єкти, що призначені для виконання певного завдання в домашніх господарствах, звичайних громадських будівлях або офісах.

Service – клас SAREF це представлення функції пристрою в мережі, що робить функцію доступною для виявлення, реєстрації та віддаленого керування іншими пристроями в мережі.

Energy related – клас-категорія SAREF, який розглядає пристрої на основі інформації про споживання енергії та профілів для оптимізації енергоефективності.

Сама по собі онтологія дуже обмежена в наданому функціоналі для опису предметних областей, адже вона є базовою для багатьох інших онтологій, які її модифікують під певні потреби розробника.

Також є онтології за типом BOnSAI, яка описує можливі послуги та веб-сервіси для середовища приватних будівель. Однак онтологія знову ж таки має занадто обмеженим функціоналом для використання в будь-яких комерційних системах автоматизації.

Онтологія SSN (Semantic Sensor Network) – це теж одна з базових онтологій як і SAREF, але більше робить упор на процеси моніторингу та збору інформації за допомогою датчиків. Дозволяє описати процес від надходження сирих вхідних даних стимулу (подразник процесу дослідження для датчиків) до використання вже оброблених даних для маніпулювання навколишнім середовищем або аналізу і створення моделей вирішення певних завдань характерних для спроектованої сенсорної мережі.

SSN більш просунута онтологія ніж SAREF, але їй також необхідні модифікації для розширення запропонованої семантики.

Такими розширеннями є:

* M3 (Machine-to-Machine Measurement) - дозволяє розробляти сумісні програми для конкретної предметної області або міждоменної семантичної мережі речей (SWoT). Основні цілі даной онтології-допомогти розробникам у швидкому прототипуванні своїх додатків і скоротити час навчання інтеграції технологій семантичних мереж;
* OntoSensor – онтологія, для датчиків з запитами та виведенням інформації, основана на SensorML;
* OpenIoT – онтологія, для створення описів віртуальних датчиків і автоматизації процесу їх семантичної анотації та реєстрації, доповнює онтологію SSN додатковими поняттями характерными для IoT додатків.

# Завдання на розробку

## Визначення використовуваного обладнання та основних цілей сенсорних мереж систем опалення

Автоматична система опалення, що базується на основі сенсорної мережі на даний час є однію з самих ефективних і надійних з точки зору комфортного регулювання температури в приміщеннях без або мінімуму участі людини в цьому процесі. Так підтримка певного температурного режиму у жилих приміщеннях є нашою повсякденністю і необхідністю тому нам як споживачам такої системи досить важливо вибрати якісне обладнання для зручного перебування у власних домах та квартирах, де температура приміщень може суттєво впливати навіть на нашу ефективність роботи.

Існують різні системи, які дозволяють нам виконати це завдання: такі як водяне, повітряне та інфрачервоне опалення і т.п.

В більшості будинках при проектувані системи опалення застосовують водяне.

Основними елементами водяної системи опалення є:

* Саме джерело теплопостачання – джерело від якого ми отримуємо тепло (котел, бойлер, тепловий насос і їх різновиди в залежності від палива).
* Трубопровід системи опалення – виконує роль системи циркуляції теплоносія для обігріву приміщень
* Опалювальний прилад – передає тепло від джерела теплопостачання у приміщення. Радиатор, фанкойл та тепла підлога є основними представниками опалювальних приладів і кожен має власні види в залежності від теплоносія системи. Передачі тепла здійснюються за різними сценаріями в залежності від загального виду системи опалення: конвективна або промениста.

Автоматичне управління системою опалення має дві задачі:

1. Підтримувати відповідну розрахункову температуру в заданому приміщенні за допомогою комплексу пристроїв, де головним є контролер мікроклімату;
2. Ефективно економіти енергоресурси та робочі ресурси пристроїв системи.

Основні види терморегулювання в системах опалення такі:

1. Якісний – зміна температури теплоносія;
2. Кількісний – відповідно зміна кількості теплоносія, що поставляється до опалювальних приладів.
3. Спільні методи терморегулювання.

При проектувані системи було обрано рішення використовувати одразу 2 види терморегулювання для досягнення максимальної продуктивності. Якісний метод виконується за допомогою бойлеру, а кількісний – колектором та сервоприводами, які працюють у парі.

Регулюванням температури у сенсорних мережах систем опалення виконується за допомогою термодатчиків, що можуть збирати інформацію як зовні так і всередині приміщень для передачі цих даних до термостату, а потім і до контролеру мікроклімату який на підставі різниць температур приймає рішення подати сигнал джерелу теплопостачання щодо зміни температури теплоносія у необхідному диапазоні.

Сучасні системи автоматичного електронного управління опаленням вже давно вийшли за рамки класичного управління мікрокліматом в приміщенні. Бездротові системи опалення дозволяють налаштовувати і виконувати моніторинг за станом системи через інтернет за допомогою смартфону користувача, а також автоматично змінювати свою поведінку в залежності від пори року і багатьох інших умовах

У якості моделей приладів сенсорної системи опалення були обрані моделі компанії Salus, що надає досить широкий спектр вибору серед пристроїв опалювального сектору, простоту використання приладів, простий монтаж додаткових модулів у систему, зрозумілу простому користувачу функціональність пристроїв та зрозуміла для проектування схема роботи взаємодії приладів як між собою так і з зовнішніми приладами по типу датчиків температури повітря або теплоносія та різних виконуючих механізмів.

## Багаторівнева модель онтології

Дана робота ставить за мету створення багаторівневої онтології сенсорної мережі системи опалення із застосуванням багатосортної мови прикладної логіки та розробленої моделі оцінки ризиків, що необхідна для з’ясування того які загрози можуть виникнути в мережі, як вони впливають на пристрої системи, а також які компоненти пристроїв підлягають зношенню в результаті утворення вразливостей після впливу загроз.

Використання даних підходів дозволяє логічно пов'язувати між собою описи різних пристроїв, що є учасниками сенсорної мережі і їх взаємодії один з одним.

Також можливо пов'язувати будь-який компонент пристрою мережі з його необхідністю в технічному обслуговуванні, що наприклад в результаті такого зв'язку він стає базою для створення моделі оцінки ризиків технічного стану сенсорної мережі.

Відповідно для досягнення поставленої мети необхідно вирішити наступні задачі:

* ввести основні терміни і сорти для опису модуля сенсорної мережі за допомогою мови прикладної логіки;
* ввести основні терміни та сорти для опису модуля оцінки ризиків за допомогою мови прикладної логіки;
* розробити модель онтології сенсорної мережі системи опалення для приватного будинку;
* розробити таблицю можливих пошкоджень пристроїв системи;
* розробити модель оцінки ризиків;
* розробити діаграму загроз сенсорної мережі на основі методології Coras;
* описати діаграму загроз сенсорної мережі за допомогою введення необхідних термінів.

Для даной роботи був розроблений модульний підхід для проектування моделі онтології опису сенсорної мережі системи опалення. В основу розробленого модульного підходу покладено створення багаторівневої моделі онтології, що складається з сукупності модулей.

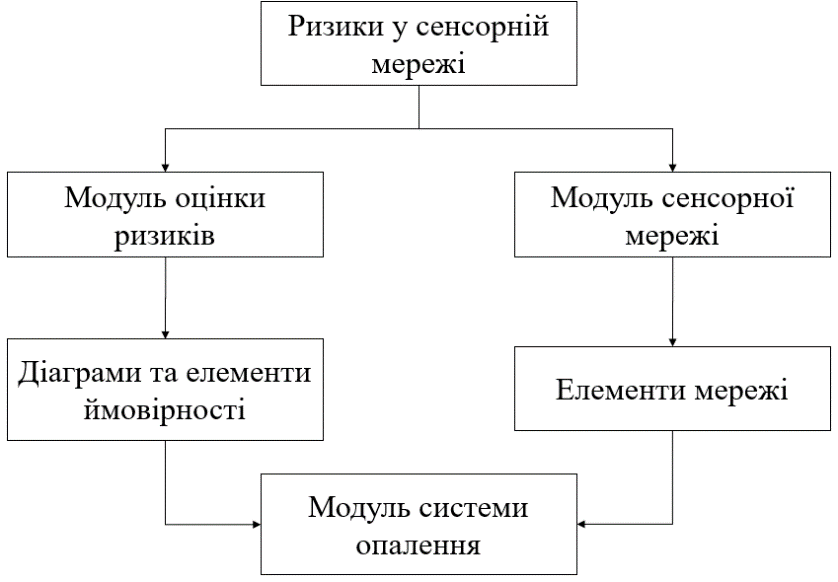


Рисунок 2.1 - Ієрархія багаторівневої онтології

Модуль оцінки ризику та модуль сенсорної мережі є модулями другого рівня, що описують основні поняття для своїх предметних областей. Використання таких модулей дозволяє розширяти при необхідності вже формалізовану мову прикладної логіки та вводити нові модулі для більш розгорнутого описання інших задач якщо необхідно.

Для створення баз знань для модулів багаторівневої онтології необхідно використати програмне забезпечення Protégé, що є редактором онтологій. Конкретно для даної роботи треба скористатися редактором Protégé-OWL, що дозволить будувати онтологію типу OWL з використанням всіх необхідних атрибутів для описання класів, їх властивостей, що визначають відношення між класами та створення відповідних екземплярів класів.

# ПРОЕКТУВАННЯ БАГАТОРІВНЕВОЇ ОНТОЛОГІЇ СЕНСОРНОЇ МЕРЕЖІ СИСТЕМИ ОПАЛЕННЯ

## Прикладна логіка модуля системи опалення

Для початку проектування багаторівневої онтології сенсорної мережі системи опалення необхідно за допомогою мови прикладної логіки ввести терміни і сорта для метаонтології 4-рівня, тобто це онтологія найвищого рівня, що по своїй суті є абстракцією основних об’єктів онтології нижчих рівнів і на яку вони будуть посилатися при описанні.

Вводимо терміни, що будуть являтися фрагментами онтології 4-ого рівня:

* Область допустимих значень: {}(R  I  N  L);
* Типи об’єктів: {}N \ {};
* Тип: {}(R  I  N  L)
* Типи компоненту об'єкта: типи об'єктів → {} типи об'єктів
* Множина об'єктів ≡ {(Тип: типи об'єктів) j(Тип)}
* Множина значень = {} (R  I  N  L)
* Відношення об'єктів = (λ (Тип: типи об'єктів → типи об'єктів) → {}множина значень )
* Властивості об'єктів ≡ (λ (Тип: Типи об'єктів)(λ(Область можливих значень: {} (Множина значень)) (j (Тип) → Область можливих значень))

Далі приступимо до модулю системи опалення, що є модулем першого рівня, тобто найнижчого рівня, що містить основні терміни і сорти, які максимально змістовно описують сенсорну мережу системи опалення з точки зору фізичних властивостей та можливостей кожного пристрою, а також їх роль і відношення в системі між собою. (Повне описання прикладної логіки модуля системи опалення визначене у додатку А)

Основні терміни сукупностей прикладної логіки:

Термін *Actuation* - позначає сукупність методів приведення в дію виконавчих механізмів виконуючих деякі процедури визначені у множині *Procedure* для зміни стану елементів системи.

сорт *Actuation* = {}N \ {}.

Термін *Capability –* позначає сукупність функціональних можливостей, що надається и визначається множиною *Service*, описується його можливостями.

сорт *Capability* ={}N \ {}.

Термін *Connections* – позначає сукупність з’єднань усіх пристроїв системи між собою.

сорт *Connections* ={}N \ {}.

Термін *Deployment* – позначає сукупність розгортань однієї або декількох систем для певної мети на платформі визначених множиною *Platform*

сорт *Deployment* ={}N \ {}.

Термін *ExpansionCard* – позначає сукупність підключаємих карт розгортання на платформах визначених множиною *Platform* у якості додаткових модулей.

сорт *ExpansionCard* ={}N \ {}.

Термін *Hardware* – позначає сукупність фізичних компонентів пристроїв, що відповідають за певні апаратні можливості.

сорт *Hardware* ={}N \ {}.

Термін *Software* – позначає сукупність всього програмного забезпечення (додатків і програм) запущених на пристрої.

сорт *Software* = {}N \ {}.

Термін *Interface* –позначає сукупність інтерфейсів, що виконують функцію зв’язку та визначають як зі службою можуть зв'язатися/підключитися інші об'єкти, такі як особи, які запитують послуги, або інші служби.

сорт *Interface* = {}N \ {}.

Термін *Memory* – позначає сукупність всіх запам'ятовуючих пристроїв приладу.

сорт *Memory* = {}N \ {}.

Термін *Observation* – позначає сукупність актів проведення процедури спостереження за об’єктами інтересу для оцінки або розрахунку вартості властивостей цих об’єктів, що є цікавими у рамках спостереження.

сорт *Observation* = {}N \ {}.

Термін *Platform* – позначає сукупність об'єктів, в/на яких розміщуються інші об'єкти, зокрема датчики, виконавчі механізми, інші платформи.

сорт *Platform* = {}N \ {}.

Термін *Device* – позначає сукупність електронних обладнань, здатне виконувати одну або кілька обчислювальних функцій і підтримує установку вбудованого програмного забезпечення або програмного забезпечення сторонніх виробників.

сорт *Device* = {}N \ {}.

Термін *Infrastructure* – позначає сукупність фізичних інфраструктур або середовищ реального світу, в яких можуть бути розгорнуті мережі датчиків.

сорт *Infrastructure* = {}N \ {}.

Термін *Procedure* – позначає сукупність робочих процесів, протоколів, планів, алгоритмів або обчислювальних методів, що визначають, як проводити спостереження або вносити зміни в стан світу. У ній пояснюються кроки, які необхідно виконати для отримання відтворюваних результатів.

сорт *Procedure* = {}N \ {}.

Термін *ProcedureInformation* – позначає сукупність вхідних та вихідних даних необхідних для роботи певних робочих процесів, протоколів, алгоритмів.

сорт *ProcedureInformation* = {}N \ {}.

Термін *Processor* – позначає сукупність апаратного забезпечення, що відповідає за обробку даних в пристроях.

сорт *Processor* = {}N \ {}.

Термін *Property* – позначає сукупність якостей сутності. Аспект сутності, який притаманний сутності і не може існувати без неї.

сорт *Property* = {}N \ {}.

Термін *Maintenance Schedule* – позначає сукупність графіків технічного обслуговування кожного пристрою системи в заданих умовах.

сорт *Maintenance Schedule* = {}N \ {}.

Термін *Operating Range* – позначає сукупність нормальних робочих властивостей систем при деяких заданих умовах, в яких очікується робота системи

сорт *Operating Range* = {}N \ {}.

Термін *SensorProperty* – позначає сукупність властивостей датчиків для збору інформації з навколишньої середи

сорт *SensorProperty* = {}N \ {}.

Термін *SensingAngle* – позначає сукупність максимальних горизонтальних та вертикальних кутів вимірювання, що забезпечуються датчиком.

сорт *SensingAngle* = {}N \ {}.

Термін *SensingRange* – позначає сукупність максимальних відстаней виявлення подразника, що забезпечуються датчиком.

сорт *SensingRange* = {}N \ {}.

Термін *Survival Range* – позначає сукупність властивостей виживання системи при деяких заданих умовах, в яких система може виконувати покладені функції без пошкоджень.

сорт *Survival Range* = {}N \ {}.

Термін *Service* – позначає сукупність служб, що являють собою комплексні програмні об’єкти, здатні надавати деякі значення (або набір значень) щодо даного доменого об’єкту.

сорт *Service* = {}N \ {}.

Термін *System* – позначає сукупність одиниць абстракції (систем) для частин інфраструктури, що реалізують певні процедури. Системи можуть мати компоненти їх підсистем, які є іншими системами.

сорт *System* = {}N \ {}.

Термін *Actuator* – позначає сукупність пристроїв, які використовуються або реалізують процедуру (приведення в дію), яка змінює стан світу.

сорт *Actuator* = {}N \ {}.

Термін *Sensor* – позначає сукупність пристроїв, що бере участь у процедурі дослідження або реалізує її. Датчики реагують на подразник (зміна навколишнього середовища; вхідні дані, складені з результатів попередніх спостережень) і генерують результат. Датчики можуть розміщуватися на платформах.

сорт *Sensor* = {}N \ {}.

Основні терміни функцій-властивостей об’єктів:

Термін *hasConnection* – позначає функцію, визначає відношення девайсів або систем до їх з’єднання, тобто функція визначає які фізичні з’єднання має той чи інший девайс або система з іншими приладами. Це відповідно зв’язок відношень екземпляру *Service* або *Device* до екземпляру *Connections*.

сорт  *hasConnection* = *System Device → Connections*

Термін *isConnectedTo* – позначає функцію, що визначає зв'язок між з’єднанням приладу та приладом до якого підключається з’єднання, тобто функція визначає які девайси або пристрої з’єднанні один з одним. Це відповідно зв’язок відношень екземпляру *Connections* до екземпляру *Device* або *System*.

сорт *isConnectedTo* = *Connections → Device System.*

Термін *has property* – позначає функцію, визначає зв'язок між об’єктом інтересу і його властивістю або між системою і її властивістю, тобто об’єкт інтересу або система мають цю властивість. Це відповідно зв’язок відношень екземпляру *Feature of Interest* або *System* до екземпляру *Property*.

сорт  *has property* = *Feature Of Interest System → Property*

Термін *has operating range* – позначає функцію, визначає відношення системи або девайсу до робочого діапазону, що описує нормальне робоче середовище системи, тобто система або девайс мають конкретний робочий діапазон. Це відповідно зв’язок відношень екземпляру *System* або *Device* до екземпляру *Operating Range*.

сорт  *has operating range* = *System Device → Operating Range*

Термін *has survival range* – позначає функцію, визначає відношення системи або девайсу до діапазону виживання, що описує виживання системи при деяких заданих умовах, в яких система може виконувати покладені функції без пошкоджень, тобто система або девайс мають конкретний діапазон виживання. Це відповідно зв’язок відношень екземпляру *System* або *Device* до екземпляру *Survival Range*.

сорт  *has survival range* = *System Device → Survival Range*

Термін *hasHardware* – позначає функцію, визначає співвідношення, що зіставляє системи або девайси з їх апаратним забезпеченням, тобто функція визначає які апаратні забезпечення мають системи або девайси. Це відповідно зв’язок відношень екземпляру *System* або *Device* до екземпляру *Hardware*.

сорт  *hasHardware* = *System Device → Hardware*

Термін *hasSoftware*– позначає функцію, визначає співвідношення, що зіставляє системи або девайси з їх програмним забезпеченням, тобто функція визначає які програмні забезпечення мають системи або девайси. Це відповідно зв’язок відношень екземпляру *System* або *Device* до екземпляру *Software*.

сорт  *hasSoftware* = *System Device → Software*

Термін *hasMaintenanceSchedule* – позначає функцію, визначає відношення девайсу або системи, або об’єкту інтересу системи опалення до їх графіка технічного обслуговування, тобто визначає який прилад або об’єкт в системі має свій графік технічного обслуговування. Це відповідно зв’язок відношень екземпляру *Device* або *System*, або *HeatingSystem* до екземпляру *Maintenance Schedule*

сорт  *hasMaintenanceSchedule* = *Device System HeatingSystem → Maintenance Schedule*

Термін *hosts* – позначає функцію, визначає зв'язок між платформою і датчиком, приладом, або платформою, розміщеної або встановленої на ній, тобто функція визначає пристрої встановлені на платформі. Це відповідно зв’язок відношень екземпляру *Platform* до екземпляру *System* або *Device*.

сорт  *hosts* = *Platform → System Device*

Термін *is hosted by* – позначає функцію, визначає зв'язок між датчиком, приладом, девайсом і платформою, на якій він встановлений або розміщений, тобто функція визначає на якій платформі встановлені пристрої. Це відповідно зв’язок відношень екземпляру *Device* або *System* до екземпляру *Platform*.

сорт *is hosted by* = *Device System → Platform*

Термін *is property of* – позначає функцію, визначає зв'язок між властивістю і об'єктом системи опалення, якому воно належить, тобто функція визначає якому об’єкту системи опалення належить конкретна властивість. Це відповідно зв’язок відношень екземпляру *Property* до екземпляру *Feature Of Interest* або *Device,* або *System*.

сорт *is property of* = *Property* *→ Feature Of Interest Device System*

Термін *is observed by* – позначає функцію, визначає зв'язок між властивістю спостереження і датчиком, здатним її спостерігати, тобто функція визначає яким датчиком спостерігається конкретна властивість спостереження. Це відповідно зв’язок відношень екземпляру *Observable Property* до екземпляру *Sensor*.

сорт *is observed by* = *Observable Property → Sensor*

Термін *made actuation* – позначає функцію, визначає зв'язок між виконуючим механізмом і приведенням в дію, яке він виконав. Це відповідно зв’язок відношень екземпляру *Actuator* до екземпляру *Actuation*.

сорт  *made actuation* = *Actuator* *→ Actuation*

Термін *made by actuator* – позначає функцію, визначає зв'язок між приведенням в дію, яке виконуючий механізм виконав і ним самим. Це відповідно зв’язок відношень екземпляру *Actuation* до екземпляру *Actuator*.

сорт  *made by actuator* = *Actuation* *→ Actuator*

Термін *made observation* – позначає функцію, визначає зв'язок між спостереженням і датчиком, що виконує його. Це відповідно зв’язок відношень екземпляру *Sensor* до екземпляру *Observation*.

сорт  *made observation* = *Sensor* *→ Observation*

Термін *made by sensor* – позначає функцію, визначає зв'язок між датчиком, що виконує спостереження і ним самим. Це відповідно зв’язок відношень екземпляру *Observation* до екземпляру *Sensor*.

сорт  *made by sensor* = *Observation* *→ Sensor*

Термін *provides* – позначає функцію, що зіставляє девайс зі службою яку він забезпечує. Це відповідно зв’язок відношень екземпляру *Device* до екземпляру *Service*.

сорт  *provides* = *Device* *→ Service*

Основні терміни функцій-даних об’єктів:

Термін *ConnectionType\_ToDevice* – позначає функцію, що визначає дані для типу з’єднання до девайсів системи опалення у вигляді тексту (xsd:string). Це відповідно зв’язок відношень екземпляру *Connections* до сукупності усіх множин.

сорт  *ConnectionType\_ToDevice* = *Connections →* {}N \ {}*.*

Термін *ConnectionType\_ToSystem* – позначає функцію, що визначає дані для типу з’єднання до систем системи опалення у вигляді тексту (xsd:string). Це відповідно зв’язок відношень екземпляру *Connections* до сукупності усіх множин.

сорт  *ConnectionType\_ToSystem* = *Connections →* {}N \ {}*.*

Термін *define\_Maintenance\_Time\_For\_Systems* – позначає функцію, що визначає дані для значення наступного технічного обслуговування пристрою системи опалення у вигляді інформації про дату та час (xsd:dateTime). Це відповідно зв’язок відношень екземпляру *Maintenance Schedule* до сукупності усіх множин.

Сорт *define\_Maintenance\_Time\_For\_Systems* = *Maintenance Schedule →* {}N \ {}*.*

Термін *define\_Previous\_Maintenance\_Time\_For\_Systems* – позначає функцію, що визначає дані для значення попереднього технічного обслуговування пристрою системи опалення у вигляді інформації про дату та час (xsd:dateTime). Це відповідно зв’язок відношень екземпляру *Maintenance Schedule* до сукупності усіх множин.

Сорт *define\_Previous\_Maintenance\_Time\_For\_Systems* = *Maintenance Schedule →* {}N \ {}*.*

Термін *define\_Max\_OperatingTemperature* – позначає функцію, що визначає дані для значення максимальної робочої температури пристрою системи опалення у вигляді цілого числа (xsd:int). Це відповідно зв’язок відношень екземпляру *Operating Range* до сукупності усіх множин.

Сорт *define\_Max\_OperatingTemperature* = *Operating Range →* {}N \ {}*.*

Термін *define\_Min\_OperatingTemperature* – позначає функцію, що визначає дані для значення мінімальної робочої температури пристрою системи опалення у вигляді цілого числа (xsd:int). Це відповідно зв’язок відношень екземпляру *Operating Range* до сукупності усіх множин.

Сорт *define\_Min\_OperatingTemperature* = *Operating Range →* {}N \ {}*.*

Термін *define\_Max\_StorageTemperature* – позначає функцію, що визначає дані для значення максимальної температури зберігання для пристрою системи опалення у вигляді цілого числа (xsd:int). Це відповідно зв’язок відношень екземпляру *Survival Range* до сукупності усіх множин.

Сорт *define\_Max\_StorageTemperature* = *Survival Range →* {}N \ {}*.*

Термін *define\_Min\_StorageTemperature* – позначає функцію, що визначає дані для значення мінімальної температури зберігання для пристрою системи опалення у вигляді цілого числа (xsd:int). Це відповідно зв’язок відношень екземпляру *Survival Range* до сукупності усіх множин.

Сорт *define\_Min\_StorageTemperature* = *Survival Range →* {}N \ {}*.*

Термін *define\_SystemPowerSupply* – позначає функцію, що визначає дані для значення електропостачання пристрою системи опалення у вигляді тексту (xsd:string). Це відповідно зв’язок відношень екземпляру *Operating Range* до сукупності усіх множин.

Сорт *define\_SystemPowerSupply* = *Operating Range →* {}N \ {}*.*

Термін *define\_SystemPowerSupply\_Type* – позначає функцію, що визначає дані для типу джерела живлення пристрою системи опалення у вигляді тексту (xsd:string). Це відповідно зв’язок відношень екземпляру *Operating Range* до сукупності усіх множин.

Сорт *define\_SystemPowerSupply\_Type* = *Operating Range →* {}N \ {}*.*

Термін *NumberOf\_InstalledSystems* – позначає функцію, що визначає дані для значення кількості встановлених систем у вигляді цілого числа (xsd:int). Це відповідно зв’язок відношень екземпляру *System* до сукупності усіх множин.

Сорт *NumberOf\_InstalledSystems* = *System* *→* {}N \ {}*.*

## Прикладна логіка модулю онтології оцінки ризиків

Модуль оцінки ризиків є модулем другого рівня, що містить основні терміни і сорти, які описують сутності-елементи, складові сутності та відносини між цими термінами, що пов’язують їх один з одним.

Якщо почати описання модуля онтології оцінки ризиків, то необхідно описати онтологію з точки зору декількох рівнів: другого та третього.

Вводимо терміни, що будуть являтися фрагментами онтології 3-ого рівня:

* Діаграми = {}N \ {};
* Елементи діаграми = {}N \ {};
* Відношення = {}N \ {};

Також необхідно ввести деякі речення прикладної логічної теорії, що будуть детальніше описувати третій рівень онтології оцінки ризиків:

* Відношення елементів діаграми = Типи Відношення об'єктів \ Ø;
* Типи об'єктів діаграми: {} Типи об'єктів \ Ø;
* Властивості діаграми ≡ (λ (Тип: властивості об'єктів) (λ (Область можливих значень: {} (Множини значень));
* Типи об'єктів ≡ {діаграми, елементи діаграми, відносини};
* Типи об'єктів діаграми ≡ (елементи діаграми).

Вводимо терміни, що будуть являтися фрагментами онтології 2-ого рівня:

* сорт Ймовірність виникнення Сценарію загрози: Сценарій загрози → властивості компонентів діаграми, (R (0, 1))
* сорт Ймовірність виникнення Небажаної події: Небажана подія → властивості компонентів діаграми, (R (0, 1))
* сорт Ймовірність виникнення Ризику: Ризик → Властивості компонентів діаграми, (R (0,1))

## Прикладна логіка модулю онтології сенсорної мережі

Модуль сенсорної мережі є модулем другого рівня, що містить основні терміни і сорти, які в загальному сенсі описують складові сенсорної мережі для подальшої оцінки ризиків цієї мережі

Якщо почати описання модуля онтології сенсорної мережі, то необхідно описати онтологію з точки зору декількох рівнів: другого та третього.

Основні визначення сенсорної мережі:

* Мережевий адаптер – інтерфейс, що виконує завдання фізичної взаємодії мережевих пристроїв.
* Пристрій - фізичний механізм використовує мережевий адаптер для здійснення передачі одиниці інформації з іншими пристроями.
* Провідне середовище - фізичне середовище за допомогою якої функціонують (передають інформацію) мережеві адаптери.
* Мережа - цілісна сутність, яка визначає множину, що складається з мережевих адаптерів, провідного середовища і пристроїв.
* Навколишнє середовище - множина об'єктів, які не входять в мережу, але побічно взаємодіють з нею тим чи іншим чином.

На основі вище зазначених визначень вводимо терміни, що будуть являтися фрагментами онтології 3-ого рівня:

* Мережевий адаптер ⸦ {}N \ {};
* Провідне середовище ⸦ {}N \ {};
* Мережа ⸦ {}N \ {};
* Навколишнє середовище ⸦ {}N \ {};
* Пристрій ⸦ {}N \ {}.

Вводимо речення прикладної логічної теорії, що будуть детальніше описувати третій рівень онтології сенсорної мережі:

* Типи елементів мережі: {} Типи об'єктів \ Ø
* Відносини елементів мережі: {} Типи відношення об'єктів
* Типи об'єктів ≡ { мережевий адаптер (NIC), провідне середовище (Wiring), мережа (Network), навколишнє середовище (Environment), пристрій (Device)}
* Тип Елемент мережі ≡ (λ (Тип: {типи об'єктів, відносини елементів мережі}) (Тип ≠ відносини елементів мережі Ø )) )

Вводимо терміни, що будуть являтися фрагментами онтології 2-ого рівня:

* сорт Вхідні дані: Мережевий адаптер → Властивості об'єктів (R (0,∞))
* сорт Вихідні дані: Мережевий адаптер → Властивості об'єктів (R (0,∞))
* сорт Пошкодження провідного середовища: Провідне середовище → Властивості об'єктів (R (0,∞))
* сорт Пошкодження компонентів пристроїв: Пристрій → Властивості об'єктів (R (0,∞))
* сорт Кількість вузлів: Мережа → Властивості об'єктів (N (0,∞))
* сорт Вхідні дані системних пристроїв: Навколишнє середовище → Властивості об'єктів (R (0,∞))
* сорт Вихідні дані системних пристроїв: Навколишнє середовище → Властивості об'єктів (R (0,∞))

# Використання БАГАТОРІВНЕВОЇ ОНТОЛОГІЇ СЕНСОРНОЇ МЕРЕЖІ СИСТЕМИ ОПАЛЕННЯ

## Онтологія найнижчого рівня на основі прикладної логіки системи опалення (SHSO)

Описання онтології сенсорної системи опалення (SHSO – Sensor Heating System Ontology) на основі онтології HSSN починається з використання опису прикладної логіки предметної області як бази для описання ієрархії класів та вузлів їх відношення один з одним. Опис онтології відбувається по методу зі статті [5].

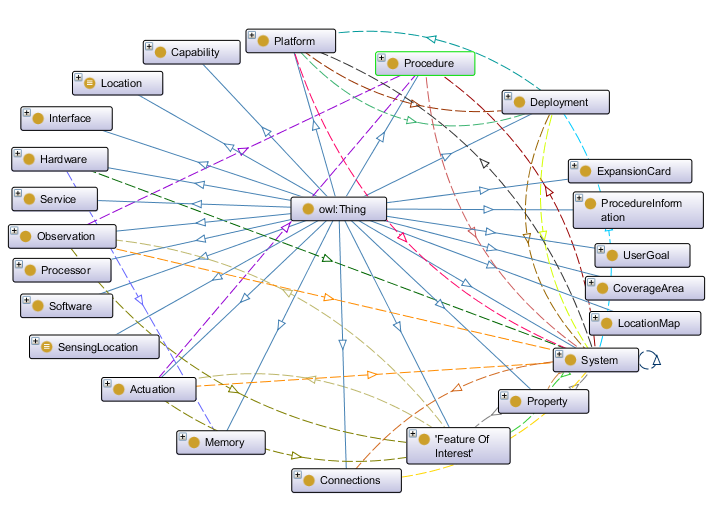


Рисунок 4.1 - Візуалізація ієрархії спроектованної онтології сенсорної мережі системи опалення

Нижче представлено описання класів та їх властивостей в онтології.

Описання основних класів:

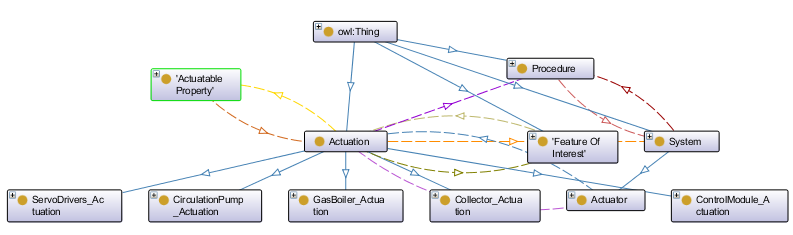


Рисунок 4.2 - Візуалізація класу Actuation та його відношень

* *sosa:Actuation*: Клас методів приведення в дію виконавчих механізмів виконуючих деякі процедури для зміни стану елементів системи.

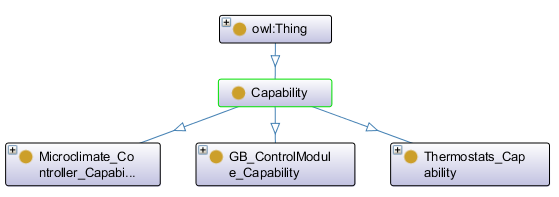


Рисунок 4.3 - Візуалізація класу Capability та його відношень

* *hssn:Capability:* Клас функціональних можливостей пристроїв, що надаються и визначаються класом *Service*, описуються його можливостями.

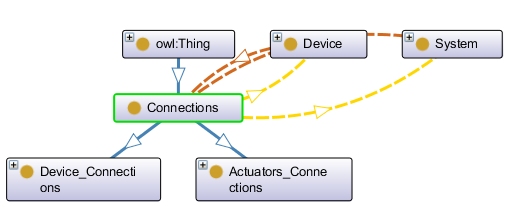


Рисунок 4.4 - Візуалізація класу Connections та його відношень

* *shso:Connections:* Клас, що репрезентує фізичне з’єднання усіх пристроїв системи між собою.

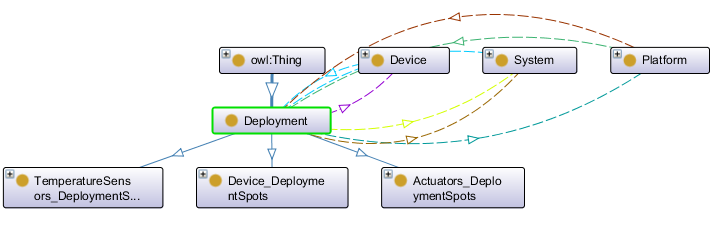


Рисунок 4.5 - Візуалізація класу Deployment та його відношень

* *ssn:Deployment:* Клас розгортань однієї або декількох систем для певної мети на платформі визначених класом *Platform*

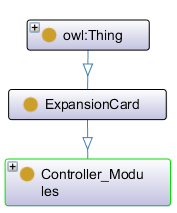


Рисунок 4.6 - Візуалізація класу ExpansionCard та його відношень

* *hssn:ExpansionCard:* Клас підключаємих карт розгортання на платформах визначених класом *Platform* у якості додаткових модулей.

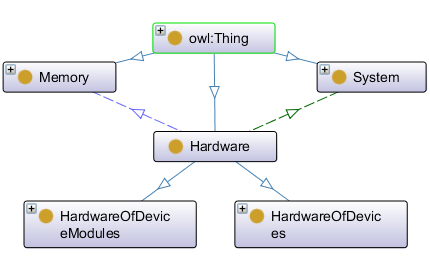


Рисунок 4.7 - Візуалізація класу Hardware та його відношень

* *hssn:Hardware:* Клас фізичних компонентів пристроїв, що відповідають за певні апаратні можливості.

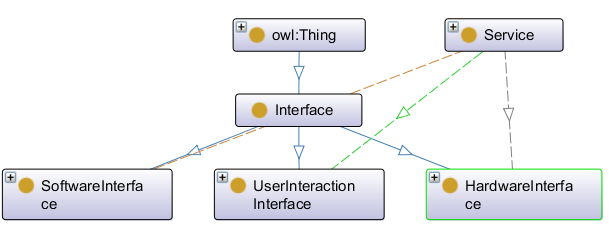


Рисунок 4.8 - Візуалізація класу Interface та його відношень

* *hssn:Interface:* Клас інтерфейсів, що виконують функцію зв’язку та визначають як зі службою можуть зв'язатися/підключитися інші об'єкти, такі як особи, які запитують послуги, або інші служби.

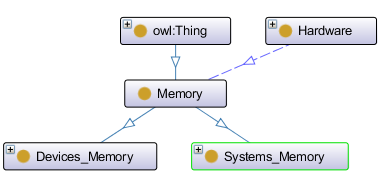


Рисунок 4.9 - Візуалізація класу Memory та його відношень

* *hssn:Memory:* Клас усіх запам'ятовуючих пристроїв приладів.

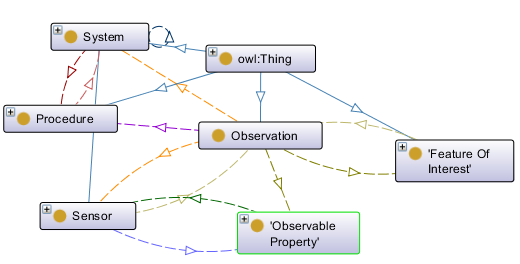


Рисунок 4.10 - Візуалізація класу Observation та його відношень

* *sosa:Observation:* Клас актів проведення процедури спостереження за об’єктами інтересу для оцінки або розрахунку вартості властивостей цих об’єктів, що є цікавими у рамках спостереження.

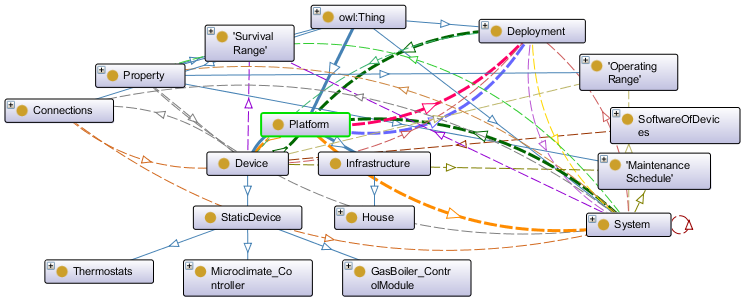


Рисунок 4.11 - Візуалізація класу Platform та його відношень

* *sosa:Platform:* Клас об'єктів, в/на яких розміщуються інші об'єкти, зокрема датчики, виконавчі механізми, інші платформи.



Рисунок 4.12 - Візуалізація субкласу Device та його відношень

* *hssn:Device:* Субклас класу *Platform*, що представляє електронні обладнання, що здатні виконувати одну або кілька обчислювальних функцій і підтримувати установку вбудованого програмного забезпечення або програмного забезпечення сторонніх виробників.

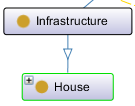


Рисунок 4.13 - Візуалізація субкласу Infrastructure та його відношень

* *hssn:Infrastructure:* Субклас класу *Platform*, що представляє фізичну інфраструктуру або середовище реального світу, в яких можуть бути розгорнуті мережі датчиків.

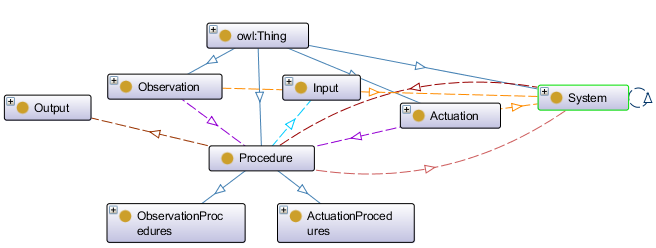


Рисунок 4.14 - Візуалізація класу Procedure та його відношень

* *sosa:Procedure:* Клас робочих процесів, протоколів, планів, алгоритмів або обчислювальних методів, що визначають, як проводити спостереження або вносити зміни в стан світу. У ньому пояснюються кроки, які необхідно виконати для отримання відтворюваних результатів.

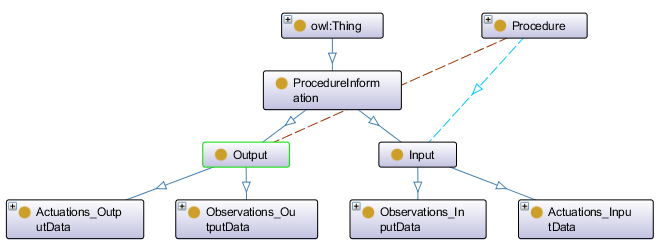


Рисунок 4.15 - Візуалізація класу ProcedureInformation та його відношень

* *shso:ProcedureInformation:* Клас вхідних та вихідних даних необхідних для роботи певних робочих процесів, протоколів, алгоритмів.

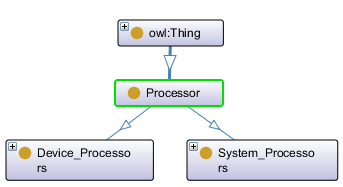


Рисунок 4.16 - Візуалізація класу Processor та його відношень

* *hssn:Processor:* Клас апаратного забезпечення, що відповідає за обробку даних в пристроях.

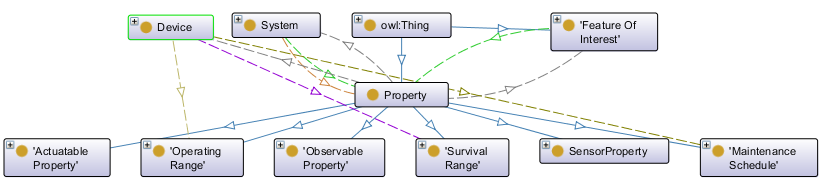


Рисунок 4.17 - Візуалізація класу Property та його відношень

* *ssn:Property:* Клас якостей сутності. Аспект сутності, який притаманний сутності і не може існувати без неї.

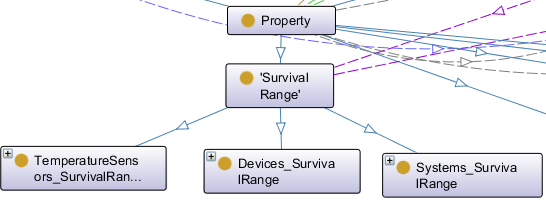


Рисунок 4.18 - Візуалізація субкласу Survival Range та його відношень

* *ssn:Survival Range:* Субклас класу *Property*, що представляє клас властивостей виживання системи при деяких заданих умовах, в яких система може виконувати покладені функції без пошкоджень.

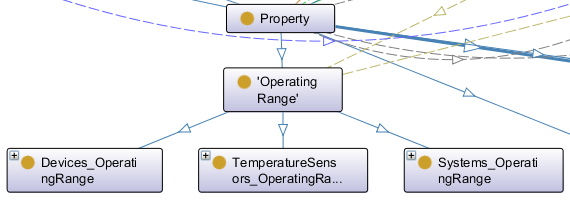


Рисунок 4.19 - Візуалізація субкласу Operating Range та його відношень

* *ssn:Operating Range:* Субклас класу *Property*, що представляє клас нормальних робочих властивостей систем при деяких заданих умовах, в яких очікується робота системи

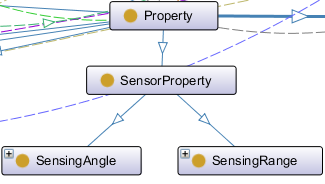


Рисунок 4.20 - Візуалізація субкласу SensorProperty та його відношень

* *shso:SensorProperty:* Субклас класу *Property*, що представляє клас властивостей датчиків для збору інформації з навколишньої середи

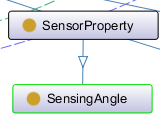


Рисунок 4.21 - Візуалізація субкласу SensingAngle та його відношень

* *hssn:SensingAngle:* Субклас класу Sensor*Property*, що представляє клас максимальних горизонтальних та вертикальних кутів вимірювання, що забезпечуються датчиком.

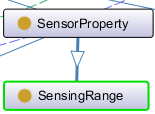


Рисунок 4.22 - Візуалізація субкласу SensingRange та його відношень

* *hssn:SensingRange:* Субклас класу Sensor*Property*, що представляє клас максимальних відстаней виявлення подразника, що забезпечуються датчиком.

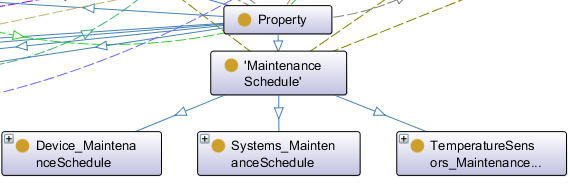


Рисунок 4.23 - Візуалізація субкласу Maintenance Schedule та його відношень

* *ssn:Maintenance Schedule:* Субклас класу *Property*, що представляє клас графіків технічного обслуговування кожного пристрою системи в заданих умовах.

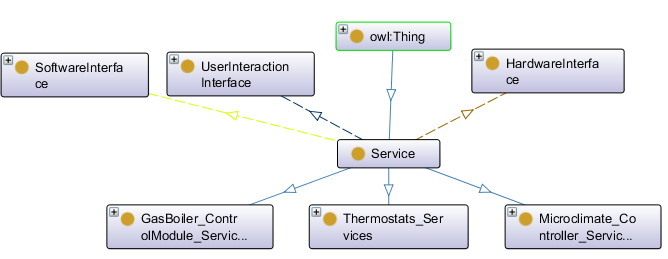


Рисунок 4.24 - Візуалізація класу Service та його відношень

* *hssn:Service:* Клас служб, що являють собою комплексні програмні об’єкти, здатні надавати деякі значення (або набір значень) щодо відповідного доменого об’єкту.

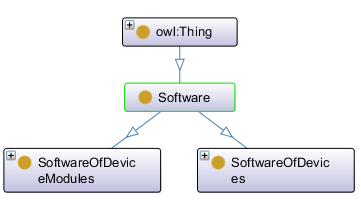


Рисунок 4.25 - Візуалізація класу Software та його відношень

* *hssn:Software:* Клас, що визначає усе програмне забезпечення (додатки і програми) запущені на пристрої.

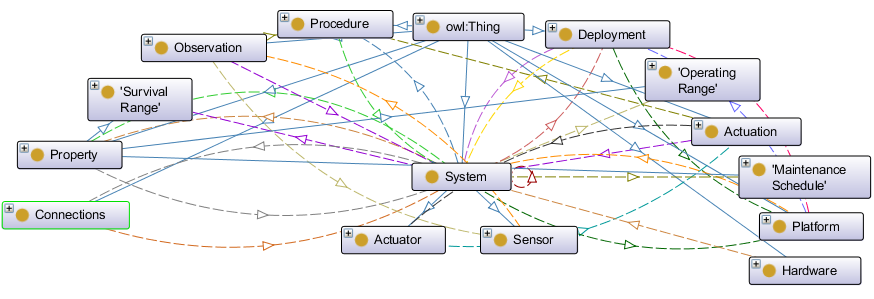


Рисунок 4.26 - Візуалізація класу System та його відношень

* *ssn:System:* Клас одиниць абстракції (систем) для частин інфраструктури, що реалізують певні процедури. Системи можуть мати компоненти їх підсистем, які є іншими системами.

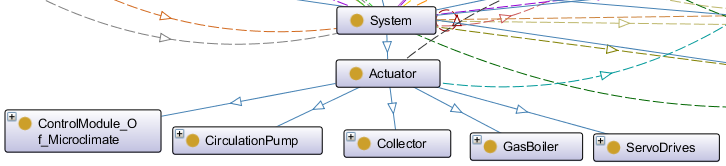


Рисунок 4.27 - Візуалізація субкласу Actuator та його відношень

* *sosa:Actuator:* Субклас класу *System*, що представляє клас пристроїв, які використовуються або реалізують процедуру (приведення в дію), яка змінює стан світу.

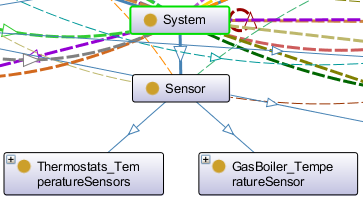


Рисунок 4.28 - Візуалізація субкласу Sensor та його відношень

* *sosa:Sensor:* Субклас класу *System*, що представляє клас пристроїв, що бере участь у процедурі дослідження або реалізує її. Датчики реагують на подразник (зміна навколишнього середовища; вхідні дані, складені з результатів попередніх спостережень) і генерують результат.

Описання основних властивостей об'єктів (object property – op) та властивостей даних (data property – dp) класів приведено нижче. Для більш точного описання використовуються позначення {Домен} → {Діапазон} для описання класу домену (тобто класу, що є первинним) та діапазону значень якому може належати властивість.

* *ssn:has property (op):* визначає відношення між об’єктом інтересу і його властивістю або між системою і її властивістю, тобто об’єкт інтересу або система мають цю властивість.

*{Feature Of Interest, System} → {Property}*

* *ssn:has operating range (op):* визначає відношення системи або девайсу до робочого діапазону, що описує нормальне робоче середовище системи, тобто система або девайс мають конкретний робочий діапазон.

*{Device, System} → {Operating Range}*

* *ssn:has survival range (op):* визначає відношення системи або девайсу до діапазону виживання, що описує виживання системи при деяких заданих умовах, в яких система може виконувати покладені функції без пошкоджень, тобто система або девайс мають конкретний діапазон виживання.

*{Device, System} → {Survival Range}*

* *hssn:hasHardware (op):* визначає співвідношення, що зіставляє системи або девайси з їх апаратним забезпеченням, тобто функція визначає які апаратні забезпечення мають системи або девайси.

*{Device, System} → {Hardware}*

* *hssn:Software (op):* визначає співвідношення, що зіставляє системи або девайси з їх програмним забезпеченням, тобто функція визначає які програмні забезпечення мають системи або девайси.

*{Device, System} → {Software}*

* *hssn:hasInterfaces (op):* визначає відношення, що зіставляє службу з її інтерфейсами, тобто кожна служба має свої інтерфейси для виконання роботи.

*{Service} → {Interface}*

* *shso:hasHardwareInterfaces (op):* визначає відношення, що зіставляє службу з її інтерфейсами апаратного забезпечення, тобто кожна служба має свої апаратні інтерфейси для підключення до джерела живлення та інших пристроїв.

*{Service} → {HardwareInterface}*

* *shso:hasSoftwareInterfaces (op):* визначає відношення, що зіставляє службу з її інтерфейсами програмного забезпечення, тобто кожна служба має свої програмні інтерфейси для комфортної роботи користувача з приладом.

*{Service} → {SoftwareInterface}*

* *shso:hasUserInteractionInterfaces (op):* визначає відношення, що зіставляє службу з її інтерактивними інтерфейсами, тобто кожна служба має свої інтерактивні інтерфейси для здійснення взаємодії користувача з приладом.

*{Service} → {UserInteractionInterface}*

* *shso:hasMaintenanceSchedule (op):* визначає відношення девайсу або системи, або об’єкту інтересу системи опалення до їх графіка технічного обслуговування, тобто визначає який прилад або об’єкт в системі має свій графік технічного обслуговування.

*{Device, System, HeatingSystem} → {Maintenance Schedule}*

* *ssn:is property of* *(op):* визначає відношення між властивістю і об'єктом системи опалення, якому воно належить, тобто функція визначає якому об’єкту системи опалення належить конкретна властивість.

*{Property} → {Feature Of Interest, Device, System}*

* *sosa:is observed by (op):* визначає відношення між властивістю спостереження і датчиком, здатним її спостерігати, тобто функція визначає яким датчиком спостерігається конкретна властивість спостереження.

*{Observable Property} → {Sensor}*

* *sosa:is hosted by (op):* визначає відношення між датчиком, приводом, девайсом і платформою, на якій він встановлений або розміщений, тобто функція визначає на якій платформі встановлені пристрої.

*{Device, System} → {Platform}*

* *sosa:hosts (op):* визначає відношення між платформою і датчиком, приводом, або платформою, розміщеної або встановленої на ній, тобто функція визначає пристрої встановлені на платформі.

*{Platform} → {System, Device}*

* *hssn:provides (op):* зіставляє девайс зі службою яку він забезпечує.

*{Device} → {Service}*

* *sosa:made actuation (op):* визначає відношення між виконуючим механізмом і приведенням в дію, яке він виконав.

*{Actuator} → {Actuation}*

* *sosa:made by actuator (op):* визначає відношення між приведенням в дію, яке виконуючий механізм виконав і ним самим.

*{Actuation} → {Actuator}*

* *sosa:made observation (op):* визначає відношення між спостереженням і датчиком, що виконує його

*{Sensor} → {Observation}*

* *sosa:made by sensor (op):* визначає відношення між датчиком, що виконує спостереження і ним самим

*{Observation} → {Sensor}*

* *shso:hasConnection (op):* визначає відношення девайсів або систем до їх з’єднання, тобто функція визначає які фізичні з’єднання має той чи інший девайс або система з іншими приладами.

*{Service, Device} → {Connections}*

* *shso:isConnectedTo (op):* визначає відношення між з’єднанням приладу та приладом до якого підключається з’єднання, тобто функція визначає які девайси або пристрої з’єднанні один з одним.

*{Connections} → {System, Device}*

* *shso:define\_Maintenance\_Time\_For\_Systems (dp):* визначає дані для значення наступного технічного обслуговування пристрою системи опалення у форматі YYYY-*MM-DDTHH:MM:SS*, де символ ‘T’ відокремлює дату та час.

*{Maintenance Schedule} → {xsd:dateTime}*

* *shso:define\_Previous\_Maintenance\_Time\_For\_Systems (dp):* визначає дані для значення попереднього технічного обслуговування пристрою системи опалення у форматі YYYY-*MM-DDTHH:MM:SS*, де символ ‘T’ відокремлює дату та час.

*{Maintenance Schedule} → {xsd:dateTime}*

* *shso:define\_Max\_OperatingTemperature (dp):* визначає дані для значення максимальної робочої температури пристрою системи опалення у форматі цілого числа.

*{Operating Range} → {xsd:int}*

* *shso:define\_Min\_OperatingTemperature (dp):* визначає дані для значення мінімальної робочої температури пристрою системи опалення у форматі цілого числа.

*{Operating Range} → {xsd:int}*

* *shso:define\_Max\_StorageTemperature (dp):* визначає дані для значення максимальної температури зберігання для пристрою системи опалення у форматі цілого числа.

*{Survival Range} → {xsd:int}*

* *shso:define\_Min\_StorageTemperature (dp):* визначає дані для значення мінімальної температури зберігання для пристрою системи опалення у форматі цілого числа.

*{Survival Range} → {xsd:int}*

* *shso:define\_SystemPowerSupply (dp):* визначає дані для значення електропостачання пристрою системи опалення у форматі рядка тексту.

*{Operating Range} → {xsd:string}*

* *shso:define\_SystemPowerSupply\_Type (dp):* визначає дані для типу джерела живлення пристрою системи опалення у форматі рядка тексту.

*{Operating Range} → {xsd:string}*

* *shso:NumberOf\_InstalledSystems (dp):* визначає дані для значення кількості встановлених систем у форматі цілого числа.

*{System} → {xsd:int}*

* *shso:ConnectionType\_ToDevice (dp):* визначає дані для типу з’єднання до девайсів системи опалення у форматі рядка тексту.

*{Connections} → {xsd:string}*

* *shso:ConnectionType\_ToSystem (dp):* визначає дані для типу з’єднання до систем системи опалення у форматі рядка тексту.

*{Connections} → {xsd:string}*

## Онтологія другого рівня на основі прикладної логіки оцінки ризиків

Описання онтології оцінки ризиків (RAO - Risk Assessment Ontology) на основі використання методології Coras починається з використання опису прикладної логіки предметної області як бази для описання ієрархії класів та вузлів їх відношення один з одним.

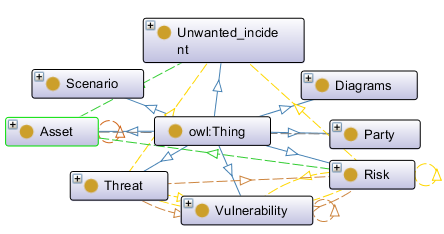


Рисунок 4.29 - Візуалізація ієрархії спроектованної онтології оцінки ризиків

Далі йде описання елементів представленної ієрархії.

Описання класів онтології оцінки ризиків:

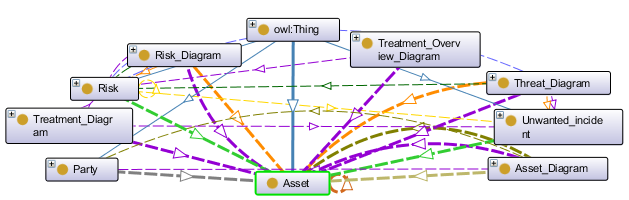


Рисунок 4.30 - Візуалізація класу Asset та його відношень

* *rao:Asset:* Клас, що представляє одиниці майна деякої системи, тобто стан або цінність тих чи інших елементів описуваної системи, які потребують захисту від загроз.

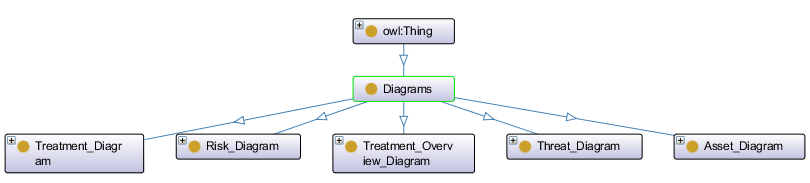


Рисунок 4.31 - Візуалізація класу Diagrams та його відношень

* *rao:Diagrams:* Клас, що репрезентує різні діаграми для оцінки ризиків (діаграми ризиків, діаграми загроз, діаграми активів, діаграми рішень).

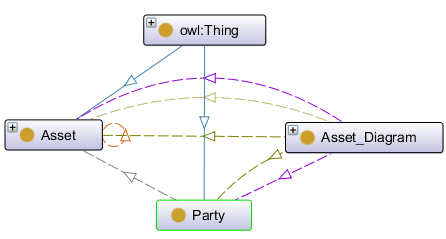


Рисунок 4.32 - Візуалізація класу Party та його відношень

* *rao:Party:* Клас, що представляє людей або організації, які можуть постраждати або вплинути на оцінку ризиків об’єкта

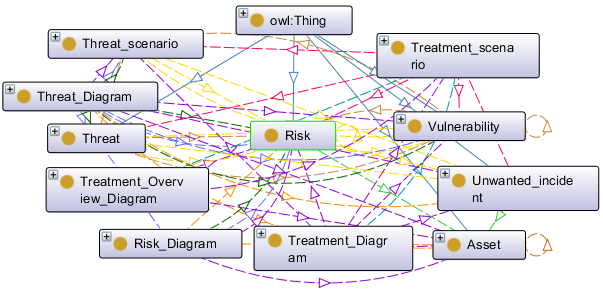


Рисунок 4.33 - Візуалізація класу Risk та його відношень

* *rao:Risk:* Клас, що представляє ймовірності виникнення конкретних подій, які мають різний вплив на активи досліджуваної системи

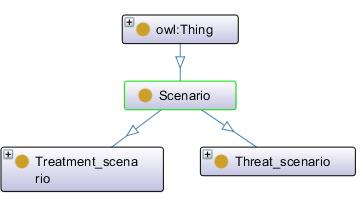


Рисунок 4.34 - Візуалізація класу Scenario та його відношень

* *rao:Scenario:* Клас елементу діаграми, що описує події які мають виникнути в результаті використання вразливостей або спосіб їх рішення

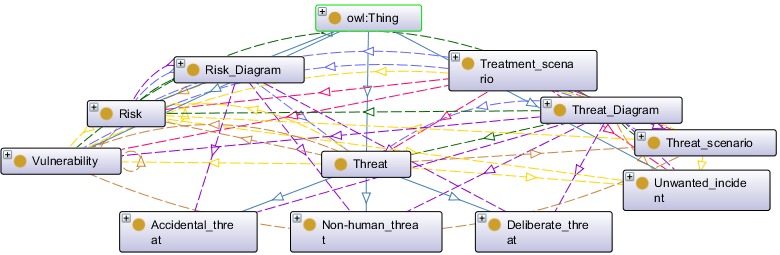


Рисунок 4.35 - Візуалізація класу Threat та його відношень

* *rao:Threat:* Клас, що описує загрози як можливі причини виникнення небажаних інцидентів в системі

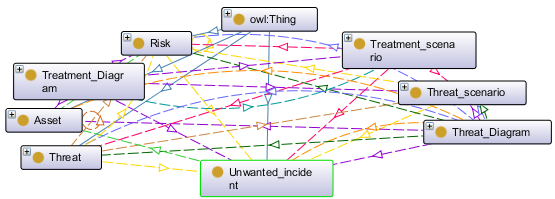


Рисунок 4.36 - Візуалізація класу Unwanted\_Incident та його відношень

* *rao:Unwanted\_Incident:* Клас, що описує небажані інциденти як події, що необхідно запобігти для того щоб не піддати загрозі безпеку активів системи

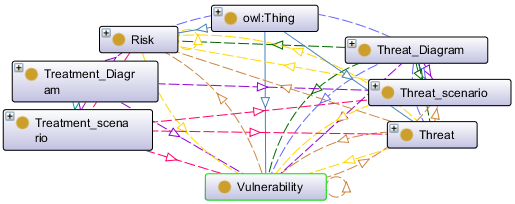


Рисунок 4.37 - Візуалізація класу Vulnerability та його відношень

* *rao:Vulnerability:* Клас, представляє вразливості системи, що можуть використати різного роду загрози.

Описання основних властивостей об'єктів класів онтології оцінки ризиків нижче виконано по такій же схемі як і описання онтології сенсорної мережі:

* *rao:impact (op):* визначає відношення між об’єктами онтології як відношення “впливає”, де ризик або небажаний інцидент впливають на елементи майна системи.

*{Risk, Unwanted\_incident} → {Asset}*

* *rao:indirect\_harm (op):* визначає відношення між об’єктами онтології як відношення “завдає непрямої шкоди”, де елементи майна завдають непряму шкоду іншим, тобто вплив на один елемент майна впливає і на інший пов’язаний з ним.

*{Asset} → {Asset}*

* *rao:initiate (op):* визначає відношення між об’єктами онтології як відношення “ініціює”, де загрози або вразливості ініціюють появу інших елементів діаграми по типу сценарієв загроз, ризиків та вразливостей.

*{Vulnerability, Threat} → {Vulnerability, Risk, Threat}*

* *rao:leads\_to (op):* визначає відношення між об’єктами онтології як відношення “призводить до”, де загрозаи, сценарії загроз та риски призводять до появи небажаних інцидентів, вразливостей або інших рисків

*{Threat, Risk, Threat\_scenario} → {Unwanted\_incident, Risk, Vulnerability}*

* *rao:protect (op):* визначає відношення між об’єктами онтології як відношення “захищає”, де організація або людина захищає важливий для них елемент майна системи

*{Party} → {Asset}*

* *rao:treatment (op):* визначає відношення між об’єктами онтології як відношення “лікує (вирішує)”, де сценарії рішення лагодить інші елементи діаграми: загрози, ризики, сценарії загроз, вразливості системи, небажані інциденти.

*{Threatment\_scenario} → {Unwanted\_incident, Risk, Vulnerability, Threat, Threat\_scenario}*

## Визначення оцінки ризиків сенсорної мережі системи опалення

Для визначення оцінки ризиків сенсорної мережі за основу була взята ситуація коли вихід за межи графіка технічного обслуговування одного пристрою або їх сукупності в мережі впливає на працездатність усієї системи через зношення апаратних компонентів цих пристроїв і створює відповідну загрозу збою працездатності обладнання сенсорної мережі. Відповідно для моделювання такого сценарію необхідно було визначити можливі пошкодження компонентів кожного пристрою мережі, що визначені у таблиці 1.1-1.2 (Повна таблиця визначена у додатку Б)

Таблиця 1.1 – Апаратні компоненти центру комутації та його модулю сенсорної мережі системи опалення схильні до зношення

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Назва пристроїв/Компоненти** | LED-індикатори стану пристрою | Провода | Порти підключення пристроїв до зон розподілу сигналів | Перемикачі функцій пристроїв | Елементи апаратного забезпечення, що згоріли в результаті сильного стрибка напруги | Порти живлення |
| Salus KL06-M (центр комутації) з класу **Device** | Х | Х | Х | Х | Х | Х |
| Salus PL07 (модуль центра комутації) з класу **Actuator** | Х | Х |  |  | Х | Х |

Таблиця 1.2 – Апаратні компоненти термостатів та терморегуляторів сенсорної мережі системи опалення схильні до зношення

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Назва пристроїв/Компоненти** | Провода | РК-дісплей | Перемикачі функцій пристроїв | Порти живлення | Порти з’єднання з центром комутації | Кнопки керування пристроєм |
| Salus ERT20 (термостат) з класу **Device** | Х |  | Х | Х | Х |  |
| Salus ERT30 (термостат) з класу **Device** | Х | Х | Х | Х | Х | Х |
| Salus ERT50 (термостат) з класу **Device** | Х | Х |  | Х | Х | Х |
| Salus HTRS230 (термостат) з класу **Device** | Х | Х |  | Х | Х | Х |

Для основи розроблюваної моделі оцінки ризиків покладемо представлення сенсорної мережі у вигляді множини пристроїв:

, (4.1)

|  |  |
| --- | --- |
|  |  |

де *L* – загальна кількість пристроїв в сенсорній мережі.

Сукупність елементів пристрою, які потребують технічного обслуговування представляється у вигляді множини:

, (4.2)

де *I(l)* – загальна кількість елементів в *L* – ому пристрої, які потребують технічного обслуговування в розглянутій сенсорній мережі. Представимо сукупність спостережуваних параметрів деякого елемента у вигляді:

, (4.3)

де – загальна кількість спостережуваних параметрів елемента .

З множин , експертним шляхом, виділяється множина параметрів , значення яких в подальшому необхідно враховувати в процесі спостереження за технічним станом деякого пристрою . Сукупність параметрів будемо позначати у вигляді:

, (4.4)

де – загальна кількість параметрів, які необхідно враховувати у процесі ідентифікації технічного стану деякого елемента .

В результаті можно зробити висновок, що таким чином збій працездатності обладнання мережі є складовою загрозою, що визначається загрозами збою працездатності всіх пристроїв сенсорної мережі, перелік яких визначається відповідно до формули (4.1).

Тобто складову загрозу збою працездатності можно описати за допомогою таких речень прикладної логіки:

* Типи об’єктів діаграми = Збій в працездатності обладнання → {Мережа}
* Типи об'єктів = Мережа → {Мережевий адаптер, Пристрої, Навколишнє середовище, Провідне середовище}.

Відповідно до таблиць компонентів пристроїв мережі (див. Таблиці 1.1-1.2) та розробленої моделі оцінки ризиків, можна побудувати діаграму загроз сенсорної мережі за допомогою ПЗ методології Coras, де основною загрозою відповідно до моделі оцінки ризиків являється збій в працездатності обладнання.



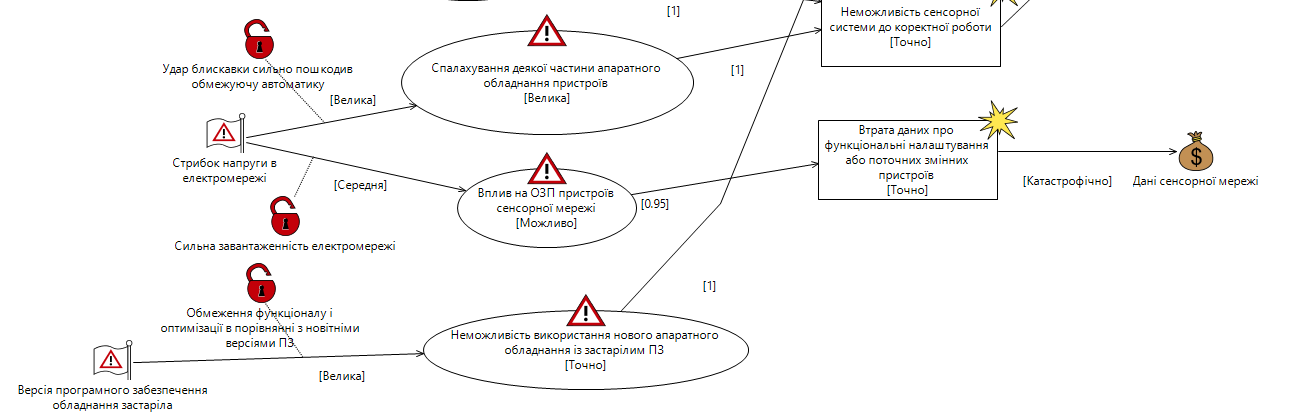


Рисунок 4.38 – Діаграма загроз сенсорної мережі системи опалення

Для детального описання оцінки ризиків необхідно ввести декілька термінів, що пов’язують компоненти мережі з елементами діаграми загроз відповідно до методу з статті [6]:

*Thi* – термін, що позначає елементи загроз, де символ *i* – порядковий номер загрози;

*ThSi,j*– термін, що позначає елементи сценарієв загроз, де символ *i* – порядковкий номер загрози, якій належить сценарій а *j* – порядковий номер сценарію загрози;

*Vbi,j* – вразливість яку використала загроза, де *i* – порядковий номер загрози, яка використовує вразливість системи, а *j* – порядковий номер вразливості по відношенню до номеру загрози;

*UIi* – небажаний інцидент, де *i –* порядковий номер інциденту;

*Asi* – актив, де *i* – порядковий номер активу;

Опис елементів діаграми загроз:

*Th1* = Збій в працездатності обладнання (*Мережа →* *{Мережевий адаптер, Пристрої, Навколишнє середовище, Провідне середовище}*);

*Vb1,1* = Перевищення/зниження температури обладнання понад встановленої робочої межі (*Пристрої →* *{Device, System}*);

*Vb1,2*= Вихід за доступні межі графіка технічного обслуговування (*Мережа);*

*Vb1,3* =Перевищення/зниження температури навколишнього середовища обладнання понад встановлені межі виживання ((*Пристрої*  *Навколишнє середовище) →* (*{Device, System}*  *{Feature of Interest}*);

*ThS1,1* = Виконання процедур датчиками температури погіршується;

*ThS1,2* = Некоректний обмін даними температури між пристроями;

*UI1* =Неправильні команди подачі теплоносія в вже опалювальні приміщення (Вплив на актив – суттєвий)

*As1* = Температурна середа дому - актив, що представляє з себе температурну середу в якій знаходиться уся сенсорна мережа і кардинальна зміна якої може спричинити жахливі наслідки для всієї системи в цілому.

*ThS1,3* =Скорочення експлуатаційного ресурсу пристроїв;

*ThS1,4* = Невідповідність між значеннями налаштованої функціональність пристроїв;

*ThS1,5* =Некоректність передачі сигналів;

*UI2* = Невиконання покладених на пристрої функціональних обов’язків (Вплив на актив – суттєвий);

*As2* = Продуктивність обслуговування пристроїв - актив, що є показником того наскільки якісно служби пристроїв надають власні послуги.

*ThS1,6*= З кожним днем зростає інтенсивність зношення апаратної частини обладнання;

*ThS1,7*= Виникнення системних збоїв;

*UI3* = Повне відключення всієї системи (Вплив на актив – катастрофічний);

*As3* = Працездатність сенсорної мережі - актив, що є серцем і показником надійності сенсорної мережі, зміна якого також впливає на всі пристрої системи;

*ThS1,8* = Пристрій не був вимкнен автоматично тому його продуктивність роботи знижена;

*UI4* = Зниження продуктивності системи (Вплив на актив – терпимий);

*ThS1,9* = Автоматичне відключення пристрою;

*UI5* = Неможливість сенсорної системи до коректної роботи (Вплив на актив – катастрофічний);

*Th2* = Стрибок напруги в електромережі (*Мережа*);

*Vb2,1* = Удар блискавки пошкодив обмежуючу автоматику (*Пристрої →* *{Device, System}*);

*Vb2,2* = Сильна завантаженість електромережі (*Провідне середовище*);

*ThS2,1* = Спалахування частини апаратного обладнання пристроїв;

*ThS2,2* = Вплив на ОЗП пристроїв сенсорної мережі;

*UI6*= Втрата даних про функціональні налаштування або поточні змінні пристроїв (Вплив на актив – катастрофічний)

*As4* = Дані сенсорної мережі - актив, що представляє з себе дані пристроїв мережі які передаються в ній і дані налаштувань пристроїв.

*Th3* = Версія програмного забезпечення обладнання застаріла (*Пристрої →* *{Software}*);

*Vb3,1* = Обмеження функціоналу і оптимізації в порівнянні з новітніми версіями ПЗ (*Пристрої →* *{Device, System}*);

*ThS3,1* = Неможливість використання нового апаратного обладнання із застарілим ПЗ

# ВИСНОВОК

В даній кваліфікаційній роботі бакалавра була вирішена задача проектування багаторівневої онтології для сенсорної мережі системи опалення з використанням методології Coras та мови прикладної логіки.

Багаторівнева онтологія сенсорної мережі розроблена в повному обсязі згідно вимогам, що висувалися на початку проекту, поставленому завданню на розробку. Спроектована багаторівнева онтологія забезпечує можливість обробки інформації з бази знань для оцінки ризиків у сенсорній мережі, детальне описання предметної області за допомогою прикладної логіки онтології рівнів з четвертого по перший, детальне описання класів та властивостей відношень онтології найнижчого рівня системи опалення на фізичному рівні взаємодії: стаціонарні пристрої, датчики температури повітря та температури води, виконавчі механізми зміни стану елементів системи, описані відповідні процедури для чіткої взаємодії всіх структур системи.

Для першого розділу проекту було проведено необхідний літературний огляд та аналіз вже існуючих подібних сенсорних мереж та схожих онтологій як основи для проекту. Були з’ясовані первісні методи описання онтологій за допомогою мови прикладної логіки, механізми візуалізації ієрархії онтологій та створення діаграми загроз на основі розробленої моделі оцінки ризиків, вибір необхідного реального обладнання автоматичних систем опалення для відтворення дійсності.

Було проведено тестування спроектованої багаторівневої онтології за допомогою проведення оцінки ризиків та створення таблиці компонентів пристроїв мережі на які можуть впливати загрози.

# Перелік літературних джерел

1. Майкл К. С., Кріс В., Дебора Л. М. “Посібник з мови веб-онтології OWL”
2. Клещев А.С., Артемьева И.Л. “Незбагачені системи логічних співвідношень”
3. Баранова О.К., Бабаш О.В. “Інформаційна безпека та захист інформації” / Баранова О.К. – Москва: РІОР, 2022. C. 336
4. Еліо М., Річард Ш., Філіп А. “HSSN: Онтологія для гібридних семантичних сенсорних мереж”
5. Омер Б.С., Ердоган Д., Сердар З. К. “Розробка онтології "розумного дому" та реалізація симулятора семантичної сенсорної мережі: підхід до Інтернету речей” / Омер Б.С.: опубліковано на міжнародній конференції з технологій та систем спільної роботи (CTS), 2015. C. 5-6
6. М.Б. Копитчук, П.М. Тішин, М.В. Цюрупа “Аналіз обчислювальних мереж з допомогою багаторівневої онтології оцінки ризиків з використанням методології Coras” / Копитчук М.Б.: опубліковано в журналі Електротехнічні і комп’ютерні системи № 10 (86), 2013. C. 120-126