

*Оліяр О.**Науковий керівник – доц. Франко Ю.П.*

## СТВОРЕННЯ СИСТЕМИ ВІЗУАЛІЗАЦІЇ ТА МОДЕЛЮВАННЯ БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖ

З розвитком науки та технологій відкрились нові перспективи у сфері збирання, передавання та обробки інформації. В даний час одним з нових актуальних напрямків в області інформаційних технологій є створення нового виду мережевих систем - сенсорних мереж (СМ, англ. - WSN). Безпроводна сенсорна мережа (WSN – Wireless Sensor Network) - це розподілена, самоорганізована, стійка до відмов елементів система, що складається з багатьох вузлів, з'єднаних між собою радіоканалом. Безпроводникова сенсорна мережа може розміщуватись на практично необмеженій території і за сучасними оцінками може містити до 65000 вузлів.

Безпроводні сенсорні мережі (Wireless Sensor Network) складаються з маленьких вузлів (сенсорів), з інтегрованими функціями моніторингу навколишнього середовища, обробки і передачі даних.

**Актуальність дослідження.** Задача візуалізації графових структур БСМ є надзвичайно актуальною та потребує з кожним роком більш якісного розв'язку. Адже із розповсюдженням Web-технологій, розвитком нових напрямків у науці та впровадженням новітніх підходів до організації освітнього і управлінського процесів, все частіше і частіше виникає потреба у графічному представленні графових структур даних різного походження. Особливо актуально ця задача стає при потребі візуалізації структур значних розмірів – зокрема при вирішенні різномірних проблем наукового, корпоративного та навіть державного характеру. Актуальність даної теми додатково підсилюється тим, що вона не є вузько спеціалізованою, тобто не орієнтована для застосування лише у певній галузі або сфері, а навпаки є легко застосовною в багатьох галузях людської діяльності. Програмне забезпечення для візуалізації графових структур БСМ створюється з метою вирішення цих проблем.

В загальному проектування і реалізація сенсорних мереж потребують вирішення безлічі складних проблем, що відносяться до різних областей досліджень.[5]

**Аналіз останніх досліджень та публікацій.** Дослідження і розробки в області СМ інтенсивно ведуться за межами нашої держави, причому результати останніх вже використовуються в конкретних програмах. Найбільш успішними дослідженнями в цій галузі були дослідження професора університету штату Каліфорнія - Крістофера Пістера.

Тематика робіт простягається від вузькоспеціальних питань, пов'язаних зі створенням окремих компонентів об'єктів мережі (мікроконтролерів, датчиків і т.д.) з низькою ціною і низьким енергоспоживанням до проблем, які виникають при експлуатації сенсорних мереж (питання пов'язані з організацією роботи мережі, розробка програмного забезпечення, прив'язка місця розташування об'єктів мережі до географічних координат та ін.) [7].

Результатом робіт по стандартизації БСМ стало сімейство стандартів IEEE 802.15.4.

БСМ як системи моніторингу параметрів об'єктів на базі дискретних бездротових сенсорних мереж описані в роботах М. Н. Терентьєва. Вирішенням завдань, пов'язаних з оцінкою, аналізом і ефективним управлінням інформаційними потоками в БСМ, займалася І. А. Іванова.

Дослідженню енергоспоживання та створенню програмних і апаратних засобів БСМ велику увагу приділяє американська компанія Texas Instruments (радіомодулі CC2530 та ін.). Також виробництвом радіомодулів займаються компанії Atmel, Digi International, Ember, Freescale, Samsung, STMicroelectronics, Microchip Zigbee [3].

**Мета статті** – вивчення та обґрунтування функціональних можливостей комп'ютерних технологій для моделювання та візуалізації сенсорних мереж.

Для досягнення мети необхідно розробити узагальнений алгоритм реалізації модифікованого пружинного методу, який, порівняно з оригінальним методом, дозволяє отримувати простіші для сприйняття подання графів за рахунок зменшення кількості перетинів ребер шляхом уникнення локальних мінімумів енергії фізичної системи.

**Виклад основного матеріалу.** Візуалізація складних концептуальних структур є ключовою компонентою в багатьох додатках в науці і техніці. Граф – це абстрактна структура, яка

використовується для моделювання інформації. Графи застосовуються для представлення будь-якої інформації, яку можна змоделювати у вигляді об'єктів і зв'язків між ними. Системи візуалізації інформації в багатьох предметних областях потребують таких алгоритмів візуалізації графів, які дозволяють спростити і полегшити сприйняття і аналіз інформації експертами і користувачами. Генерація зображення графів має багато областей застосування [6].

На даний час представлення інформації все більше і більше використовується в різних областях точних та природних наук. В програмуванні графові моделі застосовуються при створенні програмного забезпечення (управляючі графи, ієрархії класів, діаграми потоків даних), дизайні бази даних (діаграми сутність-зв'язок), розробці інформаційних систем і систем реального часу (моделювання комп'ютерних мереж, граfi станів, мережі Петрі), а також в системному програмуванні – при створенні теорії компіляції і перетворювання програм. Питання візуалізації подібних графових структур являється наріжним каменем в процесі адекватного відображення інформації [1].

До об'єкту дослідження належать процеси функціонування компонентів БСМ в умовах дії збурювальних факторів. Графи в подальшому будуть адаптовані для представлення обмеженої поточної інформації в мережі, в якій дуги між вузлами відповідають спільно використовуваній інформації. Надалі вводиться низка основних понять і позначень щодо моделювання на основі графів та висвітлюються деякі корисні властивості, пов'язані з матричним поданням графів. Топологію БСМ можна описати орієнтованим графом  $G(V,E,w)$ , який визначається набором вузлів, і дуг, пов'язаних з набором вузлів декартовим добутком, тобто, причому  $w$  – ваговий коефіцієнт (рис. 1). Кількість вузлів графа  $G$  – це його порядок, а загальна кількість дуг – розмір. Нехай набір вузлів і дуг графа  $G$  іменуються  $V(G)$  та  $E(G)$ , відповідно. Якщо застосовується символ для позначення кардинального числа (кількості елементів множини), то отримується, що порядок графа – це  $V(G)$  та його розмір  $E(G)$  – це кількість його дуг [4].

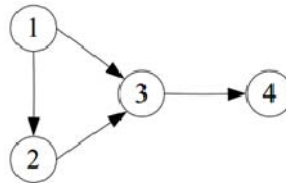


Рис. 1. Орієнтований граф для  $V=\{1, 2, 3, 4\}$  та набору дуг  $E=\{(1, 2), (1,3), (2, 3), (3, 4)\}$

Застосування об'єктно-орієнтованого підходу для реалізації цієї задачі містить етапи, наведені на рис. 2.



Рис. 2. Етапи проектування системи візуалізації

Перший етап є базовим і полягає в аналізі вимог та попередньому проектуванні системи. На початку розроблення системи після аналізу вимог необхідно розбити її на модулі. Під модулями прийнято розуміти набір класів та окремих об'єктів, підсистем та подій, взаємозв'язаних між собою.

Зв'язок (інтерфейс) між підсистемами забезпечує форму всіх взаємодій між ними, проте не специфікує внутрішню структуру чи особливості реалізації підсистеми. З огляду на це кожна підсистема може розроблятися незалежно від решти підсистем.

Підсистеми визначають послідовний спосіб розгляду прикладної задачі, для вирішення якої розробляється певна система. Наприклад, система файлів є підсистемою операційної системи, яка забезпечує набір взаємозв'язаних абстрактних операцій, що реалізуються у вигляді окремого модуля.

Підсистеми повинні визначатися так, щоб більша частина їх взаємодій залишалась всередині підсистеми і тим самим зменшувала глобальні потоки даних та залежності між ними. Існує два типи взаємодій між підсистемами: клієнт-сервер та міжпрограмна взаємодія. У першому випадку клієнт дає запит серверу на виконання певної операції, а сервер повертає результат. У випадку міжпрограмної взаємодії обидві підсистеми викликають одна одну та обмінюються даними у міру необхідності. Для реалізації прикладної системи відображення даних оптимальною є реалізація на міжпрограмному рівні, оскільки дає змогу простіше та швидше (минаючи канали зв'язку) реалізувати процедури створення графічних примітивів.

Особливістю цього етапу є синтез моделей, які дають змогу спроектувати та протестувати розроблену систему залежно від вимог, які до неї висуваються. Застосування моделей є вкрай важливим, оскільки вони дають змогу спростити розроблення, аналіз і реалізацію системи на комп'ютері внаслідок акцентування на основних особливостях розроблюваного продукту. Побудова моделей є поширеним способом вивчення складних об'єктів і явищ для перевірки працездатності розроблюваної системи та її модифікації як на початку проектування, так і на інших етапах розроблення.

Систему візуалізації графа, представимо у вигляді двох взаємозалежних моделей: функціональної та об'єктної. Вони дають змогу адекватно представити структуру проектованої системи та визначити всі функціональні залежності між об'єктами загалом.

Об'єктна модель представляє проектований додаток у вигляді сукупності об'єктів, кожен з яких відповідає певним особливостям та виконує специфічні функції. В ній зображуються об'єкти, важливі для розроблюваного додатка, які визначають прагматику досліджуваної системи. Під об'єктами розуміють поняття, абстракції або будь-які елементи з чітко окресленими границями, які мають значення у контексті розглядуваної задачі. Окреслення об'єктів переслідує дві мети: визначення розглядуваної задачі та основ для її реалізації на комп'ютері. Отже, ціллю розробки об'єктної моделі є опис елементів, що становлять проектовану систему та встановлення залежностей між ними. Об'єктну модель програмного продукту візуалізації графів наведено на рис. 3.

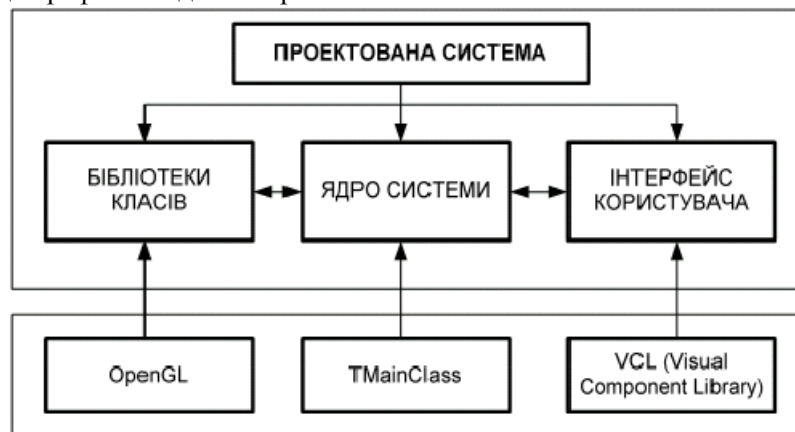


Рис. 3. Об'єктна модель системи відображення графа

Спроектвана модель складається з трьох основних частин: графічного інтерфейсу користувача, ядра системи та бібліотеки класів, що реалізує основні процедури обробки зображень.

Графічний інтерфейс користувача об'єднує в собі всі елементи та компоненти програми, які впливають на взаємодію користувача з прикладною програмною системою та виконує функції:

– навігації між блоками системи, відображення інформації про поточний стан системи, зворотний

– зв'язок з користувачем, підтримку прийняття рішень при використанні системи. Особливою 8

– функцією графічного інтерфейсу є організація процесу введення вхідної інформації, її фільтрування та збереження на зовнішніх носіях. Графічний інтерфейс програми створюється із використанням

– бібліотеки візуальних компонентів (Visual Component Library, VCL). Це є об'єктно-орієнтована бібліотека, яка застосовується при створенні програм за допомогою засобів візуального програмування. VCL містить велику кількість готових елементів керування, придатних для застосування в різних сферах, таких як робота з базами даних, взаємодія з операційною системою, програмування мережевих додатків.

Ядро системи забезпечує функціонування додатка та інтегрує в собі такі процеси, як моделювання, візуалізація, підтримка нових типів об'єктів та алгоритмів. Ядро функціонально дає змогу маніпулювати окремими об'єктами та створювати конвеєри моделювання та візуалізації з окремих алгоритмів. Застосування такого підходу дає змогу не лише створювати результуюче зображення, але й зберігати історію його створення. Володіючи даними історії створення, ядро системи може в автоматичному режимі модифікувати об'єкти зображення у випадку зміни початкових даних.

Важливою операцією, яку виконує ядро системи, є опрацювання граничних ситуацій, яке здійснюється шляхом визначення реакції кожного об'єкта й всієї системи на певні зовнішні дії.

Виділяють такі граничні ситуації: ініціалізація, термінація й обвал. Зокрема, процес ініціалізації полягає в приведенні системи у фіксований початковий стан: оголошення всіх початкових змінних та параметрів задачі та формування ієрархії класів. Це є початкова фаза роботи системи, під час якої доступною є лише частина можливостей. Процес термінації полягає у вивільненні всіх апаратних та програмних ресурсів, зайнятих під задачі системи. Забезпечення цього процесу є вкрай важливим, оскільки за його допомогою забезпечується безпомилкова робота апаратного забезпечення при завершенні роботи прикладної програми. Процес обвалу являє собою незаплановану термінацію системи. Ця ситуація може виникнути у результаті помилок користувача (при введенні некоректних вхідних даних), недостатньої кількості ресурсів системи (браку місця на зовнішніх носіях інформації) чи сторонніх впливів (вимкнення електроенергії). З огляду на зазначені чинники у ядрі системи передбачені відповідні алгоритми та методи перехоплення, що мінімізують вплив зазначених факторів на роботу програми.

Бібліотека класів – реалізує основні процедури обробки зображень. Дані та алгоритми бібліотеки класів виконані у вигляді ієрархії. Їх застосування дає змогу реалізувати основні методи опрацювання сформованих зображень відповідно до потреб користувача. Класи бібліотеки, крім основних геометричних перетворень, підтримують додаткові (об'єднання, накладання, позиціонування). Як стандарт під час розроблення бібліотеки класів застосовано бібліотеку Open Graphics Library (OpenGL) – відкриту графічну бібліотеку, яка визначає незалежний від мови програмування кросплатформений програмний інтерфейс для написання додатків, що використовують комп'ютерну графіку. Ця бібліотека містить близько 250 функцій для відображення різних графічних схем із застосуванням примітивів. OpenGL орієнтується переважно на уніфікацію реалізації програмного коду незалежно від апаратної платформи. На базовому рівні ця бібліотека описує поведінку функцій, що її формують. Реалізація стандарту дає змогу використовувати наявні функції апаратного забезпечення, а у випадку неможливості їх забезпечення – емулювати їх програмно. Основним принципом роботи бібліотеки є отримання набору векторних графічних примітивів з подальшим математичним їх опрацюванням та побудовою растрового зображення на екрані монітора.

Запропонована об'єктна модель реалізована у вигляді статичної структури проекрованої системи. Однак її знання є недостатнім для розуміння й оцінки роботи, оскільки необхідно мати засоби для опису змін, що відбуваються з об'єктами та їх зв'язками під час роботи кожної підсистеми. З огляду на це варто побудувати функціональну модель. Для її побудови необхідно визначити вхідні і вихідні значення параметрів об'єктів моделі та побудувати схему функціональних залежностей між її елементами.

Функціональна модель складається з окремих об'єктів та взаємозв'язків між ними (рис. 4).

У спроектованій моделі стан об'єктів характеризується сукупністю поточних значень атрибутів та зв'язків. Під час роботи вони взаємодіють між собою, у результаті чого змінюються їх стани.



Рис. 4. Функціональна модель побудови зображення

Одиницею впливу є події, кожна з яких змінює стан одного або декількох об'єктів у системі або спричиняє виникнення нових подій. Робота системи характеризується послідовністю подій, що в ній відбуваються. Вони можуть бути незалежними (введення матриці суміжності) або передувати іншим (візуалізація графу здійснюється після введення матриці) і передають інформацію з одного об'єкта в інший. Особливістю створеної функціональної моделі є наявність певних взаємозв'язків, що відтворюють процес візуалізації і наводять процедуру створення об'єкта візуалізації у визначені моменти часу. З огляду на зазначені особливості, представлені моделі дають змогу одержати повне уявлення про функціональність системи, що значно спрощує подальше її програмування, налагодження, супровід та модифікацію.

Важливим етапом проектування прикладної програми візуалізації даних є реалізація принципів керування програмним забезпеченням. Як видно із функціональної моделі, під час аналізу всі взаємодії в системі подаються у вигляді подій. Існує два класи методів керування ними: зовнішні та внутрішні. При зовнішньому керуванні в кожний момент часу діє одна із процедур (це є найпростіший принцип). Внутрішнє керування пов'язане з потоками управління в процесах. Створена система візуалізації даних вимагає проектування додатка з використанням принципів зовнішнього керування із застосуванням методики інтерактивного інтерфейсу. Основною особливістю цього типу систем є потреба у виділенні об'єктів, що формують інтерфейс (забезпечення їх реалізації за допомогою VCL) з використанням засобів середовищ об'єктно-орієнтованого програмування (механізмів наслідування та поліморфізму), які дають змогу утворити цілісний каркас програмного продукту та відносно легко його модифікувати [2].

Для реалізації модуля візуалізації БСМ на основі графа спроектуємо основні класи: Graph, Vertex, Edge.

Основні класи UML-діаграм наведено на рис. 5: UML-діаграма класу Graph, який служить оболонкою для об'єкту – граф, наведено; UML-діаграма класу Edge, який служить оболонкою для об'єкту ребра графа, UML-діаграма класу Vertex, який служить оболонкою для об'єкту – вершина графа.

Для реалізації програмного забезпечення використаємо мову програмування C# на платформі .NET.C# – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції–члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML.

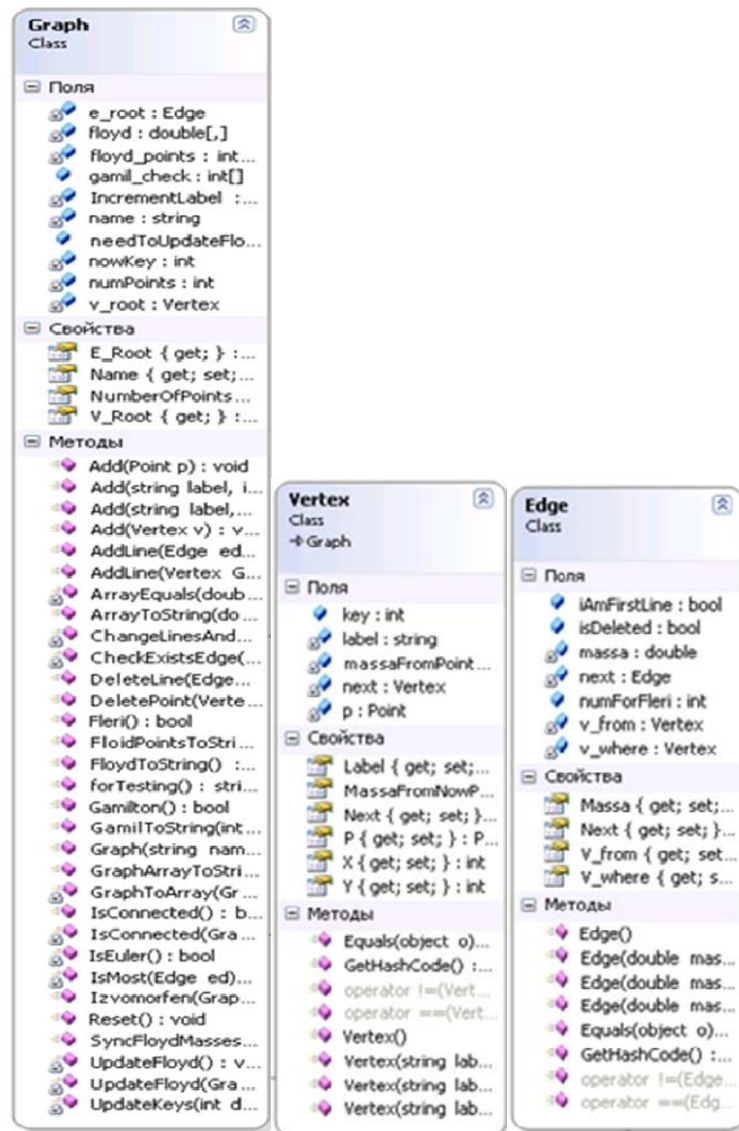


Рис. 5. UML-діаграми класів *Graph*, *Edge*, *Vertex*екту – вершина графа

.NET Framework – програмна платформа, розроблена компанією Microsoft в 2002 році. Основа платформи є віртуальна машина Common Language Runtime (CLR), здатна виконувати як звичайні програми, так і серверні веб-додатки. .NET Framework підтримує створення програм, написаних на різних мовах програмування. Програма для .NET Framework, написана на будь-якій мові програмування, спочатку перекладається компілятором в єдиний для .NET проміжний байт-код Common Intermediate Language (CIL). Потім код або виконується віртуальною машиною Common Language Runtime (CLR), або транслюється утилітою NGen.exe у виконуваний код для конкретного цільового процесора. Використання віртуальної машини краще, тому що позбавляє розробників від необхідності піклуватися про особливості апаратної частини. У разі використання віртуальної машини CLR, вбудований в неї JIT-компілятор (just in time) перетворює проміжний байт-код в машинні коди потрібного процесора. Сучасна технологія динамічної компіляції дозволяє досягти високого рівня швидкодії. Віртуальна машина CLR також сама піклується про базову безпеку, управління пам'яттю і систему винятків, позбавляючи розробника від частини роботи.

На рисунку 6 наведено головне вікно програми. Програма призначена для створення наочних ілюстрацій БСМ, завдяки простому управлінню стане зручним інструментом для дослідження.

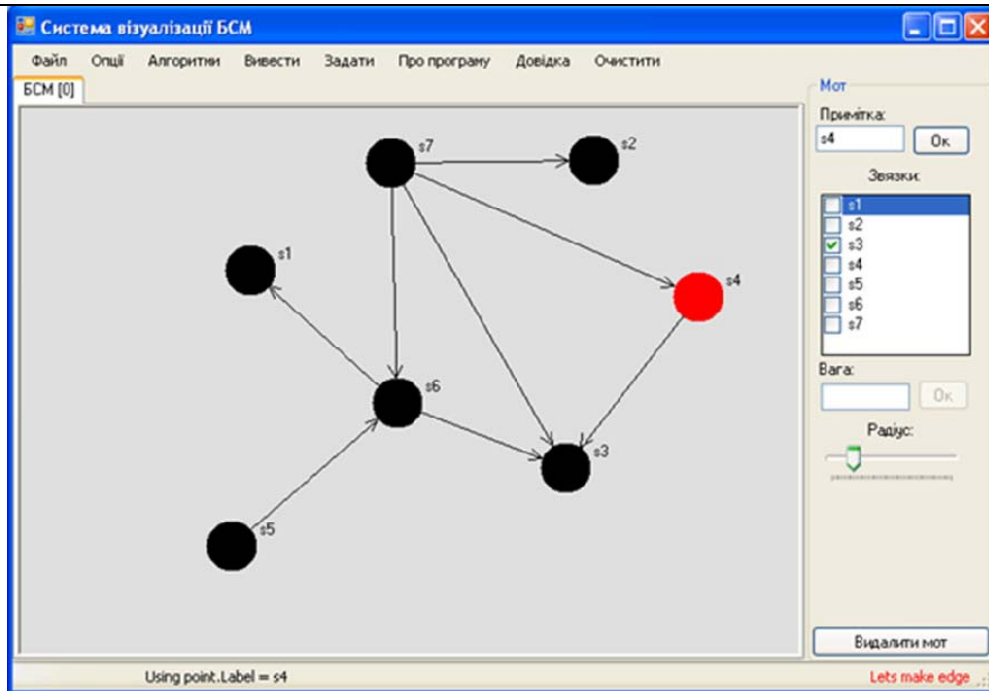


Рис. 6. Головне вікно програми

**Висновки.** Обґрунтовано використання методу візуалізації графів на основі фізичних аналогій, оскільки, фізичні аналогії, з одної сторони, роблять алгоритм візуалізації достатньо прозорим для розуміння і простим для реалізації, а з іншої, приводять до алгоритмів, які дають досить оптимальні розташування графів. Розроблено програму для візуалізації БСМ.

З розвитком, вдосконаленням і ускладненням технологій різних виробництв потреба в використанні бездротових сенсорних мереж буде тільки рости. Таким чином, бездротові технології стануть все більш і більш популярним через свою низьку вартість і простоту у використанні.

**Перспективи подальших досліджень.** Перспективність розвитку бездротових сенсорних мереж очевидна. Вже сьогодні у багатьох галузях починають використовувати WSN. З вдосконаленням технологій і ускладнення різних виробництв потреба в бездротових сенсорних мережах буде тільки рости. Таким чином, бездротові технології стають все більш і більш популярними через свою низьку вартість і простоту використання. Тому у багатьох випадках використання бездротових систем стало дешевшим, ніж використання альтернативних провідникових мереж.

#### ЛІТЕРАТУРА:

1. Chinh T. Delaunay Ctriangulation based complete coverage in wireless sensor networks / Chinh T. Vu, Yingshu Li // PERCOM '09 Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications. – 2009. С. 1–5.
2. Басюк Т.М. Основні підходи до побудови програмних засобів візуалізації даних / Т.М. Басюк // Вісн. нац. ун-ту "Львівська політехніка". - 2008. - № 631: Інформаційні системи та мережі. – С. 3-10.
3. Галкін П. В. Аналіз моделей та оптимізації збору інформації в бездротових сенсорних мережах / П. В. Галкін // Восточно-Европейский журнал передовых технологий. – 2014.– Т.5, №9(71). – С. 24-30.
4. Карпінський В. М. Перспективні засоби моделювання бездротових сенсорних мереж для мінімізації енерговитрат [Текст] / В. М. Карпінський, С. Балабан, В. Чиж // Матеріали першої науково-технічної конференції «Інформаційні моделі системи та технології». – Тернопіль, 20 травня 2011 р. – С. 36.
5. Мочалов В. А. Разработка и исследование алгоритмов построения отказоустойчивых сенсорных сетей ; дис. к. т. н.: 05.12.13 / В. А. Мочалов. – Москва, 2011. – 164 с.
6. Патент на корисну модель 64391 Україна: МПК H04W 12/00 / Карпінський В.М., Євтух П.С., Боровік Б.Л., Карпінський М.П.; власник патенту Тернопільський національний технічний університет ім. І. Пулюя. – No u 2011 03578; заявл. 25.03.11; опубл. 10.11.2011, Бюл. No 21. – 4 с.
7. Тимков А.В. Оптимізація бездротових сенсорних систем за допомогою мурашиних алгоритмів: автореф. квал. роб. магістра: 01.12.10 / А.В. Тимков – Донецьк, 2010. – 15с.

П'єнтий А.