

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Кафедра програмних систем і технологій

Прикладні програмні технології Internet of Everything, як джерело BigData

За редакцією В.Л. Шевченка

2021

УДК 002.5:004

T-39 Прикладні програмні технології Internet of Everything, як джерело BigData / за ред. Шевченка В.Л. – К.: КНУ імені Тараса Шевченка, 2021. – 248 с.
ISBN 966-8166-06-X

Монографію присвячено технологіям Інтернету всього (Internet of Everything, IoE), які є основним джерелом створення масивів великих даних (BigData). Розглянуті закономірності розвитку, галузі впровадження та конкретні приклади програмних технологій IoE. Монографія базується на результатах НДР кафедри програмних систем і технологій КНУ імені Тараса Шевченка (№ ДР 0117U002908, 2017-2020 роки). Робота адресована викладачам, науковцям, аспірантам, студентам та інженерам.

Монографію підготовлено **колективом авторів:**

Берестов Деніс Сергійович, к.т.н. (п.3.3);
Бичков Олексій Сергійович, д.т.н., проф. (вступ, пп. 1.2, 4.3);
Бражиненко Максим Геннадійович (п.4.1);
Григорьєва Тетяна Анатоліївна (пп.1.3, 1.4);
Громенко Віталій Юрійович (п.3.1);
Іванов Євген Вячеславович, к.ф.-м.н. (п.1.1.2);
Карнацький Дмитро Олегович (п.4.3);
Климко Вадим Вікторович (п.2.1, 2.2, 2.3);
Козачок Павло Анатолійович (п.4.2).
Курченко Олег Анастасійович, к.т.н., доц. (п.3.3);
Мірненко Володимир Іванович, д.т.н., проф. (п.3.1);
Мирошников Сергій Олексійович (п.4.3);
Мойсеєнко Михайло Олександрович (п.2.4);
Петрівський Володимир Ярославович (пп.1.1, 1.2, 2.5);
Радзівілов Григорій Данилович, к.т.н., доц. (пп.1.3, 1.4);
Ткаченко Максим Васильович, к.т.н. (п.1.3.3, 3.3);
Федоренко Руслан Миколайович, к.е.н. (п.3.3);
Шевченко Аліна Віталіївна (пп.3.2, 3.4);
Шевченко Віктор Леонідович, д.т.н., проф. (висн., розд.1, пп.3.1, 4.1, 4.2).

Рекомендовано до видання Вченою радою Військового інституту телекомунікацій та інформатизації імені Героїв Крут (протокол № 15 від 06 липня 2021 року)

- Рецензенти
1. Лисенко Олександр Іванович, д.т.н., проф., професор кафедри телекомунікацій Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
 2. Романюк Валерій Антонович, д.т.н., проф., професор кафедри Військового інституту телекомунікацій та інформатизації імені Героїв Крут

При повному або частковому відтворенні матеріалів посилання на видання попереджує плагіат та засвідчує рівень наукової культури.

© кафедра програмних систем і технологій КНУТШ, 2021

ЗМІСТ

СКРОЧЕННЯ	6
ВСТУП.....	7
1 Інтернет всього (Internet of Everything – ІоЕ). Загальний огляд стану та перспектив розвитку	8
1.1 ІоТ, ІоЕ. Історія. Тенденції. Визначення	8
1.1.1 Визначення	8
1.1.2 Суміжні поняття. Кібер-фізичні системи	15
1.1.3 Історія та еволюція Інтернету всього.....	17
1.1.4 Тенденції та перспективи еволюції.....	23
1.1.5 Порівняння з суміжними технологіями.....	33
1.1.6 Ризики та питання безпеки.....	37
1.1.7 ІоЕ, ІоТ: менеджмент та кадрові ресурси.....	42
1.1.8 Економічні показники	42
1.2 ІоЕ – поточні задачі та технології	44
1.2.1 Що потрібно для реалізації переваг ІоЕ, ІоТ	44
1.2.2 Яких інновацій та нових рішень потребує ІоЕ	47
1.2.3 Технології та технічні рішення ІоЕ.....	50
1.3 ІоЕ – приклади сфер впровадження	58
1.3.1 Загальний огляд.....	58
1.3.2 Промисловість	61
1.3.3 «Розумні» дома.....	62
1.3.4 «Розумні» міста	75
1.3.5 Нафтогазова галузь	80
1.3.6 Телекомунікації. ІоЕ. ІоТ	80
1.3.7 Медіабізнес	81
1.3.8 Роздрібна торгівля	81
1.3.9 Сфера послуг	83
1.3.10 Охорона здоров'я	83
1.4 ІоЕ в навчальних програмах	84
2. ВБУДОВАНІ ТА МОБІЛЬНІ СИСТЕМИ МОНІТОРИНГУ НЕБЕЗПЕЧНИХ СИТУАЦІЙ ЯК СКЛАДОВА ІОЕ.....	87
2.1 Програмне забезпечення виявлення небезпечних для користувачів ситуацій на основі аналізу даних вбудованих сенсорів мобільних пристроїв	87
2.1.1 Вступ	88
2.1.2 Аналіз роботи сенсорів та створення прототипу системи.....	90
2.1.3 Проектування та розробка.....	99
2.1.4 Додаткові можливості покращення роботи системи	111
2.2 Очищення даних від шуму.....	111
2.2.1 Рухоме середнє.....	112
2.2.2 Фільтри нижніх частот	114
2.2.3 Аналіз результатів очищення даних від шуму.....	116
2.3 Розширення можливостей сенсорів	116

2.3.1 Розширення діапазону максимально можливих значень.....	117
2.3.2 Збільшення частоти зняття даних з сенсорів	119
Висновки до розділів 2.1-2.3	123
2.4 Програмне забезпечення аналізу та прогнозування стану людини під час тренувань за допомогою сенсорів мобільних пристроїв	125
2.4.1 Вступ	125
2.4.2 Постановка задачі	125
2.4.3 Аналіз об'єкту дослідження.....	127
2.4.4 Використані програмні засоби	131
2.4.5 Вимоги до апаратного та програмного забезпечення	132
2.4.6 Створення та налагодження програми	132
2.4.7 Опис програми та її алгоритмів.....	139
2.4.8 Інструкція користувача	140
Висновки до розділу 2.4	145
2.5 Збільшення часу роботи датчиків небезпечних ситуацій шляхом регулювання енерговитрат.....	146
Висновки до розділу 2.5	151
3. ЗАХИЩЕНА СИСТЕМА АВТОМАТИЧНОГО КЕРУВАННЯ БЕЗПЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ ЯК ТЕХНОЛОГІЯ ІОЕ.....	152
3.1 Математична модель БПЛА.....	152
3.2 Послідовність нарощування рівнів захищеності за етапами розвитку системи управління БПЛА	156
3.3 Створення КСЗІ в автоматизованих системах управління БПЛА	160
3.4 Ідентифікація загроз «нульового дня» для цифрової системи управління за допомогою таксонометричного підходу.....	166
Висновки до розділу 3	172
4. ОБРОБКА 3-D СИГНАЛІВ МЕДИЧНОГО ПРИЗНАЧЕННЯ В ІНТЕРЕСАХ ІОЕ	173
4.1 Побудова сучасного сервіс-орієнтованого додатку для аналізу динамічних даних медичного призначення в інтересах Інтернету всього ..	173
4.1.1 Вступ	173
4.1.2 Аналіз транспортного шару повідомлення	174
4.1.3 LoRaWAN як транспорт для медіа-контенту.....	181
4.1.4 Розрахунок технологічних обмежень.	190
Висновки до розділу 4.1	191
4.2 Архітектура додатку аналізу медичних динамічних даних інтернету всього на основі мікросервісної архітектури	192
4.2.1 Вступ	192
4.2.2 Огляд базових архітектур.....	193
4.2.3 Огляд та аналіз існуючих рішень в сфері безсерверних обчислень ...	197
4.2.4 Огляд запропонованого рішення.....	199
4.2.5 Користувацький інтерфейс	209
Висновки до розділу 4.2	212

4.3 Обробка даних щодо 3-D моделювання руху крові в серцево-судинній системі людини.....	213
4.3.1. Використання клітинних автоматів та решітчастого газу Больцмана для моделювання потоків рідини	213
4.3.2. Застосування решітчастого газу Больцмана для моделювання руху крові в системі судин.....	228
Висновки до розділу 4.3	233
ВИСНОВКИ.....	234
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	239

СКОРОЧЕННЯ

AJAX	Asynchronous JavaScript And XML (підхід до побудови користувацьких веб інтерфейсів)
API	Application Programming Interface (Програмний інтерфейс додатків)
AWS	Amazon Web Services
AWS EC2	Amazon Elastic Compute Cloud (Amazon EC2) - веб-сервіс, що надає безпечні масштабовані обчислювальні ресурси в хмарі
BYOD	Bring Your Own Device (Приносьте свій власний пристрій)
HTTP	Hypertext Transfer Protocol (протокол передавання гіпертексту)
IDC	International Data Corporation
IoE	Internet of Everything (Інтернет всього, Всеохоплюючий Інтернет)
IoT	Internet of Things (Інтернет речей)
IT	Information technology (Інформаційні технології)
JSON	JavaScript Object Notation (текстовий формат обміну даних)
LPWAN	широкополосна мережа малої потужності
M2M	Machine to Machine (З'єднання машина з машиною)
MSSIM	Mean Structural Similarity Index Method – метод середнього індексу структурної схожості
MVC	модель-вид-контролер
PSNR	Peak-signal-to-noise-ratio – співвідношення сигналу до шуму
QoS	Quality of Service - Якість обслуговування
REST	Representational State Transfer «передача репрезентативного стану» - стиль проектування розподілених систем за допомогою обмежень
RFID	Radio Frequency Identification (радиочастотная ідентифікація)
TCP	Transmission Control Protocol (протокол, який надає можливість створювати надійні канали передачі даних)
UDP	User Datagram Protocol (мережний протокол транспортного рівня з набору протоколів TCP/IP без надійного з'єднання)
АС	Автоматизована система
ІКТ	Інформаційно-комунікаційні технології
ІС	Інформаційна система
КСЗІ	Комплексна система захисту інформації
КФС	cyber-physical system, CPS
НСД	Несанкціонований доступ
ОІД	Об'єкт інформаційної діяльності
ОС	Операційна система
ПЗ	Програмне забезпечення
ПК	Персональний комп'ютер
ПКМ	Програмно конфігуруєма мережа
ТЗ	Технічне завдання
ТЗІ	Технічний захист інформації

ВСТУП

У 2012 році компанія Cisco ввела нове поняття Інтернет Всього » (Internet of Everything, IoE, всеосяжний, всеохоплюючий Інтернет). Окремі елементи цього поняття вже існували на ринку та в дослідницьких лабораторіях. Але як цілісне явище поняття Інтернету всього було сформоване лише в означений термін.

Попередниками Інтернету Всього були низка технологій, найголовнішими серед яких були Інтернет та Інтернет Речей. Поняття Інтернету речей з'явилося відносно недавно у 1999 році, але темпи розвитку інформаційного суспільства сьогодні є настільки високими, що за цей стислий час Інтернет речей зміг не тільки здобути заслужену славу, але й породити гідного наслідувача, який піднімає якісні та кількісні показники інформаційного суспільства на принципово новий рівень.

При такій швидкій зміні технологій дуже важливо якнайшвидше визначитись з найважливішими напрямками досліджень та розвитку технологій. Оскільки тільки витримування найвищого темпу забезпечить конкурентоздатність наукових результатів.

Порядок кількості об'єктів, які ведуть обмін інформацією в технології Інтернету Всього збільшується безпрецедентними темпами, що створює протиріччя між реальною потребою практики щодо розвитку таких технологій і відсутністю теоретичного підґрунтя розвитку таких технологій, зокрема в галузі створення надійного, безпечного та ефективного програмного забезпечення Інтернету Всього.

Виходячи з наукової новизни і високої практичної значущості розвитку технологій Інтернету Всього можна зробити висновок про високу актуальність досліджень за означеною темою.

Об'єкт дослідження: Інтернет всього (він же Всеохоплюючий Інтернет).

Мета роботи: Визначення основних напрямків та розробка концептуальних підходів щодо розвитку математичних та програмних технологій Інтернету всього. Апробація розроблених підходів на прикладах окремих галузей застосування таких, як системи моніторингу небезпечних ситуацій, системи захищеного керування безпілотними літальними апаратами та технології обробки 3-D сигналів медичного призначення.

1 Інтернет всього (Internet of Everything – IoE). Загальний огляд стану та перспектив розвитку

1.1 IoT, IoE. Історія. Тенденції. Визначення

1.1.1 Визначення

Інтернет речей (Internet of Things, IoT) - це комп'ютерна мережа, що з'єднує різні пристрої підключені до інтернету [13]. В даний час, «Інтернет речей» плавно переходить в «Інтернет всього» або «Загальний Інтернет» (Internet of Everything, IoE), інша назва - «Всеохоплюючий Інтернет». Загальний Інтернет об'єднує не тільки неживі предмети, але й людей, процеси, дані.

Концепція IoT з'явилася в 90-і роки і мала на увазі оснащення фізичних об'єктів технологіями взаємодії з навколишнім середовищем [44]. Концепція IoE з'явилася пізніше, разом з повсюдним поширенням бездротової мережі Wi-Fi, і з тих пір розвивається з шаленою швидкістю. Концепція всеосяжного інтернету передбачає, що в найближчому майбутньому мережа об'єднає собою не тільки тих, хто, але і вдома, автомобілі, побутові прилади, виробничі процеси. Додаткові прибутки, які будуть згенеровані при переході компаній в «режим» всеосяжного інтернету очікуються на вражаючому рівні 19 трлн доларів в найближче десятиліття. Cisco визначає IoE як об'єднання людей, процесів, даних і речей (Рис. 1.1.1), що підвищує цінність мережевих з'єднань до небувалого рівня [72, 73, 74].



Рис. 1.1.1 Структура IoE згідно Cisco.

IoE перетворює інформацію в конкретні дії, що створюють нові можливості, що розширюють досвід користувача і формують сприятливі умови для розвитку країн, компаній і індивідуальних користувачів. Таке визначення підкреслює важливий аспект IoE, який відрізняє IoE від IoT - так званий "мережевий ефект".

Оксфордський словник, визначає IoT як «передбачуваний розвиток мережі Інтернет, при якому звичайні предмети мають можливість підключатися до мережі, що дозволяє їм відправляти і отримувати дані» [76].

Всеохоплюючий Інтернет зовсім не одне і те ж, що IoT. IoT - лише складова частина IoE. IoE ґрунтується на зв'язках між людьми, процесами, даними і речами, причому мова не йде про кожного з цих вимірів окремо. Кожне з них підсилює можливості трьох інших, і справжня сила IoE проявляється на їх перетині. Розглянемо складові IoE докладніше.

Люди [77]. За останні три десятиліття інструменти виходу в Інтернет, від непрограмований терміналів до настільних комп'ютерів і різноманітних мобільних пристроїв, включаючи лептопи, смартфони та планшети, зазнали радикальні зміни. Але все це не йде ні в яке порівняння з новою хвилею перетворень.

Google-окуляри і «розумні» годинник - тільки початок цілого ряду «носимих» технологій, які докорінно змінять наші моделі споживання і обміну інформацією. У нас вже є самоконтролюючою пристрої, такі як трекер Fitbit і браслет Nike FuelBand, що дозволяють відслідковувати наші фізичні вправи, стежити за частотою серцебиття і навіть за якістю сну. У найближчі роки ці можливості помітно зростуть [77]. З'явиться таблетка, проковтнувши яку, ми зможемо контролювати стан нашого травного тракту і відправляти відповідну інформацію своєму лікарю в потрібний час і в контексті того, чим ми займаємося в даний момент. Майбутні матері будуть носити «розумні татування», яке контролюватиме здоров'я і активність їх плода і посилятиме лікарю вчасний сигнал про те, що пологи почалися ... Ми тільки-тільки починаємо розуміти, як носимі технології змінять наше життя.

Процеси [77]. З підключенням процесів ми дещо знайомі, хоча Інтернет вже зробив революцію в тому, як підприємства управляють своїми ланцюгами поставок, і в тому, як споживачі роблять покупки.

У міру того, як наш світ буде ставати все більш технологічним, нам будуть все більш очевидними процеси, які недоступні нашому розумінню зараз, і в результаті можна буде зробити ці процеси простіше, швидше і ефективніше.

Наприклад, компанія Cisco співпрацює з великими роздрібними компаніями в області застосування датчиків, відео та аналітики для підвищення рентабельності магазинів і поліпшення користувацького досвіду. Відеокамери і датчики на парковках можуть вести облік прибуття автомобілів і кількості відвідувачів, що заходять в магазин. У поєднанні з даними, які надходять з сенсорів на візках для покупок і в результаті аналізу пересувань покупців по магазину, система може передбачати піки і, навпаки, спад купівельної активності, щоб автоматично регулювати необхідну кількість касирів. Покупці, таким чином, позбавляються від довгих черг в касах, а магазин, в свою чергу, може оптимізувати продуктивність персоналу.

Всеохоплюючий Інтернет змінює моделі того, як люди і речі пов'язуються один з одним, як ми отримуємо і використовуємо дані і як все це взаємодіє для здійснення інтелектуальних процесів.

Дані [77]. Світ переповнює інформація. Після 2013 року кожні 10 хвилин створюється більше нової інформації (в основному, завдяки мультимедіа), ніж за всю історію людства аж до 2008 року. Обсяг інформації збільшується в геометричній прогресії, і при цьому зростає цінність самої інформації. Так, ми починаємо переходити від HD відео (720/1080 пікселів) до ще більш чіткого відео (дисплеї 4K). Вартість таких дисплеїв вже впала нижче 1 000 доларів, що зробило їх доступними, як ніколи раніше. Сенсори камер теж стають більш точними. Наприклад, в 2019 році компанії Samsung та Xiaomi разом розробили фотосенсор для смартфона на 108 мегапікселів.

Ці досягнення впроваджуються швидко, і зростання нових технологій і їх споживання означає, що інформаційний ураган буде лише зростати. Цей обсяг великих даних створюють не тільки одні мультимедійні технології. Нові, ніколи раніше не існували види пристроїв починають породжувати ще більше даних - наприклад, датчики на продуктах харчування, що стежать за їх терміном придатності. Аналітика Big Data допомагає нам розібратися в цій лавині інформації, ідентифікуючи і комбінуючи відповідні дані таким чином, що це дозволяє поглянути на речі по-новому і приймати більш зважені рішення.

Речі [77]. Сьогодні до Інтернет підключені менше 1 відсотка (близько 10 мільярдів) існуючих в світі речей. Протягом наступного десятиліття це число зросте до 50 мільярдів з гаком, і тоді на кожного жителя Землі, включаючи чоловіків, жінок і дітей, припадатиме як мінімум шість підключених пристроїв. Але примітніше всього буде не те, що збільшиться кількість речей на зразок комп'ютерів, телефонів і планшетів, які, як ми очікуємо, виявляться підключені до Інтернету. Куди важливіше те, що будуть підключені (і вже починають підключатися) найнесподіваніші предмети.

Підключення автомобілі приведуть до створення самокерованих машин.

«Розумні» будівлі будуть самі себе обслуговувати, тим самим сприяючи створенню більш «розумних» і більш ефективних міст.

Підключення системи водопостачання зможуть виявляти протікання в трубах і перекривати вентиля на невикористовуваних трубопроводах, що скоротить на 30% втрати води.

Будуть підключені навіть корови разом з їх пасовищами, що допоможе фермерам і тваринникам точніше визначати час поливу, внесення добрив і перегону худоби на інші випаси - і все це буде сприяти підвищенню продуктивності таких господарств і скорочення відходів.

Головний футуролог Cisco Дейв Еванс (Dave Evans) ще в 2012 році стверджував наступ «найцікавішого періоду в історії людства - переходу від Інтернету речей (Internet of Things, IoT), в якому ми живемо сьогодні, до «всеосяжного Інтернету» (Internet of Everything, IoE)» [10]. «Ми вступаємо в

епоху, коли Інтернет може радикально поліпшити життя кожного жителя Землі - вилікувати важкі хвороби, допомогти вирішенню кліматичних проблем, удосконалити процеси ведення бізнесу і зробити кожен день нашого життя більш щасливим» [10].

Cisco описує ІоЕ як середовище, де всілякі неживі предмети будуть враховувати контекст і користуватися більш широкими обчислювальними ресурсами і сенсорними можливостями. Дейв Еванс додає в цю формулу людей і інформацію, отримуючи «мережу мереж, в якій мільярди, навіть трильйони з'єднань створюють безпрецедентні можливості, в тому числі дають дар мови речам, які ніколи раніше не мали власного «голосу». У вищенаведених термінах багато загального, але вони принципово відмінні. Наочне розмежування вищезазначених термінів дає схема з роботи IoT-Analytics (Рис. 1.1.2) [37].

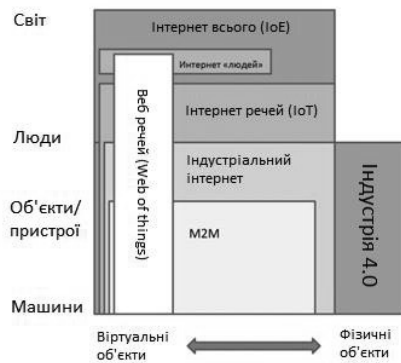


Рис 1.1.2. Співвідношення термінів, пов'язаних з поняттям «Інтернет речей» (джерело Wikipedia, McKinsey, IoT-Analytics).

У центрі схеми лежить М2М-технологія, яка відома досить давно і в найзагальнішому плані визначається як технологія, що дозволяє машинам обмінюватися даними по дротових і бездротових технологій [37]. Згідно малюнку М2М є підмножиною індустріального Інтернету (Industrial Internet). Термін «індустріальний Інтернет» ширше, ніж М2М, і описує технології, які концентруються не тільки на комунікаціях «машина-машина», але також на наданні інтерфейсу людині. Безліч технологій, що описується терміном Industry 4.0 (термін походить від четвертої промислової революції), являє собою розвиток концепції індустріального Інтернету і має на увазі створення розумних виробництв, які базуються на цифрових технологіях, включаючи хмарні обчислення, ІоТ, 3D-принтери і доповнену реальність.

ІоТ - ширше, ніж поняття індустріального Інтернету, оскільки включає споживчі рішення, наприклад носимі людиною ІоТ-пристрої [37].

IoE, на думку IoT-Analytics, поки що досить розпливчате поняття, яке прагне описати набір технологій, що забезпечують всі види з'єднань, і таким чином має найвище охоплення (рис. 1.1.2) [37].

Картину доповнює прямокутник, який описує безліч технологій, що відповідають терміну «Web речей» (Web of Things, WoT), та який перетинається з усіма раніше перерахованими блоками. WoT - це набір сервісів прикладного рівня для створення додатків «Інтернету речей. WoT включає принципи, архітектурні стилі та програмні шаблони, які дозволяють об'єктам реального світу стати частиною Всесвітньої павутини [37].

Невеликий розмір прямокутника "Інтернет людей" підкреслює, що кількість підключених пристроїв, керованих за безпосередньої участі людини, становить все менший обсяг. Але навіть при наявності формальної узгодженості вищевказаних термінів залишаються різночитання в деталях. Найбільш повно ці розбіжності проаналізовані в публікації компанії IDTechEx. IDTechEx досліджувала дані різних аналітиків і дала графічну інтерпретацію тлумачення IoT-екосистеми, підкреслюючи, що кількісна оцінка ринку IoE залежить від того, як він визначається [37].

На рис. 1.1.3. показана спрощена схема, яка містить три підмножини [37].

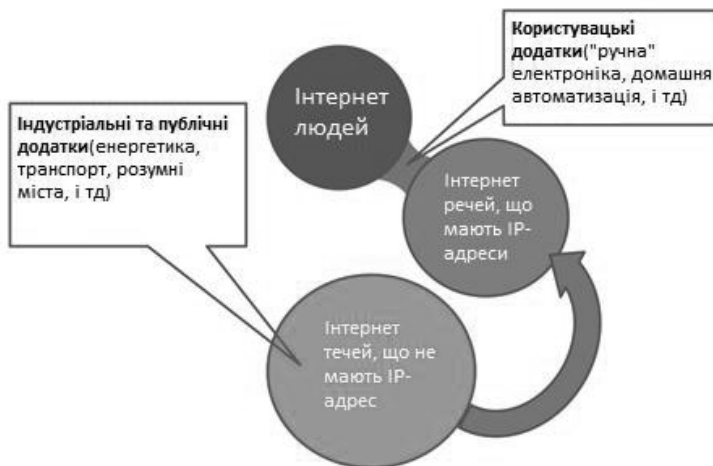


Рис.1.1.3. Таксономія ринку IoT за даними IDTechEx.

Перше - це так званий Інтернет людей (Internet of People (IoP)), до даної безлічі відносяться смартфони та інші персональні електронні пристрої з виходом в Інтернет [37].

Наступна за радіусом окружність характеризує рішення на базі інтелектуальних пристроїв (речей), які включають різного роду датчики для фіксації даних та мікроконтролери для локальної обробки цих даних. Ключовим моментом є те, що ці пристрої мають IP-адреса і можуть взаємодіяти з іншими речами без участі людини. Як видно з малюнка, перша і друга окружність з'єднані перемичкою, що позначає перехідну область, яка має властивості обох множин, але формально не належить їм. Саме до цієї області віднесені споживчі програми, такі як носима електроніка, додатки для здоров'я та фітнесу, засоби домашньої автоматизації. Частина аналітиків відносить ці рішення до IoT-ринку, частина не відносить, посилаючись на те, що в більшості цих рішень комунікації вимагають участі людини. Слід зазначити, що аж ніяк не завжди потрібні рішення на базі IP-датчиків. Існують, наприклад, дуже дешеві рішення на базі пасивних RFID-міток, які в порівнянні з класичними IoT-пристроями набагато дешевше. І рішень, що використовують «не IP-речі», безліч [37].

Тому третя, найбільша окружність визначає безліч спеціалізованих "речей", які не мають IP-адреси на рівні цих "речей" і використовують інші мережі та інші протоколи обміну даними. Їх не можна в повній мірі віднести до IoT, хоча деякі аналітики відносять їх до цієї категорії, що і призводить до різних оцінок IoT-ринку. Проте, як видно з рис. 1.1.3, загальна тенденція така, що поступово відбувається перехід з рішень на базі «не IP-датчиків" (які мають ряд обмежень, включаючи проблему залежності від конкретного постачальника) на універсальні рішення на базі IP-датчиків. На рис. 1.1.3 даний процес позначений стрілкою [37].

IoE відкриває перед організаціями, окремими людьми, спільнотами і цілими країнами безпрецедентні перспективи з точки зору можливості отримувати набагато більшу користь з мережевих з'єднань між людьми, процесами, даними і матеріальними об'єктами [4]. Цінність IoE пояснюється сукупним ефектом таких з'єднань і тією користю, яку вони будуть приносити в міру того, як все більше людей, процесів, даних і фізичних предметів будуть охоплюватися Всесвітньою павутиною. Інтернетом же речей позначають лише мережеві з'єднання фізичних предметів, тобто він не включає такі компоненти IoE, як «люди» і «процеси». IoT є перехід від однієї технології до іншої, а IoE - перехід до безлічі технологій, включаючи IoT.

Зазвичай, коли люди користуються словом Інтернет, навряд чи вони можуть собі уявити фізичні з'єднання в реальному світі, що утворюють цю мережу [31]. Швидше Інтернет представляється людям як якась колекція підключень до тих чи інших джерел інформації. Це «місце», куди люди можуть прийти для знаходження інформації або для того, щоб поділитися інформацією. У сучасному уявленні, Інтернет - це бібліотека XXI століття, відеомагазин, персональний фотоальбом. Ті зміни, які вніс Інтернет в сьогоденне життя, складають невелику частину того, що станеться, коли весь фізичний світ буде підключений до мережі Інтернет. В цьому випадку і відбудеться перехід від Інтернету до «всеосяжної інтернету» - IoE. Це ера розумних світлофорів і

парковок, електричних лампочок з інтернет-адресами, підключених до Інтернету автомобілів, інтелектуальних систем водопостачання, вивезення сміття і багато, багато іншого.

Основою функціонування ІоЕ є комп'ютерні мережі різного призначення і розміру [31]. ІоЕ, як і Інтернет, є «мережею мереж». Кінцевими пристроями мереж є комп'ютери, мережеві принтери, ІР-телефони, обладнання відеоконференцій, охоронні відеокамери, мобільні персональні пристрої (смартфони, планшети), пристрої читання кредитних карт, сканери штрих-кодів, датчики різного призначення та інше. Ці пристрої і складають базове поняття «речей» в архітектурі ІоЕ. До 2012 року кількість «речей», підключених до Інтернету, перевищила число людей на планеті і далі продовжує дуже швидко рости. Однак при цьому фахівці вважають, що 99% фізичних об'єктів, які одного дня стануть частиною ІоЕ, досі ще не підключені і не використовуються.

Особлива роль в архітектурі ІоЕ відводиться датчикам. Вони призначені для збору даних від пристроїв, які не є комп'ютерами, і їх передачі комп'ютерам для обробки. Прикладами датчиків є: датчики вологості, датчики температури повітря, радіаційні датчики, датчики руху і т.д. Все більш поширеним типом датчиків є мікрочіпи, що функціонують за принципом радіочастотної ідентифікації (RFID). Є величезна кількість прикладів використання таких датчиків. Останньою досить поширеною практикою у автовиробників стає установка в кузові автомобіля RFID-мікрочіпа, що містить всю інформацію про автомобіль: дата і місце виробництва, вироблені сервіси, власники і інша інформація. Сучасні RFID-мікрочіпи, реалізовані на основі інтегральних мікросхем, мають життєвий цикл до 50 років з можливістю виконання до 100 тис. циклів перезапису. Внаслідок великої гнучкості у використанні і низького енергоспоживання, RFID-мікрочіпи надають великі можливості для підключення "не-комп'ютерів» до ІоЕ.

Різномайття визначень. На початку 1990-х років суть кожної нової технології представлялася, як ворота в інформаційну супермагістраль [53]. Насправді такі види аналогій не допомагали дати точне визначення Інтернету. Інтернет виявився новим способом поділу Вашому життю з іншими людьми, нової бізнес-моделлю здійснення покупок, а також новим способом дати освіту Вашим дітям. Інтернет визначається більшою мірою тим, що можете робити Ви, а не тим, що робить він. Від подібного непорозуміння страждає і «Інтернет речей».

ІоТ. Великі компанії - гравці на полі інформаційних технологій - визначають поняття «Інтернету речей» по-різному.

Cisco називає його «Інтернетом всього» (або «Всеосяжним Інтернетом» - Internet of Everything) і каже, що це буде «остання хвиля Інтернету - з'єднання фізичних об'єктів для досягнення більш високого рівня безпеки, комфорту та ефективності».

ІВМ описує «Інтернет речей» як «абсолютно нову всесвітню павутину, що складається з повідомлень, які можуть передавати між собою пристрої в цифровій формі. Це той же самий Інтернет, але інша мережа».

General Electric (GE) використовує концепцію «Промисловий Інтернет» (Industrial Internet) - мабуть, найбільш захоплюючу, оскільки вона дає можливість безпосередньо представити себе нові додатки. З точки зору GE промисловий Інтернет являє собою «об'єднання агрегатів і інтелектуальних даних з метою створення геніальних машин».

Компанія RTI має девіз - **Your Systems. Working as One**, тобто «Ваші системи. Працюючи як один». У компанії наполягають на тому, що «Інтернет речей» буде являти собою абсолютно новий сервіс. Він буде таким же ґрунтовним, як мережа, GPS або сам Інтернет.

Компанія Qualcomm розглядає ІоЕ [12] як зонтичний термін, що визначає нові можливості ринку, пов'язані зі зростаючою кількістю підключень, новими джерелами інформації, М2М-з'єднаннями і наступним поколінням пристроїв.

Кожен описує майбутнє по-своєму, але всі сходяться на думці, що ІоЕ, ІоТ і інтелектуальні системи, які вони дозволять створити, принципово змінять наш світ.

1.1.2 Суміжні поняття. Кібер-фізичні системи

Тісно пов'язаними з концепціями ІоТ та ІоЕ є поняття кібер-фізичної системи (КФС, cyber-physical system, CPS). Даний термін набув широкого вжитку починаючи з 2006 року [56, 63, 91], коли Національний науковий фонд США (National Science Foundation, NFS) почав використовувати його для позначення ключового напрямку перспективних досліджень щодо інтеграції обчислювальних та фізичних процесів та створення складних “розумних” систем, що поєднують комп'ютери, комп'ютерні мережі, програмне забезпечення та фізичні процеси. Значення даного терміну може бути пояснене такою цитатою Хелен Гілл (Helen Gill) з National Science Foundation:

“Cyber-physical systems are physical, biological, and engineered systems whose operations are integrated, monitored, and/or controlled by a computational core. Components are networked at every scale. Computing is deeply embedded into every physical component, possibly even into materials. The computational core is an embedded system, usually demands real-time response, and is most often distributed.” [88]

З цього пояснення зрозуміло, що КФС можуть розглядатися як розвиток поняття вбудованої системи (embedded system) та його вихід на новий рівень, що підкреслює розподілену організацію та тісну взаємодію із фізичним оточенням. Прикладами КФС вважаються автономні автомобілі, автономні пошукові роботи, автономні безпілотні літальні апарати, роботизоване медичне обладнання, “розумні електромережі” (smart grid), розумні будинки, розумні міста і таке інше.

Із визначення та прикладів очевидно, що КФС має перетин із раніше введеними концепціями повсюдних обчислень / Ubiquitous computing (M. Weiser, 1988), та Internet of Things (K. Ashton, 1999). Однак є деякі відмінності. Однією з головних проблем на шляху до створення КФС вважається вироблення продуктивного способу поєднання абстракцій та підходів до проектування систем, які застосовуються інженерами / розробниками фізичних компонент кібер-фізичних систем з одного боку та розробниками програмного забезпечення з іншого боку. Спрощений варіант формулювання даної проблеми може бути знайдено у такій цитаті із карти концепцій КФС [68], запропонованої S.S. Sunder з Національного інституту стандартів і технологій США (NIST) та рядом інших авторів:

“A key CPS challenge is to conjoin the engineering abstractions for continuous dynamics (such as differential equations) with computer science abstractions (such as algorithms).” [68]

Акцент на даній проблемі дозволяє вирізнити КФС як галузь наукових досліджень з-поміж ряду суміжних та подібних галузей, а також призводить до певного розділення наукових спільнот, що займаються дослідженнями у галузях IoT/ІоЕ та КФС. Ілюстрація даного розділення може бути знайдена у [88]. Проте, з точки зору охоплення різноманітних типів пристроїв та програмно-апаратних рішень, у даний час поняття КФС та IoT можуть розглядатися більш-менш як синоніми [88].

1.1.3 Історія та еволюція Інтернету всього

1.1.3.1 Загальна історія Інтернет

У березні 1989 року Сер Тімоті Бернерс-Лі, науковий співробітник Європейської лабораторії з ядерних досліджень (CERN), придумав концепцію семантичної павутини (Всесвітньої павутини) як рішення практичного завдання щодо відстеження комплексних великомасштабних проєктів, не втрачаючи при цьому важливу інформацію [14].

Сьогодні Всеохоплюючий Інтернет не обмежується підключенням одних тільки даних. Конвергенція підключень людей, об'єктів, даних і процесів перетворює організації, галузі промисловості та все наше життя. Розвиток мобільних технологій і хмарних обчислень ще більше сприяє інноваціям і подальшому зростанню найрізноманітніших з'єднань [14].

На початку 1990-х років Інтернет з'єднував між собою тільки стаціонарні пристрої [30]. Для виходу в Інтернет треба було сісти за робочий стіл, на якому стояв персональний комп'ютер, "тонкий клієнт" або інший фіксований пристрій доступу. На піку цього етапу в кінці минулого століття до Інтернету підключилося 200 млн пристроїв.

На початку нового століття пристрої стали переміщатися разом з власниками. Перші стільникові телефони були розміром з цеглину? Швидко зростання кількості фіксованих і мобільних пристроїв призвело до істотного збільшення числа мережевих з'єднань. Сьогодні до Інтернету підключено близько 10 млрд пристроїв. В результаті виник Інтернет речей. Д.Еванс називав цю епоху "Століттям пристроїв".

Щоб зрозуміти масштаб перетворень, згадаємо, як ми жили всього 20 років тому. Загальнонаціональне опитування, проведене в Сполучених Штатах центром Pew Research в зв'язку з 25-ою річницею Всесвітньої павутини, показав, що в 1995 році 42% дорослого населення США навіть не чули про Інтернет, 21% мали про нього лише туманне уявлення, знаючи тільки, що це якось пов'язано з комп'ютерами, а доступ до Інтернету мали лише 14% американців [14].

Сьогодні життя докорінно відрізняється від того, що було в середині 1990-х. Всесвітня павутина трансформувалася і стала Всеосяжним Інтернетом, де практично все або вже підключено, або буде підключено до Інтернету. Від розумних автомобілів і окулярів з доступом в Інтернет до дерева, яке веде свій твіттер, Всеохоплюючий Інтернет не тільки підключає непідключене, але і допомагає витягувати величезну вигоду з цих підключень.

Основні етапи еволюції Інтернету [31]

1-й етап. Підключення до джерел інформації (з середини 1980 років)

Перехід до цифрового доступу до інформації:

- електронна пошта;
- Web-браузери;
- пошук в Інтернеті.

2-й етап. Реалізація економічних вигод від мережі (з пізніх 1990 років)

Переклад бізнес-процесів в цифрову форму:

- електронна комерція;
- супутні цифрові програми;
- початку застосувань спільних мережевих практик.

3-й етап. Спільні практики (с ранніх 2000 років)

Переклад взаємодій (в бізнесі і соціальних) в цифрову форму (цифровизация суспільства):

- соціальні мережі: Facebook, Instagram, Twitter;
- мобільні пристрої;
- хмарні обчислення;
- відео-контент.

4-й етап. Всеохоплюючий інтернет (з 2013-2014 рр.)

Реалізація всіх взаємодій в світі в цифровій формі

- люди;
- процеси;
- дані;
- «речі».

Четверта фаза - всеосяжний інтернет охопить в чотири рази більше людей, речей і процесів, ніж все, що було створено до цього [44]. Світ став орієнтуватися саме на додатки, він став практично повністю підключеним до мережі.

При цьому в значній мірі зміниться і співвідношення джерел інформації:

- Зараз 65% інформації в мережі збирають користувачі, а на частку бізнесу доводиться тільки 35%,
- в найближчі роки, навпаки, 65% інформації буде надходити саме від підприємств і промисловості.

1.1.3.2 Частка історії Інтернет, яка була каталізована Cisco

Справжнім каталізатором змін останніх етапів Інтернету стала компанія Cisco, яка сформулювала принципові риси нового напрямку в ІТ - всеохоплюючий Інтернет. Сам по собі цей термін видається досить всеосяжним, що не містить ніякої конкретики, при цьому з плином часу і появою численних рішень від різних учасників ринку, він наповнюється різним вмістом [2].

В 1984 році подружня пара Леонард Босак (Leonard Bosack) і Сандра Лернер (Sandra Lerner) заснувала компанію Cisco Systems. Хоча Cisco не була першою компанією, яка розробляла і продавала маршрутизатори - пристрої, що перенаправляють комп'ютерний трафік з однієї мережі в іншу-вона створила перший комерційно успішний багатопрокольний маршрутизатор [2]. Цей пристрій дозволяв раніше несумісним комп'ютерам спілкуватись між собою, навіть якщо вони використовували різні мережеві протоколи. Можливість такого спілкування природним чином привела до того, що протокольний «звіринець» зник, компанія Cisco залишилася, а мережа стала **всеосяжною в**

географічному сенсі - все окремі майданчики (кампуси) з їх локальними мережами стали об'єднуватися. У цьому ж 1984 році була розроблена глобальна система доменних імен (Domain Name System, DNS), що забезпечує ідентифікацію мережевих ресурсів. Вже до 1984 року інтернет став всеосяжним в сенсі географії: локальні мережі почали об'єднуватися в єдиний глобальний інформаційний простір. Однак той Інтернет був всеосяжним вельми специфічно: доступ до нього забезпечувався через термінали великих ЕОМ, тобто для абсолютної більшості людей був закритий [2].

Але саме в цей же час з'явилися персональні комп'ютери. Вони не відразу отримали доступ в мережу, але коли це сталося, синергетичний ефект був просто приголомшливим. Так інтернет став **всеосяжним** в тому сенсі, що прорик **в будинки звичайних громадян** [2].

Спочатку це був комутований доступ, який не потребує ніякого спеціального обладнання на стороні оператора зв'язку, потім ADSL, який хоча і вимагає пристрою DSLAM на АТС, проте використовує колишні телефонні лінії [2]. Мабуть, саме ADSL зробив революцію: він привернув таку кількість клієнтів до Інтернету, що з'явився сенс прокладати спеціальні лінії, що забезпечують гарну швидкість в обидві сторони. І тут Інтернет стає **всеосяжним в плані наданих сервісів**. Комутований доступ дозволяв без напруги тільки отримувати і відправляти пошту: підготував листи в папці «Вихідні», натиснув кнопку «Відправити / Отримати» - і можеш досить довго чекати результату. Відкрити сайт і подивитися свіжу інформацію було вже складно. Розробники сайтів це знали і намагалися робити їх максимально аскетичними.

Саме в цей час формувалось усвідомлення нагальної потреби в мобільному інтернеті. Перші бездротові мережі в нашому географічному регіоні почали відкриватись в 2004 році [2].

Інфраструктуру забезпечувала компанія Cisco, а клієнтську частину Intel. (Intel Centrino) [2]. Мудрість була в тому, що Intel вирішила встановлювати бездротовий модуль на всі види продукції «примусово», а не опціонально, і таким чином досить швидко забезпечила для бездротових сервісів солідну клієнтську базу. Інтернет став **всеосяжним** в рамках **громадських будівель і особистих помешкань**.

У травні 2007 року компанія Fujitsu-Siemens (зараз Fujitsu) представила ноутбук з вбудованим модемом під SIM-карти. На той момент жоден з операторів не пропонував USB-модеми, тому пропозиція Fujitsu-Siemens була унікальною. Однак в цей час Intel планувала вивести на ринок свою технологію операторського зв'язку WiMax і не збиралася підтримувати стандарт 3G, тому рішення Fujitsu не отримало належного розвитку. Проте, оператори мобільного зв'язку перейшли до швидкостей інтернету 3G, потім 4G, і таким чином інтернет став всеосяжним на більшій частині обжитий території [2].

Всеосяжним Інтернет став за допомогою таких пристроїв: планшети, смартфони, пристрої, що трансформуються з смартфона в планшет, підключені годинник-смартфон або годинник-гарнітура [2].

У цей момент з'являється концепція Cisco Internet of Everything. І, як вже було згадано, дещо раніше з'явилася інша концепція під назвою Internet of Things. Дивно, але це нове словосполучення замигтіло вже після того, як багато звичних нам по своєму призначенню речей стало цифровими. Фотоапарат, телевізор, радіоприймач, телефон [2].

По-англійськи виходить каламбур: Things - Everything. Суть цих ідей в тому, щоб оснастити різні неживі об'єкти мережевими інтерфейсами (в основному бездротовими), через які вони могли б щось повідомляти іншим об'єктам (тоді це буде Internet of Things) або в тому числі взаємодіяти з людьми (Everything). Сукупність цих повідомлень після обробки може породжувати нову інформацію, що представляє інтерес або здатну запобігти можливим неприємностям.

Для розширення сфер застосування концепцій IoT, IoE, корпорація Intel запропонувала платформи Galileo і Edison як рішення для ентузіастів, які можуть їх застосовувати практично в будь-якій сфері людської діяльності без заздалегідь нав'язаних рекомендацій.

Таким чином [43] світ стає все більш «підключеним» - практично все починає з'єднуватися з усім на світі. Саме результатом повсюдної доступності зв'язку стала поява Всеохоплюючої Інтернету (Internet of Everything, IoE), тобто мережових з'єднань людей, процесів, даних і об'єктів.

1.1.3.3 Нова історія Інтернет: IoT, IoE

Інтернет повністю змінив способи комунікацій і спільної роботи людей [53]. Він ознаменував собою початок нової ери вільної інформації для кожного, трансформуючи життя до такої міри, яку важко було уявити раніше. Однак наступний етап розвитку Інтернету пов'язаний не з людьми. Він стосується інтелектуальних пристроїв з можливістю встановлення між ними зв'язку. Щоб успішно взаємодіяти з реальним світом, ці пристрої повинні спільно працювати на таких швидкостях і володіти такими можливостями, які значно перевершують ті, які потрібні людям або використовуються людьми. «Інтернет речей» змінить світ, можливо, більш ґрунтовно, ніж сучасний Інтернет, орієнтований на людей.

Концепція IoT була озвучена в 1999 році, а бурхливий розвиток почалося в 2008-2009 роках [13]. Автор і ідеолог концепції Інтернету речей - Кевін Ештон [12]. З 1999 Інтернет речей еволюціонував від ідеї радіочастотної ідентифікації пристроїв до нової концепції Всесвітньої мережі. Промислове обладнання, автомобілі, міський транспорт, носимі пристрої, «розумні будинки» - сьогодні все підключено до мережі інтернет.

У 1969 році ARPANET - прообраз Інтернету - об'єднував всього кілька десятків комп'ютерів [31].

У 1984 році, коли була заснована компанія Cisco, до Інтернету було підключено всього три тисячі пристроїв [44].

У 2008 році число приєднаних до мережі пристроїв вже дорівнювала кількості жителів Землі - 7 мільярдів [44].

У 2014 році число підключень людей, всіляких об'єктів, процесів і даних подвоїлося і вже становило 16,25 млрд [43].

На сьогодні через дротові і бездротові мережі до Інтернету підключено близько 50 мільярдів всіляких пристроїв, що генерують зеттабайт даних на базі хмарних технологій [29].

Але у цей показник не включені сенсори [44]:

в 2012 році по всьому світу було 2 млрд датчиків,

в 2013 році - 8 мільярдів,

в 2015- 24 мільярди.

Сьогодні Інтернет забезпечує з'єднання веб-сайтів і робочих станцій [53]. В даний час в ньому налічується величезна кількість сайтів - майже мільярд, що впливає на всі сторони нашого життя (рис. 1.1.3.1).



Рис. 1.1.3.1. Останнім часом відзначається різке збільшення числа пристроїв з виходом в мережу.

З лякаючою швидкістю - усього за кілька років - мобільна революція забезпечила стільниковими телефонами більшу частину населення світу. Маючи в своєму розпорядженні понад 5 мільярдів смартфонів, більшість людей вже має або незабаром отримає мобільний доступ до мережі [53].

Крім того, в світі налічується 50 млрд інтелектуальних пристроїв. «Інтернет речей» дозволить їм встановлювати між собою зв'язок і забезпечить створення в повному розумінні слова розподілених додатків з міжмашинним (machine-to-machine - M2M) зв'язком. «Інтернет речей» в 50 разів перевищить за обсягом традиційний Інтернет. Він забезпечить з'єднання в 10 разів більшої кількості пристроїв, ніж це дозволила свого часу зробити мобільна революція, і стане стимулом подальших перетворень нашого життя [53].

Ці пристрої вже є частиною нашого повсякденного життя. У новітніх автомобілях застосовується понад 100 процесорів. Інтелектуальні пристрої

проникають в промислові комплекси, лікарні, будинки, транспортні системи та інші сфери життя. Сьогодні ці системи пов'язані поки не сильно, але такий стан речей швидко змінюється [53].

Поступово ефект від IoT знижується [12]. У той же час підвищується потреба і підвищується ефект від застосування IoE. У чому закономірності і логіка змін?

У **M2M** головним об'єктом дії є пристрої.

У **IoT** поступово з'являється людина (наприклад, який замовляє піцу по телефону).

IoE розглядається як зв'язок пристроїв, людей, даних і процесів. Тобто, включає ще більше об'єктів, що створює нові можливості для бізнесу та світової економіки в цілому.

Перспектива шалена, оскільки, за оцінками Cisco, 99% фізичних об'єктів, які одного дня можуть стати об'єктами Інтернету всього – все ще не підключені до Мережі [12].

IoE може означати наступний етап після IoT, який настане, коли будуть створені зрозумілі широкому користувачеві механізми роботи з великою кількістю даних від різних пристроїв. Для цього IoT повинен набути широкого поширення практично у всіх сферах людської діяльності, включаючи сільське господарство, виробництво і комерцію, повсякденне життя. Тільки тоді нові види бізнесів, нові товари і нові розваги вже з самого моменту свого створення будуть відразу отримувати і цифрову складову, що принципово змінить сам стиль життя і стратегії поведінки людей [12].

Зараз M2M знаходиться в процесі поглинання з боку IoT, який є більш загальним концептом по відношенню до M2M. Так саме через якийсь час IoE поглине IoT [12].

1.1.4 Тенденції та перспективи еволюції

1.1.4.1 Причини та тенденції

«Інтернет змінив все, що нас оточує. Але ці зміни - ніщо в порівнянні з тим, що поки ще попереду [12]. Всеохоплюючий Інтернет викличе грандіозні зміни в усіх галузях економіки. Це означає нові перспективи, нові компанії, новий досвід, нові послуги, а також величезні перспективи перед людьми, компаніями та цілими країнами».

(Голова ради директорів, головний виконавчий директор компанії Cisco Джон Чемберс (JohnChambers))

Роб Ван Краненбург, вважає, що «IoT в кінцевому рахунку зробить» реальний світ "більше схожим на стан знаходження в інтернеті, коли ми через браузер дивимося на віртуальний світ [12]. Коли нинішнє покоління вимикає свої комп'ютери, вони озираються навколо і знаходять реальний світ брудним, заплутаним і незадовільним».

Швидкість поширення технологій IoT зростає з кожним днем [12]. Сьогодні кількість «розумних» речей в світі перевищує кількість смартфонів і ПК і вже сягає величини 25 млрд пристроїв. Кількість M2M-з'єднань значно менше (Рис. 1.1.3.2).

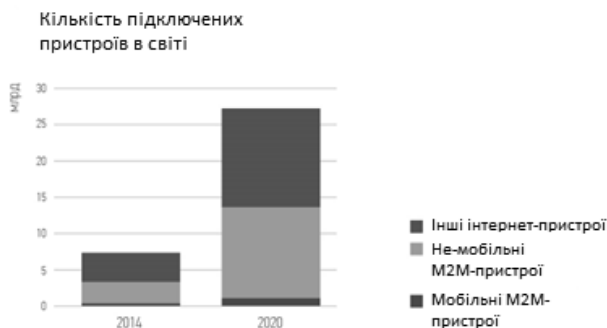


Рис. 1.1.3.2. Кількість підключених пристроїв.

Причини стримкого зростання IoT:

1. Суттєве зменшення вартості ключових технологій.

За останні 10 років комунікації стали дешевше майже на 90%. Знизилася ціна різних датчиків, тарифи підключення до інтернету, вартість зберігання і обробки даних.

2. Змінилися самі технології зберігання, маніпулювання, аналізу та застосування даних (Big Data).

3. Поява технології хмарних і розподілених систем.

4. Проникнення інтернету по всьому світу (включаючи як фіксований, так і, в першу чергу, мобільний інтернет).

Інтернет вже приносить реальну користь безлічі індивідуальних користувачів, компаній і цілих країн [72]. Він допоміг вдосконалити систему освіти за допомогою демократизації методів доступу до інформаційних ресурсів. Всемирная сеть стимулює економічне зростання шляхом електронної комерції і прискорює інноваційні процеси в бізнесі, розвиваючи спільну роботу. Що принесе нам наступне десятиліття розвитку Інтернету?

- ІоЕ вдихне життя в усе, що тільки можна собі уявити, підключаючи неживий світ до Інтернету. Використовуючи мережеві мікросенсори, предмети повсякденного користування підключаються до Інтернету і знаходять «інтелект» [72].

- Всеохоплюючий Інтернет буде сприяти розвитку економіки. За нашими оцінками, економічний потенціал Всеохоплюючої Інтернету на найближче 10-річчя в глобальному масштабі становить 19 трлн доларів. За 25 років, без сумніву, ця цифра істотно зросте, коли все більше підприємств і організацій стануть впроваджувати інтернет-технології в самих різних галузях [14].

- Всеохоплюючий Інтернет стане більш персональним і прогнозованим. У міру того, як у Всеосяжний Інтернеті будуть стикатися все більше підключених людей, об'єктів, даних і процесів, під його впливом отримає розвиток те, що фірма Gartner називає в своєму звіті «персональними світами». За допомогою мобільних технологій і хмарних обчислень Всеохоплюючий Інтернет об'єднає фізичний і віртуальний світи, створивши виключно персоналізовану і багато в чому прогнозоване підключене середовище [14].

- Всеохоплюючий Інтернет сприятиме схрещуванню технологій з гуманітарними науками. Всесвітня павутина була задумана як спосіб обміну комплексної інформацією, і майбутнє Інтернету додасть цій концепції нового потужного імпульсу. Розвиток Всеохоплюючої Інтернету буде визначатися все більшою зв'язністю, яка як ніколи наблизить до нас потрібну інформацію. Це матиме неабиякий вплив на наш світ. У найближчі 25 років Всеохоплюючий Інтернет допоможе масштабувати найцінніші ресурси і управляти ними, підвищуючи в результаті якість нашого життя. І тільки від нас залежить, як в реальності будуть виглядати ці перетворення [14].

За оцінками компанії Cisco, 99% фізичних об'єктів, які можуть стати частиною ІоЕ, до Мережі поки що не підключені [4]. У наступному десятилітті завдяки підключенню того, що ще не підключено, ІоЕ може забезпечити приватному сектору світової економіки прибуток, що обчислюється 14,4 трлн. доларів США . Йдеться про з'єднання між людьми (people-to-people, P2P), між машинами і людьми (machine-to-people, M2P) і між машинами (machine-to-machine, M2M).

В ході дослідження під назвою «IoE Value Index» («Індекс цінності Всеохоплюючої Інтернету») було опитано 7500 керівників компаній та ІТ-підрозділів з 12 країн. Виявилося, що учасники дослідження, перш за все, розглядають ІоЕ як критично важливий фактор, здатний допомогти їм випереджати хвилю інновацій, яка набирає силу [4].

Три ключові чинники, що сприяють прийняттю ІоЕ бізнесом [4].

- зростання темпів інновацій (36% опитаних);
- необхідність задовольнити зацікавленість споживачів в нових методах взаємодії (35%);
- необхідність автоматизувати бізнес-процеси (35%).

Опитані керівники бізнесу та ІТ-підрозділів переконані в тому, що ІоЕ допоможе їм досягти успіху в умовах експоненціального зростання кількості пристроїв, даних і технологій [4]. В якості трьох **головних технологічних факторів** впровадження ІоЕ респонденти назвали

- нові типи пристроїв (33%),
- зростаючий обсяг даних (32%),
- хмарні технології (31%).

46% опитаних в якості головної переваги ІоЕ назвали **операційну ефективність**, що пов'язується з більш ефективним використанням наявних активів і ланцюжків поставок, а також з підвищенням продуктивності праці [4].

Двома **іншими перевагами** Всеохоплюючої Інтернету учасники нашого дослідження вважають

- поліпшення послуг, що надаються замовникам (34% респондентів)
- більш ефективну спільну роботу всередині компанії в результаті впровадження інновацій та підвищення продуктивності праці (31%).

Більшу частину вищезгаданої прибутку в 14,4 трильйона доларів США забезпечать трансформація бізнес-процесів з урахуванням специфіки галузі та вмиле виволікання переваг, що надаються ІоЕ [4].

9,5 трильйона доларів принесуть специфічні галузеві нововведення (наприклад, впровадження в електроенергетиці технології smart grid або експлуатація підключених до Інтернету вантажних автомобілів).

Решта 4,9 трильйона доларів - такі міжгалузеві новації, як робота у віддаленому режимі і спільна робота в гнучких в налаштуванні офісах [4].

Щоб отримати максимальну користь з економіки ІоЕ, компанії повинні взяти на озброєння стратегію, яка передбачає:

- 1) інвестування в високоякісні технологічні інфраструктури та інструменти,
- 2) повсюдне впровадження методів бізнесу, що дозволяють всім співробітникам вносити особистий вклад у загальну справу,
- 3) розробку ефективних методів управління інформацією.

Вирішальну роль відіграють люди і процеси. Хоча аналіз даних чимало сприяє впровадженню ІоЕ, а дані придбали повсюдне поширення, самі по собі вони не можуть бути визначальним чинником [4].

Головним фактором, що визначає ефективність використання можливостей ІоЕ, є якість технологічної інфраструктури та відповідних інструментів. Без цього бізнес не зможе витягти той самий прибуток в 14,4 трильйона доларів.

Доступ до нових технологій на кшталт хмарних обчислень отримує все більше компаній, і це зрівнює сили на полі конкурентної боротьби. Тому тим більш важливо якомога ефективніше застосовувати інновації, щоб отримувати максимальну користь з ІоЕ [4]. Отже, необхідно:

1. Інвестувати в правильні технологічні інфраструктури.
2. Грамотно інтегрувати нові технології в існуюче ІТ-середовище.
3. Модернізувати процеси за допомогою нових технологій.

Компанії повинні інвестувати в ті області ІоЕ, які забезпечують підвищення конкурентоспроможності. Висновок дослідження «ІоЕ Value Index»: щоб отримати максимальну користь з ІоЕ, компаніям слід зосередитися на тих його можливостях, які найбільш значимі для їх галузей [4].

Для **виробничих підприємств** це - багатовимірний аналіз даних в реальному часі, спільна робота з інтеграцією відеозв'язку, віддалений контроль фізичних активів.

Для **енергетичних компаній** - інтеграція даних лічильників, можливість знаходити потрібних фахівців, попереджуючий аналіз.

Для компаній **роздрібно́ї торгівлі** - попереджуючий аналіз і візуалізація даних, BYOD (Bring Your Own Device), взаємодія з замовниками шляхом застосування мультимедійних технологій, мобільні платежі і віддалений моніторинг споживачів.

Керівники бізнесу очікують, що ІоЕ принесе зростання робочих місць і підвищення зарплат, а також зміцнення інформаційної безпеки [4].

47% керівників бізнесу вважають, що в їхніх компаніях ІоЕ сприятиме збільшенню заробітної плати (така точка зору особливо поширена на ринках, що формуються), а одна третина респондентів вважає, що це призведе до більшої зайнятості. Крім того, 50% опитаних сподіваються, що ІоЕ зробить корпоративну інформацію більш захищеною [4].

Шанси на найбільший абсолютний прибуток завдяки ІоЕ мають США, Китай і Німеччина. З точки зору ефективності найбільшу користь з ІоЕ витягнуть Німеччина (62,2%), Японія (57,7%), Франція (54,2%), Сполучені Штати (53,5%), Великобританія (53,4%) і Канада (53,1%).

Країни з ринком, що формується поки відстають, але демонструють розуміння потенціалу ІоЕ. Так, за шкалою від 1 до 10 керівники бізнесу та ІТ-відділів в Індії (8,2), Китаї (8,0), Бразилії (7,9), Мексиці (7,4) і Росії (7,2) налаштовані оптимістичніше своїх колег в розвинених країнах (там усереднений показник склав 6,7) в тому, що стосується здатності їх компаній скористатися можливостями, які відкриває ІоЕ.

У приватному секторі ІоЕ зрівнює конкурентоспроможність. Країни з ринком, що формується і компанії середнього розміру можуть витягти істотну вигоду з ІоЕ і в цьому плані досягти паритету, а то й вийти вперед, поряд з розвиненими країнами і більш великими фірмами. Такого роду демократизація в тому, що стосується ІТ і загальної користі ІоЕ, здатна в рекордні терміни змінити ландшафт конкурентної боротьби.

Крім того, ІоЕ є стартовий майданчик для проривних інновацій. Результат конкурентної боротьби буде залежати від того, наскільки успішно ті чи інші компанії зможуть об'єднати дані з людьми, процесами і фізичними об'єктами. Поки що лише окремі фірми цілком розуміють користь, яку вони можуть отримати з ІоЕ, і найбільшу вигоду здатні отримати самі прозорливі з них [4].

Конкуренція буде наростати в міру того, як у всьому світі ІоЕ буде грати роль загального урівнювача між компаніями будь-яких розмірів. У зв'язку з цим на ландшафті конкурентної боротьби практично в будь-якій галузі можуть відбутися разючі зміни. Фірми і компанії середнього розміру з країн з ринком, що формується мають відмінні і при цьому зростаючі шанси скласти гідну конкуренцію нинішнім лідерам. У тому, що стосується здатності отримати максимальну користь з ІоЕ, компанії з країн з ринком, що формується налаштовані більш оптимістично. За шкалою від 1 до 10, де 10 означає вищу ступінь впевненості, керівники бізнесу з країн з ринком, що формується зафіксували показник 7,8, тоді як їхні колеги в розвинених країнах - 6,7. Такі країни, як Бразилія, Китай і Індія, в останні роки збільшували інвестиції в інформаційні технології набагато швидше, ніж це робилося в світі в цілому. Крім того, такі фактори впровадження ІоЕ, як хмарні послуги і мобільність, дали змогу фірмам з країн з ринком, що формується швидко наздогнати ті компанії в розвинених країнах, які не подбали про те, щоб йти в ногу з новаціями [4].

1.1.4.2 Аналіз розбіжностей оцінок

При безперечній тенденції до зростання ІоТ, ІоЕ конкретні чисельні оцінки від різних аналітичних компаній істотно відрізняються. Розкид в оцінках перспектив зростання ІоТ знижує ступінь довіри до даних [37]. Наприклад, згідно з публікацією IDC витрати на ринок Інтернету речей (ІоТ) в 2014 році склали \$ 655,8 млрд і виростуть до \$ 1,7 трлн в 2020 р при рівні CAGR 16,9%.

За даними Harbor Research, в 2014 р ІоТ-ринок склав \$ 180 млрд, а до 2020 року його обсяг перевищить \$ 1 трлн.

За даними Machina Research, розмір ІоТ-ринку в 2014 році склав \$ 900 млрд, і виросте він до \$ 4,3 трлн до кінця 2024 року.

Згідно з аналізом Strategy Analytics сукупний ринок всіх елементів ІоТ-ринку складе \$ 150 млрд в 2016 році і виросте до \$ 550 млрд 2025 році.

Крім багаторазової відмінності в оцінках, звертає на себе увагу і те, що ІоТ-ринок становить досить велику частку від сукупних витрат на ІКТ [37]. Як стикаються настільки оптимістичні прогнози зростання ринку ІоТ при помірних прогнозах ринку ІКТ?

Проаналізуємо причини розкиду в оцінках, порівняємо підходи різних аналітиків.

У публікації ІоТ Analytics наводиться порівняльний аналіз прогнозів різних компаній щодо кількості підключених пристроїв в світі до 2020 р. Відповідно до даної публікації думки явно розділилися:

- Global Insight прогнозує, що таких пристроїв буде 18 млрд;
- ABI Research і IDC - близько 28 млрд,
- Cisco і Ericsson вважали, що підключених пристроїв буде близько 50 млрд.

Зазначені аналітики не тільки дають різні прогнози на далеку перспективу, але і по-різному оцінюють дані за 2014 р. Наприклад,

- ABI Research вказує число - 6 млрд,
- Cisco - 14 млрд.

У чому причина таких різних оцінок? Можливо, частина різночитань виникає з не зовсім коректного використання термінів, що мають відношення до IoT. Наприклад, в деяких публікаціях IoT-рішення згадуються як IoT / M2M або терміни M2M і IoT взагалі використовуються як синоніми. Часом терміни «промисловий Інтернет», «Інтернет людей», «Інтернет речей», «Web речей», «Інтернет всього» використовуються без уточнення меж означених понять.

Вплив сегментації ринку IoT, IoE на узгодженість прогнозів

Вище ми розглянули відмінності в тлумаченні терміну IoT і відзначили [37], що дані різночитання можуть призводити до різних оцінок щодо обсягу підключених IoT-пристроїв. Коли ж ми переходимо від аналізу кількості поставок підключених речей до опису ринку IoT-рішень - у аналітиків додається додаткова "ступінь свободи" в тому, як описувати ринок. Перш за все, слід нагадати, що аналітичні компанії використовують два типи сегментування ринків.

При першому підході, всі сегменти ринку ділять **за певною функціональністю** так, що всі ці сегменти складають 100%. Наприклад, ринки апаратного забезпечення, програмного забезпечення і IT-послуг разом складають 100% ринку IT. Коли ж ми говоримо, наприклад, про ринки хмарних послуг, ринки великих даних, ринки IoT - як правило, ми говоримо про ринки, які взаємоперетинаються. Тобто одні й ті ж сервери, програми, IT-послуги та телекомунікаційні-послуги можуть бути підраховані як в рамках ринку великих даних, так і в рамках ринку Інтернету речей або хмарних послуг. Тому описуючи ринок IoT-рішень, кожне з аналітичних агентств, в принципі, може сама вирішувати, від яких базових ринків і яку частку слід включити в ринок IoT-рішень. І за таких же оцінок загального обсягу ІКТ-ринку розподіл його на взаємоперетинаючіся частини може бути дуже різним. В якості підтвердження цієї тези розглянемо, як визначають межі ринку IoT різні аналітичні компанії і які сегменти в нього включають [37].

Згідно з даними **IDC** витрати на світовій IoT-ринку в 2014 році склали \$ 655,8 млрд і виростуть до \$ 1,7 трлн. Вартість підключення Інтернету речей та IT-послуг будуть складати основну частину ринку IoT і разом вони складуть понад 2/3 ринку IoT. Більшу частину ринку (31,8%) складуть пристрої (модулі / датчики).

Визначення IDC: IoT - мережа мереж однозначно ідентифікованих підключених пристроїв (або "речей"), які обмінюються даними на базі IP-з'єднань без участі людини. Автономне підключення є ключовим атрибутом у визначенні, і не враховує в якості IoT-пристроїв смартфони, планшетні комп'ютери і ПК.

Які ж горизонтальні сегменти IoT-ринку виділяє IDC? Згідно з публікаціями компанії, до IoT-ринку, крім апаратного і програмного забезпечення, відносяться і IT-послуги, і телеком-послуги. Тобто ринок IoT знаходиться на стику IT- та телекомунікаційного ринку.

Телекомунікаційні послуги - це послуги провайдерів дротового і бездротового зв'язку для підключення IoT-пристроїв. До IT-послуг відносяться консалтинг, послуги з установки апаратного і програмного забезпечення, системна інтеграція, послуги підтримки і супроводу, навчання, а також послуги аутсорсингу інформаційних систем, включаючи хмарні послуги.

До **апаратного забезпечення** в частині IoT-ринку IDC відносить різного роду підключені IoT-пристрої:

- контролери,
- датчики,
- RFID-мітки,
- інші дротові або бездротові IoT-пристрої.

При цьому, якщо пристрій, "рiч" або датчик має IP-адресу і автономно підключений до мережі в якийсь момент протягом свого життєвого циклу, він підпадає під визначення IoT.

До сегменту апаратне забезпечення відносяться і інші елементи ІКТ-інфраструктури:

- сервери,
- системи зберігання даних,
- мережеве обладнання, включаючи комутатори і маршрутизатори,
- апаратні системи забезпечення безпеки.

Види **ПЗ** на IoT-ринку:

- аналітичне ПЗ використовує дані, зібрані за допомогою підключених речей, для переведення їх в інформацію для прийняття управлінських рішень;
- прикладне ПЗ надає інтерфейс користувача, працює з аналітичними додатками, забезпечує функції різного роду галузевої специфіки;
- IoT-платформи інтегрують ряд функцій і зазвичай включають: моніторинг і управління підключеними пристроями; збір, перетворення і управління IoT-даними, а також розробку IoT-додатків; частина функціональності IoT-платформ може перетинатися з вищепереліченими категоріями ПЗ;
- ПЗ безпеки.

Компанія **Machina Research** використовує інше сегментування ринку (Рис. 1.1.4.1). За даними Machina Research, IoT-ринок виросте до \$ 4,3 трлн до кінця 2024 року, що відображає значне зростання в порівнянні з \$ 900 млрд в 2014 році.

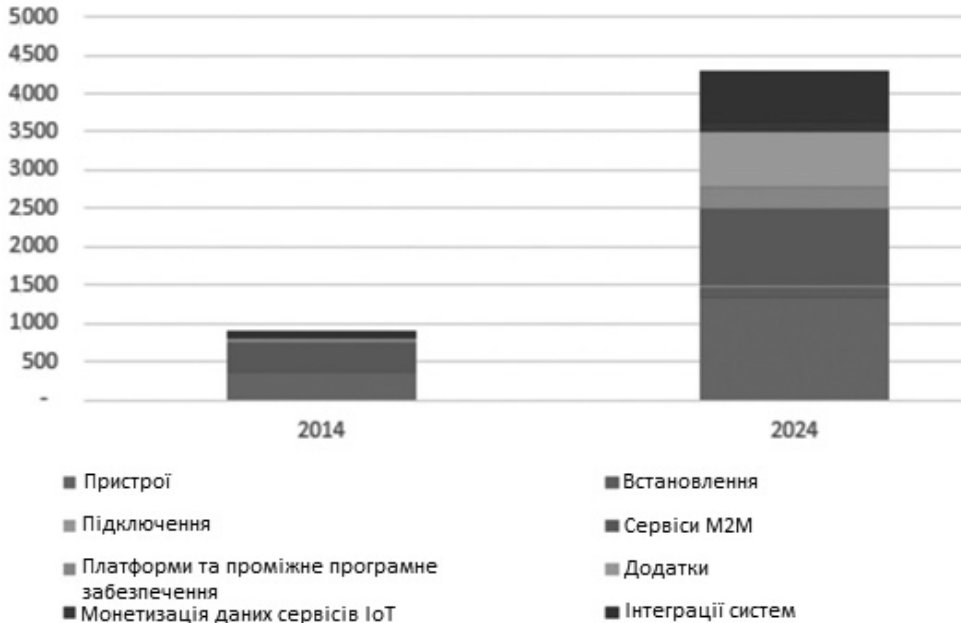


Рис. 1.1.4.1. Прогнози світового IoT ринку, \$ млрд (джерело Machina Research, 2015).

Як відзначають аналітики Machina Research, основні витрати на IoT включають додатки, професійні послуги, M2M -сервіси та пристрої. Як видно з Рис. 1.1.4.1, компанія використовує відмінну від IDC сегментацію ринку:

- Device - пристрої;
- Connectivity - витрати на зв'язок;
- Platforms and Middleware - платформенне ПЗ і ПЗ проміжного шару;
- Data Monetization and IoT Services - засоби монетизації і IoT-сервіси;
- Installation - послуги з встановлення;
- M2M Services - M2M-сервіси;
- Applications - додатки;
- Systems Integration and Strategy - послуги системної інтеграції та розробки стратегії.

Компанії **Harbor Research** пропонує свій варіант структурування ринку IoT (Табл. 1.1.4.1)

Таблиця 1.1.4.1.

Прогноз розвитку ринку IoT, \$ млн за даними компанії Harbor Research

	2014	Доля сегментів у 2014	2020	Доля сегментів у 2020	Ріст (2020/2014), %
Керовані сервіси (дані та аналітика, системне ПЗ, мобільні та хмарні застосунки, інші сервіси)	86919	48,2%	571845	57,0%	558%
Апаратне забезпечення (провідні та безпроводні модулі, модулі, що під'єднуються до мережі)	15186	9,0%	43782	4,4%	170%
Мережеві сервіси	77273	42,8%	387453	38,6%	401%
Разом	18037	100,0%	100308	100,0%	456%

Тут категорія апаратного забезпечення в 2014 році становить лише 9%, і її частка буде знижуватися з часом. По всій видимості, в даній методиці в апаратну інфраструктуру не включається та її частина, яка надається в режимі керованих сервісів, а саме як IaaS.

До сих пір ми дивилися на горизонтальну структуру ринку IoT-рішень. Аналіз публікацій на тему вертикальної структури ринку показує, що дані різних аналітиків також спираються на різну таксономію. Як приклад в табл. 1.1.4.2 і 1.1.4.3 показують прогнози Harbor Research і McKinsey.

Таблиця 1.1.4.2.

Вертикальна структура ринку IoT за даними Harbor Research

	2014, \$ млрд	2017, \$ млрд	2020, \$ млрд
Дім/користувачі	79,4	180,3	397,8
Транспорт/мобільність	10,4	29,4	76,1
Здоров'я/тіло	6,2	18,5	48,7
Будівлі/інфраструктура	25,0	77,08	210,0
Міста/промисловість	59,2	129,8	270,0

Як видно з таблиці 1.1.4.3, прогноз McKinsey єдиний з вищезазначених дається з розкидом, причому по деяких категоріях песимістична і оптимістична оцінка відрізняються в рази. Таким чином, оцінки ринку IoT, надані різними аналітичними компаніями, носять неузгоджений характер. Розмір ринку може відрізнитися в кілька разів. Структурування ринку виконується за внутрішніми методиками, що ускладнює порівняння даних від різних аналітиків. Публікації в пресі, які стосуються розміру ринку IoT, як правило, не приводять методики оцінки ринку і його структуру, що погіршує плутанину. З огляду на важливість

кількісного опису IoT-ринку, слід привернути увагу IoT-спільноти до проблеми. Вендорам, які є клієнтами аналітичних агентств, слід в більшій мірі впливати на вироблення таксономії, на ступінь її узгодженості між різними провайдерами консалтингових послуг. Це дозволить внести більшу ясність в оцінки ринку, ролі вендорів, які зможуть чіткіше визначити свій внесок і своє місце на ринку.

Таблиця 1.1.4.3.

Вертикальна структура ринку IoT 2025 р даними McKinse

		\$ млрд (песимі- стична оцінка)	\$ млрд (опимі- стична оцінка)
Моніторинг фізичного стану людини	Моніторинг фізичного стану, моніторинг та керування хворобами, покращення стану здоров'я громадян	170	1590
Розумний будинок	ЖКП, керування енерговитратами, створення засобів розумного будинку	200	350
Торгівля	Оптимізація компонування, автоматизація контролю та оплати товарів, персоналізований маркетинг в середині магазину, автоматизація контролю товарних запасів	410	1160
Офіси	Моніторинг співробітників. Використання доповненої реальності для навчання співробітників, створення систем безпеки в офісі	70	150
Фабрики/ заводи	Автоматизація операційної діяльності, прогнозований технічний огляд, автоматизації інвентаризації, автоматизація техніки безпеки	1210	3700
Виробничі ділянки	Автоматизація операційної діяльності, науково-дослідна діяльність з використанням IoT, автоматизація підтримки та ремонту обладнання	160	930
Транспорт	Технічне обслуговування на основі моніторингу стану автомобіля, індивідуальна страховка на основі моніторингу водіння	210	740
Розумні міста	Суспільна безпека та здоров'я, контроль дорожньо-транспортного навантаження, керування ресурсами	930	1660
Віддалені об'єкти	Моніторинг віддаленими об'єктами	560	850

1.1.4.3 Цінність ІоЕ та Інтернет. Аккумуляованна цінність

Відповідно до закону Меткалфа, цінність мережі пропорційна квадрату кількості її вузлів. Чим більше користувачів і пристроїв підключено до ІоЕ, тим вона цінніша. Що станеться, якщо люди, предмети і дані, - почнуть додавати свою власну "цінність" до цієї формули [15]? На думку інтернет-видання <http://techcrunch.com/> [114], мережевий ефект буде посилюватися завдяки "аккумуляованій цінності" (stored value), виробленої користувачами і дасть компаніям можливість утримувати клієнтів і нарощувати свою частку ринку [15]. По суті, мережевий ефект збільшує "цінність" мережі в геометричній прогресії, створюючи небувалі можливості [15]. ІоЕ дасть нам більш ясну картину світу і розширить наші горизонти, дозволивши нам приймати більш зважені рішення, засновані на більш точному сприйнятті дійсності.

Але, щоб не впасти в ейфорію, важливо зважити всі ризики і наслідки, пов'язані з Інтернетом, чия міць зростає експоненціально. Більшість представників активної ділової діяльності найбільше турбує захист особистих даних та інформаційна безпека. З філософської точки зору, Інтернет - всього лише ще один предмет. Сам по собі він не може бути ні хорошим, ні поганим, і в кінцевому рахунку все залежить від того, що ми робимо з Інтернетом. З технологічної точки зору, ІоЕ надасть людям ширші знання і можливість краще розуміти суть речей - приблизно так, як за допомогою Інтернету споживачі тепер отримують більш детальну інформацію про ринкові ціни і можуть приймати більш обґрунтовані рішення при покупках. Ніщо не заважає прийняти правильні рішення, що дозволяють ІоЕ повністю розкрити свій потенціал в тому, що стосується підтримки інновацій в бізнесі, підвищення прозорості державного управління або поліпшення якості життя. Але для цього важливо активно формувати структуру ІоЕ.

1.1.5 Порівняння з суміжними технологіями

Межі між інтернетом і Інтернетом речей дуже умовні [12]. Наприклад, телефон з мобільним доступом в Інтернет сам по собі не є об'єктом ІоТ. Але якщо на телефоні встановлено додаток, за допомогою якого користувач оплачує паркування або замовляє піцу, яку доставив автоматичний дрон - то в такому випадку ми вже говоримо про Інтернет речей. Інакше кажучи, будь-які сценарії використання мереж зв'язку не для спілкування по телефону - це і є те нове, що приносить ІоТ в наше життя.

Також, термін Інтернет речей часто використовується, а іноді і ототожнюється з технологією М2М [12]. М2М застосовується ще з 80-х років, коли американська компанія Qualcomm почала розробку рішень, що дозволяють відслідковувати комерційний транспорт за допомогою бездротових засобів зв'язку. Останні десять років концепція М2М вважалася одним з головних драйверів зростання як на транспорті, так і в банківському секторі, промисловості, енергетиці та Рітейл. До тих пір, поки її не замінив Інтернет

речей. Одні фахівці інтерпретують IoT як самостійну нову науково-технологічну теорію, інші впевнені в наступності дотеперішніх систем (M2M). Спочатку M2M і Інтернет речей виникли як наукові концепти, але незабаром стали позначати цілі напрямки на ринку зв'язку. З науково-технологічної точки зору вони являють собою концепції, що складаються з безлічі технологій, які змінювалися з плином часу.

M2M. Відмітна риса M2M - взаємодія об'єктів, оснащених обчислювальним і комунікаційним модулями, з обмеженою участю людини [12]. M2M пристрої зв'язуються один з одним, збирають дані і передають їх по мережах мобільного та дротового зв'язку. У мобільних M2M-пристроях використовуються SIM-карти з особливим тарифом, які передають дані через Інтернет або за допомогою SMS, при цьому збирається обмежена кількість даних. Наприклад, банкомати, автоматизовані системи обліку електроенергії, платіжні термінали та системи ГЛОНАСС, система моніторингу вантажного транспорту.

IoT розглядає не тільки пристрої с обчислювальними і комунікаційними компонентами, але й речі, оснащені спеціальними засобами ідентифікації (радіочастотними мітками, візуальними штрих-кодами, QR-кодами і ін.). Наприклад, система розпізнавання реєстраційних знаків автомобіля, що дозволяє визначати номери машин, що потрапили в поле зору камер відеоспостереження. В такому фокусі автомобіль також є об'єктом Інтернету речей, навіть без додаткового комунікаційного модуля. IoT - пристрої обмінюються даними за допомогою IP-мереж зв'язку з різними протоколами [12]. При цьому IoT - поступово, але постійно стандартизується. Дані збираються від величезної кількості датчиків - в рази або на порядки більше, ніж у M2M-пристроїв. Крім того, IoT має на увазі не тільки збір інформації, але й реагування на неї в автоматичному режимі, а також глибоку аналітику даних. Наприклад, M2M-пристрої можуть попередити користувача про несправності датчика води, а за допомогою IoT можна встановити причини збою, знайти попередні несправності і видати прогноз про те, як уникнути поломки в майбутньому.

Рівні застосування технологій [12].

- M2M - це конкретні локальні сценарії використання телекомунікаційних систем (з'єднання двох комп'ютерів).
- Інтернет речей - це глобальна мережа. (Інтернет, сервери, інфраструктура)

Ефект від технологічних змін IoT незрівнянно більше, ніж ефект від M2M [12]. Зміни відбуваються прямо зараз в самих різних галузях бізнесу - і, в першу чергу, в телекомунікаціях. Доходи операторів зв'язку від традиційних послуг (голосових викликів і смс-повідомлень) поступово знижуються. Директор з питань глобалізації компанії Cisco Вім Елфрінк в своїх інтерв'ю заявляє, що вже сьогодні оператори повинні думати про те, які додаткові сервіси вони можуть запропонувати клієнтам і як заробити в таких сферах, як Інтернет речей і M2M.

У 2015 році 468 операторів в 190 країнах пропонували M2M-послуги. Тобто три з десяти операторів світу використовують технології M2M для нових бізнес-ініціатив, з них 2/3 - це європейські оператори.

M2M [93] (машино-машинна взаємодія, англ. *Machine-to-Machine, Mobile-to-Machine, Machine-to-Mobile*) — загальна назва технології, яка дозволяє просто, надійно та вигідно забезпечити передачу даних між різними пристроями. Ринок рішень на базі мобільного M2M існує понад 20 років. За оцінками Harbor Research на кінець 2010 року у світі налічувалося кілька десятків мільярдів так званих «розумних» пристроїв (smart devices) — електронних машин, здатних взаємодіяти між собою.

Види M2M

1. *Стационарний* M2M забезпечує використання різноманітних рішень: управління технологічними процесами, охоронний моніторинг, платіжні термінали, лічильники, автомати роздрібною торгівлі;

2. *Мобільний* M2M дозволяє управляти автопарком та передбачає велику кількість автомобільних додатків. Авторинок дає безліч можливостей для використання M2M як бортового пристрою, здатного здійснювати моніторинг, діагностику, навігацію, позиціонування, охорону та власне мобільний зв'язок (Рис. 1.1.5.1).



Рис. 1.1.5.1. Кількість з'єднань.

Способи реалізації M2M-рішень [93]

- **DTMF** — дозволяє посилати кодовані повідомлення, використовуючи звичайні голосові канали. Це може ефективно використовуватися для потреб охорони (DTMF-формати передачі інформації, наприклад, Contact ID), найпростішого дистанційного управління, для контролю за персональною інформацією, наприклад, в голосовій пошті. Зазвичай автовідповідач запитує пароль для допуску в

меню. Це основна M2M-можливість, яка використовується в більшості мереж.

- **SMS** — сервіс коротких повідомлень. Там, де оператори стільникового зв'язку надають цей сервіс, для користувачів M2M-технологій відкриваються величезні можливості. Головна перевага SMS — це те, що повідомлення може бути передано за мілісекунди та містити велику кількість інформації. SMS відкриває можливості для більшості фіксованих і мобільних M2M-додатків, таких як автомати роздрібною торгівлі, платіжні термінали, комунальне господарство та ін.

- **DATA** і «Always on» DATA — модемний зв'язок. Використання комутованих каналів для передачі інформації обмежена низькою, порівняно з виділеними каналами «Always on» DATA, швидкістю передачі інформації. Системи передачі пакетної інформації GPRS або CDPD вже доступні і дозволяють здійснювати інтерактивні комунікації в реальному часі. «Always on» DATA є ключовим орієнтиром у розвитку M2M. З розвитком GPRS і CDPD стали можливими телемедицина та розваги.

- **WAP-IP** призначені для рішень «людина-машина» (man-to-machine), і починають ставати буденними для електронних замовлень, покупок і платежів, об'єднуючи в собі інформаційний сервіс та додатки з управління і моніторингу. Для автоматів роздрібною торгівлі мережа WAP-IP могла б підготувати платформу для розвитку охоронних та антивандальних систем.

- **3G.** Третє покоління стільникового зв'язку забезпечує необмежені можливості в телекомунікаційному середовищі, коли всі машини та обладнання для побудови мереж відповідає одному і тому ж протоколу.

Застосування M2M [93]

- **Системи доступу.** Дають можливість за допомогою стільникового телефону заходити в певні приміщення, відкривати електронні замки, двері та ін. Такі дії здійснюються за допомогою звичайного телефонного дзвінка або після відправлення певного коду.

- **Системи охорони приміщень.** M2M-рішення дозволяють здійснювати дистанційну бездротову охорону приміщень. Дає можливість користувачам систем самостійно дистанційно ставити-знімати з охорони такі приміщення. Також можлива дистанційна передача відео чи зображень за допомогою GSM.

- **Системи безпеки.** Бездротові системи безпеки (пожежна, аварійна, персональна та ін.). M2M дозволяє незалежно і дистанційно відслідковувати стан об'єктів та, за необхідності, автономно посилати сигнали тривоги або інформацію про стан об'єктів.

- **Дистанційний контроль та управління «домашнім» обладнанням.** Дає можливість за допомогою мобільного телефону дистанційно здійснювати контроль домашньої техніки, підтримувати

певні умови в приміщеннях, дистанційно керувати домашньою технікою (обігрівачі, кондиціонери, сауни та ін.).

- **Автотранспорт і моніторинг рухливих об'єктів.** Застосування систем M2M на транспорті практично необмежена. Це і автоматичні повідомлення про аварії в службу порятунку, з повідомленням точних координат місцезнаходження; системи проти викрадення авто; надання менеджера транспортної компанії інформації про рух автомобіля, його технічний стан, охорону вантажу.

- Автомати з продажу, кавові автомати, платіжні термінали, банківські та касові автомати та ін. M2M-обладнання дозволяє дистанційно контролювати стан, дистанційно охороняти **вендінгове обладнання**, збирати інформацію про стан роботи, здійснювати контроль працездатності обладнання.

- **Охорона здоров'я.** M2M дозволяє проводити віддалену діагностику пацієнтів; надавати допомогу людям з обмеженими можливостями; автоматизувати екстрені виклики.

1.1.6 Ризики та питання безпеки

1.1.6.1 Ризики

Чим більше об'єктів потрапляють в цифровий світ, тим тонше стає грань між перевагами розумних технологій і можливими проблемами в особистому житті [13]. Уже зараз ваш оператор мобільного зв'язку або ваш банк можуть знати про ваше життя набагато більше, ніж ваша дружина або чоловік.

Актуальними стають такі питання

- з огляду на вплив, який Інтернет вже здійснює на бізнес, як на цей бізнес може вплинути експоненціальна хвиля підключень нових пристроїв [10]?

- Які потенційні переваги і ризики, створювані середовищем ІоЕ для моєї компанії (або державної установи) [10]?

- Як зміняться питання управління і розподілу відповідальності в світі ІоЕ [10]?

Аналітики Machina Research прогнозують, що сьогодні кількість M2M-підключень на транспорті зростає величезними темпами і незабаром роумінгові тарифи стануть справжньою проблемою для операторів зв'язку [12]. Наприклад, вантажівки на міжнародних трасах, забезпечені спеціальними позначками, навантажують мережу незалежно від кількості переданих даних. На відміну від людей, які контролюють свої витрати в роумінгу, машини просто не вміють цього робити. Фахівці вважають, що операторам вже зараз варто оптимізувати M2M-сервіси і починати діалог з регуляторами, для того, щоб роумінгові тарифи поступово знижувалися.

Ще одна небезпека полягає в тому, що чим більше пристроїв буде підключатися до Мережі, тим більше будуть обсяги оброблюваних даних. При

такому навантаженні мережі потребуватимуть надійних методів обробки і зберігання великих даних.

Традиційний підхід до побудови та експлуатації мережевих інфраструктур для ери ІоЕ вже не годиться - потрібні нові архітектури і рішення [31]. Зокрема, ця проблема може бути вирішена при переході до програмно-конфігурованим мереж (ПКМ).

Чітка структура і зрозумілі правила розвитку і поширення ІоЕ дозволять максимально використовувати можливості Всеохоплюючого Інтернету і розкрити його «мирний» потенціал [10]. Зрештою, інтернет - всього лише ще один предмет, як стверджує головний футуролог Cisco Дейв Еванс. ІоЕ не може бути добрим чи поганим - все залежить від того, з якою метою ми маємо намір його використовувати.

Негативні наслідки від всепроникного і всеконтролюючого Глобального Інтернету виникають через невідповідність між рівнями розвитку технологій, які забезпечують інформаційну безпеку, і самих пристроїв [10]. Це найбільш гостра проблема, що викликає у користувачів занепокоєння щодо збереження їх особистих даних. Зрозуміло, що вихід один - зосередитися на боротьбі з кіберзлочинністю та створенні ефективних рішень, які захистять від несанкціонованого доступу і фальсифікації даних. У ІТ-компаній є всі необхідні технологічні ресурси для цього. Важливо, щоб і держава підтримувала нові технології і розробки в галузі безпеки і брала участь у формуванні, наприклад, правових механізмів захисту.

1.1.6.2 Ризики зникнення підприємств, які ігнорують прогрес

В результаті цифрової революції в найближчі п'ять років в 12 галузях зникне 40% компаній [50]. Для компаній важливо не упустити момент зміни технологій [44]. Що стосується ТОП-500 світових корпорацій, то тут в найближчі десять років очікується зникнення 35% компаній, які пропустять або не впишуться в цей цифровий перехід. В результаті цифрової революції процеси зникнення компаній і перепрофілювання ринків будуть відбуватися з небувалою швидкістю [50]. До такого висновку прийшли автори звіту, опублікованого Глобальним центром цифрових перетворень бізнесу (Global Center for Digital Business Transformation, DBT Center) - спільного проекту компанії Cisco і розташованого в Лозанні Міжнародного інституту управлінського розвитку (International Institute of Management Development, IMD). Звіт під назвою «Цифровий вир, або Як цифрова революція реформує промисловість» (DigitalVortex: How Digital Disruption is Redefining Industries), складений за результатами опитування керівників 941 компанії з 12 галузей в 13 країнах (Австралія, Бразилія, Великобританія, Німеччина, Індія, Італія, Канада, Китай, Мексика, Росія, США, Франція, Японія). Термін «цифровий вир» позначає рушійні сили, створювані в результаті цифровізації у всіх галузях, а також дію цих сил, в результаті чого компанії з неминучістю і все глибше втягуються в процес цифровізації.

Більшість опитаних керівників визнають позитивний вплив цифровізації на бізнес і суспільство. Однак 43% або не бачать ризиків, пов'язаних з цифровою революцією, або ставляться до них недостатньо серйозно.

1.1.6.3 Інформаційна безпека

В першу чергу, необхідно забезпечити інформаційну безпеку [44]. Конфіденційність і захист даних стають викликом для всіх. Генеральний директор компанії Target, одного з найбільших ритейлерів у компанії США, втратив свою посаду через вкрадення даних. Причому, хакери змогли до них добратися через систему кондиціонування, управління якою здійснювалося через загальну інформаційну систему компанії.

Як забезпечити безпеку і захист особистих даних в умовах все більш підключеного світу [30]? Хакерські атаки стали повсякденним явищем, від них не застрахована жодна організація. Якщо хакерам так легко вкрати інформацію і зловжити нею в сучасному підключеному світі, виникає природне побоювання, навіть страх, з приводу майбутнього, коли люди, процеси, дані і пристрої стануть невід'ємною частиною ІоЕ.

На думку Алекса Пентланда професора Массачусетського технологічного інституту (провідний експерт в області мережевих підключень) [30]:

- особиста інформація буде закріплена за вами в законодавчому порядку. ... Разом з тим ми спостерігаємо зрушення, здатні перевести обговорення питань інформаційної безпеки в зовсім іншу площину: в майбутньому ви зможете стати повноправним власником своїх особистих даних. Власником цих даних будете саме ви, а не ті, хто їх збирає (Amazon.com, Facebook, Google і т.д.). Наприклад, якщо сьогодні ви викладаєте фотографію в "Фейсбук", її власником вважається "Фейсбук", а не ви. В даний час безпека особистих даних регулюється правилами, яким користувачі підкоряється, навіть не знаючи їх. Коли ваші права на особисті дані будуть закріплені в законодавчому порядку, ви зможете краще контролювати процеси їх збору та використання. Ідея про те, що саме користувач повинен володіти своїми особистими даними, поступово стає все популярнішою, зокрема, у Великобританії, серед учасників Всесвітнього економічного форуму і в Європейському Союзі, де захист особистих даних проголошений однією з фундаментальних прав людини.

- Особиста інформація стає валютою. Коли питання володіння особистими даними остаточно проясниться, сирі дані, що становлять ліву частку сучасного інтернет-трафіку, будуть доповнені інтелектуальною інформацією. До звичайних пакетів мережевого трафіку будуть прикріплюватися метадані, що включають відомості про те, де і ким дані були створені і куди вони передаються. В результаті дані стануть подобою валюти, що лежить на вашому онлайн-банківському рахунку. За нею можна буде стежити, її можна буде передавати. Крім того, особисту інформацію можна буде перевіряти, щоб, в разі необхідності, забезпечити дотримання корпоративних правил і законодавчих вимог. І, нарешті, право на володіння

особистою інформацією буде зафіксовано в законодавчих та юридичних системах і буде захищатися ними з такою ж ретельністю, як права на інтелектуальну власність.

- Для захисту ваших даних будуть використовуватися персональні сховища. Вони допоможуть не тільки переглядати особисті дані і управляти ними, а й захищати інформацію, якою ви володієте. Крім того, сховища допоможуть вам отримувати від інформації додаткову вигоду. Наприклад, ви зможете стягувати з компаній невелику плату за надання даних про свої уподобання, при цьому дані будуть як і раніше залишатися в добре захищеному сховищі. Таке завдання буде вирішуватися за допомогою аналітичних засобів, що дозволяють робити висновки на підставі вашої особистої інформації. Якщо та чи інша компанія захоче дізнатися про вас щось корисне для себе, вона буде направляти свої питання в сховище даних, яке з вашого дозволу видаватиме відповіді, не розкриваючи самих даних. За твердженням аналітичної компанії Ctrl-Shift, на ринку вже діє кілька сховищ такого типу.

На додаток до вищесказаного зауважимо, що великий вплив на підвищення рівня безпеки Всеохоплюючої Інтернету буде здійснювати сама мережа. Сьогодні велика частина інтелектуальних мережевих функцій надається в централізованому порядку і часто знаходиться на "периферії" Інтернету. По мірі розвитку мереж в епоху ІоЕ з'явиться можливість приймати обгрунтовані рішення миттєво, в той момент, коли дані будуть потрапляти в мережу і передаватися по ній. У цьому випадку доставка послуг стане більш розподіленою, розташованою в безпосередній близькості до користувача. В результаті можна буде миттєво розпізнавати, локалізувати і усувати загрози безпеки до того, як вони встигнуть завдати реальної шкоди [30].

Сьогодні ніхто не може точно спрогнозувати, як саме буде розвиватися Всеохоплюючий Інтернет, але я вважаю, що, як це не раз було в минулому, спільні дії користувачів, організацій, компаній і урядів допоможуть успішно вирішувати проблеми інформаційної безпеки і захисту особистих даних по мірі їх виникнення. Технологія стане невід'ємною частиною цих рішень. У будь-якому випадку кінцевий результат залежить від наших власних зусиль. Кожен з нас повинен зробити все можливе для того, щоб по мірі розвитку Всеохоплюючого Інтернету всесвітня мережа залишалася потужним фактором, що змінює наше життя на краще [30].

1.1.6.4 Програмно-конфігуруєма мережа, як засіб зменшення проблем і ризиків

Програмно-конфігуруєма мережа - це мережа, у якій управління мережею відокремлено від мережевих пристроїв, що забезпечують просування інформації в мережі, і програмується безпосередньо [31]. Таке переміщення управління, раніше жорстко прив'язане до конкретних мережевих пристроїв, в доступний обчислювальний пристрій дозволяє додаткам і мережевим сервісам абстрагуватися від нижчого фізичного обладнання та розглядати мережу як

якусь логічну і віртуальну сутність [31]. На рисунку 1.1.6.1 зображена логічна структура архітектури ПКМ, що реалізується на базі інтерфейсу OpenFlow.

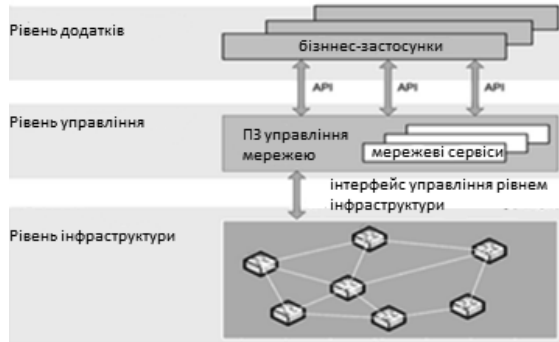


Рис. 1.1.6.1. Логічна структура архітектури ПКМ.

Архітектура ПКМ включає три рівня, доступ до яких здійснюється через відкриті API («інтерфейс програмування додатків» - API) [31].

- **Рівень додатків**, що складається з бізнес-додатків кінцевих користувачів. Його взаємодія з нижнім рівнем здійснюється через API північного кордону (northbound API). Використовуючи зазначені API, можна реалізувати весь набір мережевих сервісів: маршрутизацію, ширококомвні розсилання, безпеку, списки доступу, управління смугою пропускання, інженірінг трафіку, якість обслуговування, оптимізацію роботи процесора і пам'яті, оптимізацію використовуваної енергії, всі форми управління мережевими політиками, спеціальні настройки для бізнесу.

- **Рівень управління** здійснює логічно централізоване управління, яке забезпечує просування трафіку на інфраструктурному рівні за допомогою відкритого інтерфейсу OpenFlow, званого також інтерфейсом південного кордону (southbound interface). Рівень управління постійно «бачить» всю мережу і здатний керувати всіма мережевими пристроями, тому додаткам і мережевим політикам мережа представляється як єдиний логічний комутатор. Вказана обставина надзвичайно спрощує створення, управління і обслуговування мережі, також спрощуються і самі мережеві пристрої.

- **Рівень інфраструктури** складається з мережевих пристроїв, які забезпечують просування пакетів даних.

Перехід на технологію ПКМ і створення на її основі ІоЕ - одні з найбільш проривних і перспективних завдань ІТ-індустрії. У світі ці технології з розряду інновацій поступово переходять в ранг практично використовуваних промислових технологій, але дискусії про способи їх конкретної реалізації ще далекі від завершення. І в ПКМ-технології, і в питаннях створення ІоЕ акценти зміщуються з апаратного рівня, де позиції України слабкі, на програмний рівень, що відкриває нові можливості для наших розробників. Україна має багато хороших програмістів, яких вже оцінили у всьому світі. Таким чином, і Україна може вплинути на те, як будуть розвиватися комп'ютерні мережі та Інтернет в найближчому майбутньому.

1.1.7 ІоЕ, ІоТ: менеджмент та кадрові ресурси

Керівникам компаній варто усвідомити, що зміни в світі цифрових технологій відбуватимуться все швидше [44]:

невиробничі компанії:	раніше - 3-5 років, тепер - 1-1,5 років.
промисловість:	раніше - 10-12 років, тепер - 2 роки.

З огляду на стрімке зростання Інтернету, прогнозоване на наступні 10 років, керівникам бізнесу та державним діячам, та й всім громадянам, дуже важливо вже сьогодні почати готуватися до майбутнього [72]. Ось ряд питань, які кожен з нас повинен задати самому собі [72]:

- Як розставити пріоритети щодо використання можливостей, що відкриваються в мережевому світі ІоЕ?
- З огляду на вплив, який Інтернет вже здійснює на мій бізнес, як на цей бізнес може повліяти експоненціальна хвиля підключень нових пристроїв?
- Які потенційні переваги і ризики, створювані середовищем ІоЕ для моєї компанії (або державної установи)?
- Як краще адаптувати організаційну структуру організації до майбутнього?
- Як зміняться питання управління і розподілу відповідальності в світі ІоЕ?

За даними опитування компанії Vodafone, в 2016 році, 28% з 1,1 тисячі керівників різних підприємств відповіли, що вони вже використовують Інтернет речей [13]. Ще 35% заявили, що збираються впровадити подібні проекти протягом року.

Компанії, які вже використовують технологію ІоТ, витрачають на це приблизно 24% ІТ-бюджету. Решта йде на хмару і хостинг, аналітику і підтримку мобільності.

63% опитаних менеджерів заявили, що вони бачать значне збільшення повернення інвестицій при впровадженні ІоТ. У 25% організацій, де застосовують ІоТ, до інтернету речей підключено більше 100 пристроїв, а в 10% уже має більше 10 000 підключених пристроїв.

1.1.8 Економічні показники

Результати проведеного компанією Cisco [66] дослідження дають підстави вважати, що в найближче десятиліття Всеохоплюючий Інтернет збільшить сукупні прибутки світових корпорацій приблизно на 21%. Найважливішим джерелом цього "фонду" стануть нові послуги, викликані до життя безліччю мережевих датчиків, пристроїв та інших фізичних об'єктів, які становлять Інтернет речей і радикально перетворять нинішні бізнес-процеси, роблячи їх динамічними, інтелектуальними, цифровими та інноваційними.

"Протягом наступного десятиліття в міста переїдуть ще 700 млн чоловік. Міста що існують піддадуться серйозним перетворенням, а в світі з'явиться 100 нових міст з населенням більше мільйона чоловік, - заявив головний директор по глобалізації, виконавчий віце-президент компанії Cisco Вім Елфрінк (Wim

Elfrink), що відповідає за нові технологічні рішення. - Всі ці зміни вимагають трансформації методів доставки послуг міським жителям і організаціям. Інтернет речей, заснований на інтелектуальній мережевої інфраструктури, в поєднанні з рухом в сторону відкритих даних і додатків, в тому числі хмарних, стане основою цієї трансформації і дозволить більш продуктивно, безпечно і відповідально доставляти освітні, медичні, транспортні, житлові, захисні та інші сервіси".

За прогнозом дослідницького агентства IDC, світовий ринок IoT в 2017 році повинен був перевищити позначку \$ 800 млрд, а до 2020 досягти \$ 1,29 трлн [13]. У ці показники входять закупівлі обладнання, послуги, зв'язок і софт в даній сфері. Найбільш ємні сфери застосування - промисловість (\$ 185 млрд), транспорт (\$ 85 млрд) і інженерні системи (\$ 66 млрд). За консервативними оцінками, в світовому масштабі потенційна економічна вигода ІоЕ в найближче десятиліття може скласти \$ 19 трлн [43]. З них

- \$ 14,4 трлн - приватний бізнес
- \$ 4,6 трлн - держсектор.

Ця економічна вигода буде створюватися різними способами:

- підвищення ефективності використання активів і логістики поставок,
- створення інноваційних рішень,
- поліпшення якості обслуговування замовників і населення.
- і багато інших.

Однак потенційна вигода ІоЕ в розмірі \$ 19 трлн просто не може дістатися будь-якої однієї компанії, тому, перш за все, необхідно створення глобальної партнерської екосистеми.

Одним з головних генераторів великих даних стане урбанізація. Передбачається, що в одному тільки Китаї в найближчі 30 років додасться до 100 міст-мільйонників. В Індії ж за той же період міське населення збільшиться на 400 млн осіб. На цьому тлі зростання продуктивності праці, що спостерігалось останні 20 років, не виглядає таким вже значним в порівнянні з тим, що потрібно зробити з огляду на прийдешніх змін (Рис. 1.1.8.1).



Рис. 1.1.8.1. Огляд змін.

1.2 ІоЕ – поточні задачі та технології

1.2.1 Що потрібно для реалізації переваг ІоЕ, ІоТ

Основні можливості для реалізації ІоТ вже є [13].

- Новий інтернет-протокол - IPv6 надає практично нескінченне число IP-адрес ($3,4 \times 10^{38}$), що достатньо для очікуваного підключення 50 млрд пристроїв.
- Британська компанія ARM Holdings створила мікročіп, який маючи розмір 1 кв.мм з наднизьким споживанням енергії дозволяє вийти в Мережу майже будь-якому предмету.
- Стартап Supermechanical розробив бездротової модуль Twine, здатний підключити до Мережі практично будь-яку річ і навчити її відправляти SMS-повідомлення, електронні листи або "твіти" (вартість модуля від \$ 100).

Додаткові умови для створення повноцінного ІоЕ [31]:

- реалізація мобільності, тобто можливості доступу до ресурсів будь-якого пристрою в будь-який час і з будь-якого місця;
- реалізація хмарних обчислень, тобто виконання обчислень із залученням розподілених по мережі комп'ютерних ресурсів та сервісів;
- реалізація концепції «великих даних» (Big Data), тобто забезпечення можливості створення, збору і обробки великих обсягів даних;

ІоЕ докорінно змінює структуру бізнесу, створює колосальні економічні багатства і незліченні соціальні цінності. Але для успішного впровадження технологій ІоЕ необхідна міцна технічна і комерційна платформа, на якій легко вибудовувати різні вертикальні рішення для раціонального і ефективного досягнення обіцяних комерційних, економічних і соціальних переваг [29]. В основі такої платформи - надійний зв'язок і технологічна інфраструктура, операційні та управлінські сервіси, що підтримують ряд вертикальних і горизонтальних рішень. Ідеальна платформа ІоЕ повинна відповідати вимогам всіх вертикальних рішень до створення масштабованої технічної та комерційної архітектури за принципом «створив один раз - користуйся багаторазово». В Cisco вважають, що сервіс-провайдери мають всі можливості для розробки і реалізації платформи ІоЕ. Крім багаторічної роботи в сфері побудови та експлуатації мереж, служб і клієнтських сервісів, багато провайдерів тепер пропонують розширені хмарні сервіси і сервіси по інтеграції мереж. Сервіс-провайдерам слід уважно вивчити свої стратегічні можливості по розробці і експлуатації платформи ІоЕ.

Стратегічні можливості провайдерів при реалізації платформи зростання [29]

Все залежить від кожного конкретного провайдера. Разом з тим стратегічні можливості тягнуться від технічного потенціалу провайдерів у

самого «фундаменту» платформи до менш профільних можливостей рівнів обслуговування та управління на її «вершині»:

1. **Основна діяльність:** Мережевий зв'язок (розширення основного мережевого бізнесу).

2. **Суміжні сфери:** Мережевий доступ, технологічна платформа (ефективне використання можливостей при розгортанні та забезпечення послуг доступу до мережі і хмарних обчислень).

3. **Розширення:** Вертикальні і горизонтальні рішення, платформа загальних служб (можливе партнерство / придбання для створення рішень на базі Всеохоплюючої Інтернету; розширення внутрішнього центру управління мережею і клієнтських центрів).

4. **Нові сфери:** Професійні послуги, управління розробкою і підтримкою програм (хоча деякі сервіс-провайдери мають такі можливості, більшості доведеться знаходити партнерів або поступово розвивати консультаційні та сервісні можливості).

5. **Монетизація:** розширення і розвиток існуючих можливостей в сфері реклами і аналізу даних для створення вертикальних можливостей монетизації.

Приблизно половина прибутку від впровадження платформи доводиться на основну діяльність або суміжні сфери, які не дуже далекі від поточного бізнесу провайдера [29].

Рекомендації относительно активності провайдерів при реалізації платформи зростання [29]

Сервіс-провайдери у всьому світі намагаються скористатися величезними можливостями платформи ІоЕ. Але поки вони, як правило, зосереджені на своїй основній діяльності щодо забезпечення мережевих з'єднань. Якщо ж піднятися над платформою ІоЕ, то відкриються ще більші можливості по розробці і забезпеченню нових джерел прибутку.

Сервіс-провайдерам слід, перш за все, вийти за звичні рамки в суміжні сфери мережевого доступу і технологічної платформи і почати розширення і освоєння нових областей, нарощуючи існуючі можливості, залучаючи партнерів або розглядаючи ті чи інші варіанти придбання. Разом з тим необхідно досліджувати області, де можна створити нові схеми монетизації по окремих вертикалях, ефективно використовуючи існуючі і нові бізнес-можливості. Побудова і обслуговування масштабованої платформи ІоЕ, яка може бути розгорнута на базі безлічі вертикальних рішень, - ось та бізнес-модель, яка необхідна сервіс-провайдерам, щоб отримати значну частину з 20 трильйонів доларів, які, за оцінкою IDC, будуть протягом наступних трьох років витрачені на реалізацію потенціалу ІоЕ. Сервіс-провайдери мають всі можливості очолити цей процес в якості провайдерів платформи ІоЕ. Але їм краще поквартитися: розробники технологій, які обслуговують компанії, стартапи та нішеві гравці теж не дримають, прагнучи отримати частку в цьому привабливому бізнесі.

Але ІоЕ охоплює набагато більше, ніж одні тільки «речі» [29]. Насправді Інтернет речей має на увазі чотири надійно захищені технологічні опори, що формують принципово новий підключений світ:

- мобільні технології,
- хмарні технології,
- великі дані
- речі.

Успішна реалізація рішень на базі ІоЕ - не ізольований і незалежний процес. Для цього потрібно технічна і комерційна платформа, на якій можна буде легко вибудовувати різні рішення для раціонального і ефективного досягнення обіцяних комерційних переваг. В основі такої платформи ІоЕ лежать надійний зв'язок і технологічна інфраструктура, операційні та управлінські сервіси, а також ряд вертикальних і горизонтальних рішень.

Досвід Cisco показує, що для реалізації рішень на базі ІоЕ всі технічні та комерційні елементи повинні забезпечувати потрібний результат. Ми впевнені, що ефективне розгортання систем ІоЕ забезпечить таку платформу для всього бізнесу або навіть для всіх галузей, яка дозволить реалізовувати цілий ряд унікальних прибуткових рішень на базі ІоЕ.

Рівні реалізації, починаючи з нижнього:

1. **Мережеві підключення** - з'єднання всіх рішень, даних і додатків за допомогою оптоволоконної транзитної або ліцензованої мережі.
2. **Мережевий доступ** - керування мережею Wi-Fi або інша неліцензована бездротова мережа для підключення всіх датчиків і додатків.
3. **Технологічна платформа** - платформа, що забезпечує швидке і надійне підключення нових пристроїв до архітектури за принципом «підключи і працюй», а також з'єднання з хмарними сервісами зберігання і обробки даних.
4. **Вертикальні і горизонтальні рішення** - сукупність пристроїв і додатків, що забезпечує унікальні рішення для різних вертикальних і горизонтальних галузевих сегментів.
5. **Платформа монетизації** - в деяких вертикалях, таких як «розумні» міста і сегмент В2С, існують можливості ефективного використання платформи для створення нових джерел прибутку.
6. **Загальна платформа управління** - загальна платформа, що забезпечує управління, обслуговування клієнтів і сервіси для всіх рішень.
7. **Професійні послуги** - спеціальні сервіси, такі як інтеграція систем, планування і проектування.
8. **Керівництво проектом** - сервіси з управління проектом, операціями і екосистемою партнерів.

Успішне розгортання рішень і отримання величезної потенційної вигоди від ІоЕ залежить не тільки від класних речей і додатків. Для втілення ідей та сподівань в життя необхідна комплексна, технічна, операційна і організаційна платформа ІоЕ (Рис. 1.2.1.1).

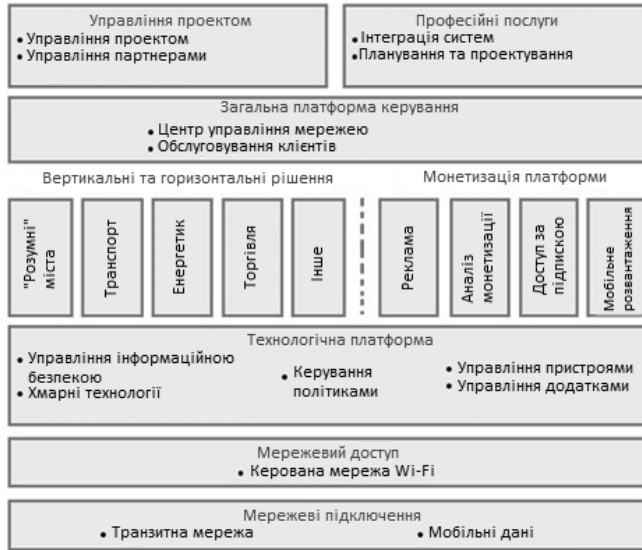


Рис. 1.2.1.1. Технічна та комерційна платформа ІоЕ.

1.2.2 Яких інновацій та нових рішень потребує ІоЕ

«Кожна нова технологія породжує ще дві або три технології, і ми не знаємо, що нас чекає за найближчим роком. Ще 20 років тому не було пошукових систем, 15 років тому не було соціальних мереж, 10 років тому не було YouTube, 5 років тому не було інтернету речей, тому ми не можемо сказати, що буде через 10-20 років. Ця історія буде розвиватися незалежно від нас. Я спробую висловити свою думку так: ще 10 років тому ми могли розділяти реальний і віртуальний світ. Зараз реальний світ - це віртуальний світ, а віртуальний світ - це реальний. Ми їх не можемо розділяти»

(Вольфганг Кляйнвайхтер, професор Орхусського університету, Данія, 2017 р. [13])

1.2.2.1 Стимуляція іновацій від ІоЕ

«Всеосяжному Інтернету потрібні розподілені, зорієнтовані на додатки мережі, а також обчислювальні системи і платформи зберігання даних, здатні підключати речі так, як ще вчора це було неможливо зробити і навіть не піддавалося людській уяві. Вибухоподібне зростання кількості підключень неможливе без IP-пристроїв, широкого розповсюдження ширококутового доступу і протоколу IPv6. Наша мережа створює інтелектуальну, керовану, відмінно захищену інфраструктуру, здатну масштабуватися і підтримувати мільярди підключених пристроїв»

(Голова ради директорів, головний виконавчий директор компанії Cisco Джон Чемберс (JohnChambers, [13])

ІоЕ поклав край «старому» Інтернету, давши при цьому початок новому, що дозволяє створювати широкий спектр «електронних результатів» [43]. «Електронні результати» - наслідок можливості обробляти великі дані. Простота створення джерел даних і вилучення з них цифрової інформації обертається фундаментальною проблемою: що з усіма цими даними робити?

Електроніка в звичайній машині Toyota за рік виробляє 2 петабайта інформації [44]. Щоб прослухати один петабайт музики, завантаженої на mp3 плеєрі, вам буде потрібно дві тисячі років. Сьогодні інформація подібна нафті - її багато, але вона вимагає переробки, тобто осмислення, і тільки після цього вона може стати джерелом знань і мудрості. Саме тому так важливі аналітика і додатки, які користуються все більшим попитом і популярністю:

в 2007 році було всього три тисячі мобільних додатків,
а в 2014 році було скоєно 77 млрд завантажень!

Головне завдання полягає в тому, щоб перетворити ці дані в інформацію і знання, а потім, шляхом сценарного планування, зробити на цій основі розумні висновки і з їх допомогою отримувати економічну вигоду [43]. Однак неможливо реалізувати весь потенціал ІоЕ без універсальних стандартів зв'язку між пристроями [43]. Згадаймо: сам Інтернет починався з доброї сотні різних протоколів, число яких сьогодні зросло до мінімуму. Те ж саме повинно відбутися і в промисловості, де в одному лише будівництві застосовується більш 300 різних протоколів. Величезний шанс для такої стандартизації дає мережа, де всі пристрої прагнуть говорити на мові протоколу IP. Галузі і компанії, які першими почали перетворення, стануть лідерами, а ті, хто упустили свій шанс, ризикують зникнути, бо амортизаційні цикли промислового обладнання тривають роками.

1.2.2.2 Окремі іновації IoT, IoE: проект Cesanta - Mongoose OS

Сьогодні мережевий трафік - це, перш за все, комунікації "людина-людина" [13]. Перехід до трафіку "машина-машина" (M2M) або "фізичний об'єкт-фізичний об'єкт" є справжню революцію, хоча мережеві об'єкти з'являються тихо і непомітно. Для об'єднання сотень мільярдів - трильйонів крихітних пристроїв потрібна автоматична Мережа, яка не тільки практично миттєво реагує на запити, але й здатна викликати, відслідковувати, контролювати, включати і вимикати пристрої без втручання людини. Програмно-визначаємі мережі (Software Defined Networking, SDN) стануть основою для будівництва Інтернету нового покоління.

На початку 2015 року, колишні співробітники Google Анатолій Лебедев і Сергій Любка вирішили створити свій бізнес в сфері IoT - проект Cesanta . Вони розробили операційну систему Mongoose OS, яка прискорює процес створення приладів і пристроїв з виходом в Мережу. Коли Білл Гейтс з невеликою командою в Альбукерке розробляв в 1975 році мову програмування Basic для декількох тисяч ентузіастів, це нікому не здавалося вражаючим, згадає Лебедев, але «подивіться, чим це є зараз в світі операційних систем для персональних

комп'ютерів». Слоган молодого бізнесу звучить: «Bringing all devices online» - «Підключити всі пристрої до інтернету». Основний продукт компанії, за задумом засновників, має в перспективі декількох років стати повноцінною екосистемою для підключення і взаємодії між собою різноманітних фізичних пристроїв, від кухлів і холодильників до мікрохвильових печей і автомобілів.

Mongoose OS - це продукт для розробників, який допомагає скоротити час перетворення будь-якого нового приладу в «розумний». У більшості випадків розробка IoT-продукту починається з нуля: продумується логіка пристрою (які функції і завдання повинні виконуватися за певних умов), потім починається інфраструктурна робота - інтеграція мікроконтролера, написання спеціальної прошивки, її програмування на певну послідовність дій, поновлення і т.д. Перша частина циклу займає близько 10-20% часу, друга - все інше. Mongoose OS, за словами Лебедева, містить в собі як раз ці 80-90% інфраструктури. «Розробникові залишиться витратити трохи часу на фінальну логіку. Якщо ми говоримо про розробку середнього пристрою, створення прошивки для нього займає 12-18 місяців. З нами це можна зробити за тиждень - максимум місяць-два», - обіцяє він [13].

Клієнтами Mongoose OS є кілька десятків фірм (точне число не розкривається). Всі вони платять роялті за кожний зроблений пристрій. За словами Лебедева, вартість розраховується індивідуально. Mongoose OS є партнером Cloud IoT Core і хмарної системи Amazon AWS.

Один з клієнтів Cesanta - швейцарська компанія Duagon AG, яка виробляє електронну начинку для управління рухом поїзда. Завдяки Mongoose Web Server обладнання можна підключити до каналу зв'язку (інтернету або Ethernet) і отримати до нього доступ. «Без Mongoose Web Server розробник отримав б не зрозумілий інтерфейс, а програмний код з нулів і одиниць», - каже Лебедев. Влітку 2015 користувачем Mongoose Web Server стало NASA, яке впровадило продукт на Міжнародній космічній станції для проведення експериментів. Крім NASA Mongoose Web Server використовують Apple в картах Apple Maps, Dell, Samsung, Bosch, Google, HP, Epson, Qualcomm, Nintendo і інші [13].

Аналогічними рішеннями, схожими на Mongoose OS, є платформи mbed OS і Lite OS. Керівник відділу «Лабораторії Касперського» Андрій Нікішин розповів журналу РБК, що в 2016 році була створена Kaspersky OS для інтернету речей.

«Операційна система Mongoose OS розрахована на прості пристрої, що володіють невеликою потужністю і функціоналом», - розповідає Нікішин. Kaspersky OS призначена для використання в індустріальних і промислових мережах і пристроях. Вона встановлена в маршрутизатор Kraftway, який продається на російському ринку. За словами Нікішина, Kaspersky OS орієнтована «на більш складні і потужні кінцеві пристрої», наприклад на «розумні» камери відеоспостереження, роутери та комутатори. «Тому наша операційна система не конкурує з Mongoose OS», - упевнений представник Kaspersky Lab [13].

1.2.3 Технології та технічні рішення ІоЕ

1.2.3.1 ІоТ. Огляд протоколів

Пристрої (D) повинні встановлювати один з одним зв'язок (D2D) [53]. Потім потрібно зібрати і передати дані з цих пристроїв в серверну (S) інфраструктуру (D2S). Ця серверна інфраструктура повинна спільно використовувати дані (S2S), маючи можливість передавати дані назад пристроям, програмам аналізу або людям. Якщо подивитися «з висоти пташиного польоту», то можна виділити такі протоколи для вирішення завдань в цій інфраструктурі:

- **MQTT**: протокол для збору даних пристроїв і передачі їх серверів (D2S);
- **XMPP**: протокол для з'єднання пристроїв з людьми, окремий випадок D2S-схеми, коли люди з'єднуються з серверами;
- **DDS**: швидка шина для інтегрування інтелектуальних пристроїв (D2D);
- **AMQP**: система організація черг для з'єднання серверів між собою (S2S).

Кожен з цих протоколів широко поширений. Є принаймні 10 варіантів реалізації кожного з них. На верхньому рівні протоколи досить схожі. По суті, всі чотири перераховані вище протоколи є протоколами «Інтернету речей» реального часу з публікацією / підпискою, які здатні поєднувати тисячі пристроїв. Справедливість цього залежить від того, як Ви визначаєте терміни «реальний час», «речі» і «пристрої». Насправді ці протоколи дуже різні. Сучасний Інтернет підтримує сотні протоколів. ІоТ буде підтримувати ще на сотні протоколів більше. Важливо усвідомити сферу застосування кожного з них. Схема, на рис. 1.2.3.1, відображає основні сценарії використання цих протоколів. Звичайно, насправді це не так просто. Наприклад, «підсистема управління» (control plane), яка забезпечує контроль за всіма з'єднаннями і управління ними, досить складна. Багато протоколи взаємодіють саме в її рамках [53].

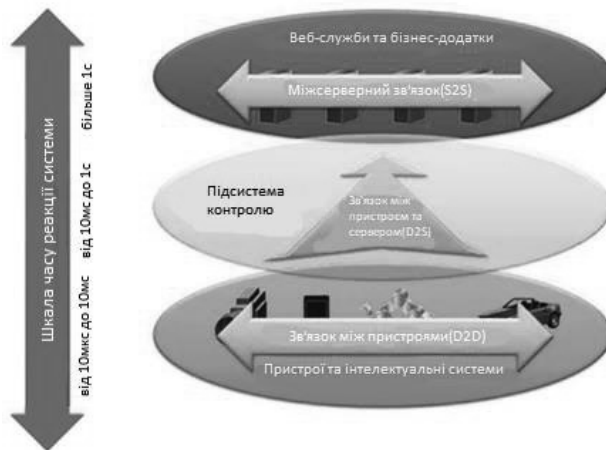


Рис. 1.2.3.1. Протоколи ІоТ та час реакції системи.

MQTT

Протокол MQTT (Message Queue Telemetry Transport) обслуговує збір даних з пристроїв (Рис. 1.2.3.2). Як впливає з назви цього протоколу, основне його призначення - телеметрія, або дистанційний моніторинг. Його завдання полягає в зборі даних від безлічі пристроїв і передача цих даних в IT-інфраструктуру. Він призначений для великих мереж невеликих пристроїв, якими потрібно управляти з хмарної мережі [53].

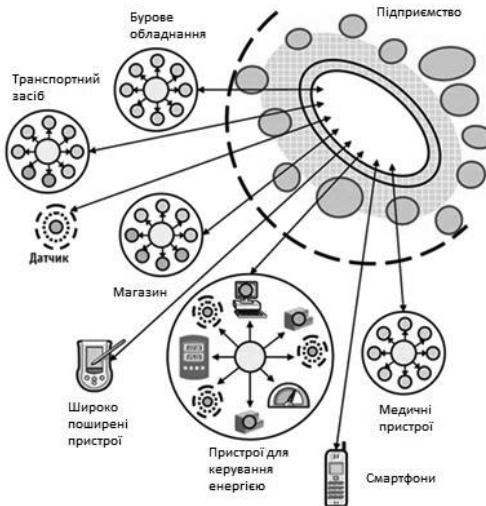


Рис 1.2.3.2. Протокол MQTT реалізує систему на основі зіркоподібній мережі.

Протокол MQTT забезпечує лише можливість передачі між пристроями, але не розподіляє дані по багатьом одержувачам. Оскільки протокол MQTT має цілком певне застосування, він нескладний і забезпечує всього декілька опцій управління. Крім того, йому не потрібно бути дуже швидким. У зв'язку з цим режим «реального часу» зазвичай вимірюється в секундах. Для MQTT зазвичай використовується архітектура зіркоподібній мережі. Всі пристрої підключені до сервера концентратора даних, подібно до нового рішення MessageSight від IBM. Оскільки втрачати дані зовсім не бажано, цей протокол працює поверх TCP, що забезпечує простий і надійний потік даних. У зв'язку з тим, що ці дані використовує IT-інфраструктура, вся система призначена для простої передачі даних в системи підприємств і компаній, наприклад такі, як ActiveMQ і сервісні шини підприємства (enterprise service buses - ESB) [53].

Протокол MQTT забезпечує такі додатки, як моніторинг витоків та контроль за можливим умисним псуванням нафтопровідної мережі. Тисячі датчиків такої системи повинні бути зосереджені в одному місці для аналізу. Коли в такій системі виникає якась проблема, є можливість взяти заходів для корекції цієї проблеми. Іншими додатками для MQTT можуть бути контроль споживання енергії, управління світлом і навіть інтелектуальне садівництво. Цей протокол бере участь в зборі даних від безлічі джерел і робить їх доступними для IT-інфраструктури [53].

XMPP

Протокол XMPP спочатку називався **Jabber**. Він був розроблений для системи миттєвого обміну повідомленнями для зв'язку між людьми за допомогою текстових повідомлень (рис. 1.2.3.3). XMPP означає Extensible Messaging and Presence Protocol, або розширюваний протокол обміну повідомленнями та інформацією про присутність. І знову назва дає неправильне уявлення про цільове використання, адже присутність означає безпосередню участь людей [53].

У протоколі XMPP використовується текстовий формат XML в якості вбудованого типу, забезпечуючи природний зв'язок між людьми. Подібно MQTT він працює по TCP або, можливо, по HTTP поверх TCP. Його перевагою є метод адресації виду `name@domain.com`, який допомагає з'єднати користувачів в величезному просторі Інтернету. Стосовно «Інтернету речей» XMPP забезпечує простий спосіб адресації пристроїв. Це особливо зручно, коли дані передаються між віддаленими, найчастіше незалежними точками, як у випадку зв'язку між двома абонентами. Цей протокол не володіє високою швидкістю. В більшості реалізацій цього протоколу використовується метод опитування або перевірки доповнень тільки на вимогу. Протокол, званий BOSH (Bidirectional streams over Synchronous HTTP) дозволяє серверам приймати повідомлення. Але масштаб «реального часу для XMPP близький до людського масштабу і вимірюється в секундах. XMPP реалізує істотну перевагу, наприклад, при підключенні Вашого домашнього термостата до веб-сервера, щоб Ви могли отримати до нього доступ з вашого телефону. Сильними сторонами цього протоколу є адресація, безпека і масштабованість, що робить його ідеальним для додатків «Інтернету речей» з орієнтацією на споживача [53].

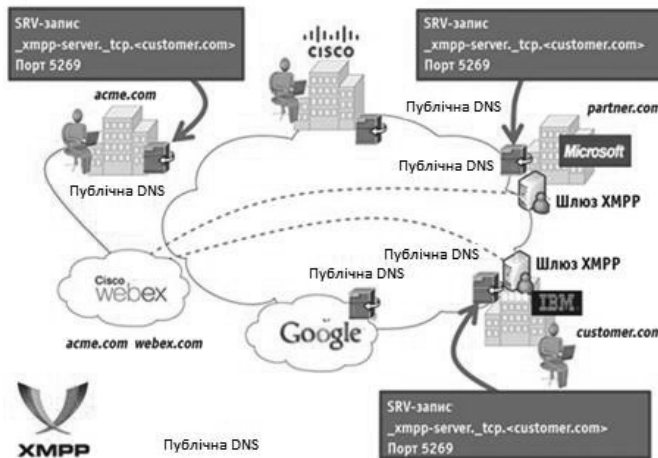


Рис 1.2.3.3. Протокол XMPP забезпечує обмін текстовими повідомленнями між точками доступу.

DDS

На відміну від MQTT і XMPP протокол DDS (Data Distribution Service - сервіс розподілу даних) обслуговує пристрої, які безпосередньо використовують дані пристроїв. DDS розподіляє дані по іншим пристроям (Рис. 1.2.3.4). Хоча DDS підтримує інтерфейс з IT-інфраструктурою, основне його завдання полягає в з'єднанні пристроїв з пристроями. Це орієнтований на дані стандарт вбудованого ПЗ, що використовується в високопродуктивних військових, промислових і вбудованих додатках. DDS здатний ефективно і синхронно доставляти безлічі одержувачів мільйони повідомлень в секунду .



Рис. 1.2.3.4. Протокол DDS реалізує архітектуру публікації і підписки.

Пристрої запитують дані інакше, ніж IT-інфраструктура. По-перше, пристрої працюють швидко. Масштаб «реального часу» часто вимірюється в мікросекундах. Пристроєм потрібно здійснювати зв'язок з іншими пристроями, використовуючи складні шляхи, тому прості і надійні двоточкові TCP-потіки даних обмежують можливості такої передачі. DDS забезпечує деталізований контроль QoS, багатоадресну передачу, переналаштувану надійність і всеосяжну надмірність. Сильною стороною DDS є розгалуження даних. DDS забезпечує потужні способи фільтрації та відбору даних за адресами призначення. Синхронних одержувачів даних можуть бути тисячі. Деякі пристрої досить компактні, тому існують полегшені версії DDS. Для використання даних від пристроїв зіркоподібна мережа зовсім не годиться. Замість цього DDS реалізує прямий шинний зв'язок між пристроями на базі реляційної моделі даних. У компанії RTI її називають шиною даних (DataBus) - мережевий аналог бази даних (database). Подібно до того, як база даних управляє доступом до збережених даних, шина даних управляє доступом до даних і оновленнями одночасно багатьма користувачами. Високопродуктивні системи інтегрованих пристроїв використовують DDS. Це єдина технологія, яка забезпечує гнучкість, надійність і швидкість, необхідні для побудови складних додатків реального часу. Ці програми включають в себе військові системи, вітроелектростанції, інтегровані системи лікарень, системи діагностичної

візуалізації, системи супроводу ресурсів і автомобільні системи випробувань і забезпечення безпеки. DDS з високою швидкістю з'єднує пристрої всередині працюючої розподіленої системи [53].

AMQP

Протокол AMQP (Advanced Message Queuing Protocol, вдосконалений протокол організації черги повідомлень) іноді розглядають як протокол «Інтернету речей». AMQP обслуговує виключно черги і пересилає транзакційні повідомлення між серверами. AMQP, як орієнтований на повідомлення проміжного програмного забезпечення, був створений для банківської галузі і здатний обробляти тисячі організованих в чергу транзакцій. В AMQP велику увагу приділено боротьбі з втратою повідомлень. Різні комунікаційні лінії - від постачальників контенту до комутаторів каналів і від списків черговості до користувачів - використовують протокол TCP, який забезпечує високонадійне двухточкове з'єднання. Крім того, кінцеві точки повинні підтверджувати прийняття кожного повідомлення. Даний стандарт описує також опціональний режим транзакцій з формальної багатофазної послідовності відновлення транзакцій. Залишаючись вірним своїм походженням в банківській галузі, проміжне програмне забезпечення AMQP сфокусовано на спостереженні за всіма повідомленнями і забезпеченні їх доставки за призначенням, незалежно від наявності збоїв і перезавантаження системи (Рис. 1.2.3.5). AMQP використовується головним чином в обміні діловими повідомленнями. Зазвичай під «пристроями» розуміють мобільні телефони, що підтримують зв'язок з центрами обробки даних в операційних офісах. У разі «Інтернету речей» AMQP найкращим чином підходить для реалізації функцій «підсистеми управління» (control plane) або аналітичних функцій на базі серверів [53].

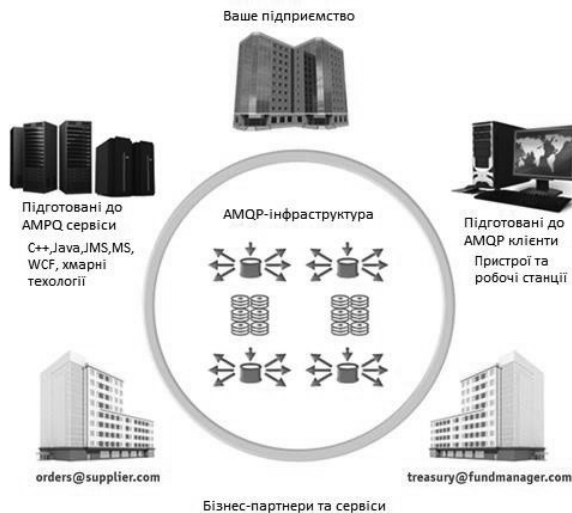


Рис. 1.2.3.5. Протокол AMQP - орієнтоване на повідомлення проміжне ПО, створене для банківської галузі.

Висновок щодо протоколів

IoT вимагає безлічі протоколів. Розглянуті типи протоколів помітно різняться між собою. Найпростіше класифікувати їх за ключовими параметрами: якість сервісу, адресація і додаток [53].

QoS значно краще характеризує протокол, ніж поняття «режиму реального часу». Контроль якості сервісу пов'язаний з гнучкістю доставки даних. Систему зі складним контролем якості сервісу важче програмувати, але на її основі можна побудувати додаток з більш високими вимогами [53].

Надійність. Більшість протоколів працює поверх TCP, що забезпечує надійність. Кожен байт, що потрапляє в комунікаційний канал, повинен бути доставлений одержувачу, навіть якщо знадобиться безліч спроб. Це реалізується просто і найчастіше без проблем, але не дозволяє управляти синхронізацією. У разі повільного абонента одноканальний TCP-трафік дублюється [53].

За контролем якості сервісу помітно відрізняється протокол DDS, оскільки орієнтований на комунікації між пристроями. Крім надійності, DDS забезпечує також контроль так званої «життєздатності» (при дослідженні проблем), виявлення і використання ресурсів, а також синхронізації [53].

Фундаментальною проблемою є пошук даних у величезній базі даних IoT. Тут краще проявляє себе протокол XMPP. Його схема адресації типу user @ domain повністю використовує правила, що встановилися в Інтернеті. Однак XMPP не здатний за простою обробляти великі масиви даних, що відносяться до одного сервера. Протокол MQTT, завдяки своєму зв'язку з сервером, справляється з подібним завданням краще. Якщо Ви можете підключитися до такого сервера, то Ви з'єднуєтеся з мережею. Черги AMQP діють подібно до серверів, але для S2S-систем. І в цьому випадку DDS сильно відрізняється від інших протоколів. Замість сервера він використовує спеціальний фоновий протокол, який автоматично відшукує дані. DDS-системи зазвичай більш обмежені. Особливої уваги потребує пошук по розподіленій мережі (WAN) або великим групам пристроїв [53].

Ключові особливості протоколів залежать від їх передбачуваного застосування. Використання даних між пристроями в корені відрізняється від збору даних на пристрої. Наприклад, включення освітлення (в цьому випадку оптимальним варіантом є XMPP) абсолютно відрізняється від генерування енергії (DDS), моніторингу ліній передачі (MQTT) або аналізу споживання енергії в центрі обробки даних (AMQP). Звичайно, де що при застосуванні протоколів перетинається. Наприклад, DDS може обслуговувати і приймати дані з хмарних середовищ, а MQTT - передавати інформацію назад в пристрої. Проте, основні завдання всіх чотирьох протоколів різні, різними є архітектури та можливості. Всі ці протоколи критичні до (швидкої) еволюції «Інтернету речей». «Інтернет речей» - великий простір, в якому існує безліч протоколів. Тому до вибору оптимального протоколу для свого застосування потрібно підходити ґрунтовно, і об'єктивно зважувати всі позитивні і негативні властивості кожного з них [53].

1.2.3.2 Беспроводні та мобільні пристрої

Глобальний Інтернет, так само як і Інтернет речей, не може існувати без мобільного зв'язку і бездротових підключень [10]. За даними Cisco, з 2000 по 2015 рік число абонентів мобільного зв'язку на Землі збільшилося вп'ятеро, а сьогодні їх вже $\frac{3}{4}$ від населення світу (близько 5,5 млрд.). Число користувачів мобільного зв'язку, «розумних» пристроїв, мобільного відео та мереж 4G в найближчі п'ять років буде зростати лавиноподібно, що призведе до восьмикратного збільшення мобільного трафіку. Принципово важливо, що більше $\frac{2}{3}$ усіх мобільних пристроїв і з'єднань відносяться до категорії «розумних», а смарт-гаджети генерують майже 98% мобільного трафіку передачі даних. У Нідерландах вже запущена національна мережа Інтернету речей [13]. Найпоширенішими персональними бездротовими пристроями стають смартфони: на їх частку припадає близько 80% усього мобільного трафіку [10]. На першому місці за темпами зростання серед усіх мобільних додатків - мобільне відео (понад 75% світового мобільного трафіку даних).

1.2.3.3 Аудиовізуальні технології (AVVR)

За прогнозом дослідницької компанії Gartner, вже в 2017 році ринок VR-окулярів і шоломів мав вирости в 4,5 рази - з 1,4 млн до 6,3 млн одиниць в порівнянні з 2016 роком [10]. CitiGroup оцінює обсяг VR-ринку до 2020 року \$80 млрд, а до 2035 року - \$2,16 трлн. Пристрої віртуальної реальності вже присутні в продуктових лінійках найбільших вендорів. Так, в 2014 році Facebook за \$2 млрд придбав виробника VR-окулярів Oculus, Google с цього ж часу розвиває лінійки пристроїв Daydream і Cardboard і набір сервісів Google VR. Власні VR-гарнітури випускає Intel спільно з італійським виробником окулярів Luxottica. Інші помітні гравці VR-ринку - Microsoft (Microsoft HoloLens), Sony (Sony PlayStation VR) і HTC (Vive).

ABI Research в найближчі п'ять років чекає бурхливе зростання ринку технологій доповненої реальності (AR) перш за все корпоративного застосування [10]. «Проект Google Glass був тільки початком великої трансформації ринку, яку готують AR-технології. Вже скоро ринок значно прискориться, виростуть обсяги інвестицій і впроваджень. Пристрої з більш потужним процесором, такі як окуляри ODG R7 і «розумний» шолом Daqri, найкраще підходять для промислового використання в охороні здоров'я», - заявив старший аналітик ABI Research Ерік Аббруцці (Eric Abbruzzese). При цьому, компанія Google визнала, що проект "розумних окулярів" Google Glass необхідно повністю переосмислити перед тим, як продовжувати займатися ним далі. Про це заявив фінансовий директор Google Патрік Пічетт під час конференц-кола за підсумками 2014 року [13]. В середині січня 2015 Google оголосила, що припиняє продаж першого покоління своїх очок, а сам проект Google Glass виходить з-під крила інноваційної лабораторії Google X. При цьому напрямок Google Glass стане відокремленим, а його курируванням

займеться Тоні Феделл, гендиректор компанії з виробництва " розумних "термостатів Nest Labs, яку Google купила на початку 2014 року за 3,2 млрд доларів [13].

Головний науковий співробітник Oculus Майкл Абраш (Michael Abrash) прогнозує, що основним способом управління VR і AR-пристроїв на найближчі 40 років залишаться контролери, подібні Oculus Touch [10]. У той же час технічний директор Google Рей Курцвейл (Raymond Kurzweil) переконаний, що віртуальна реальність в перспективі 10 років стане частиною гаджетів, що проєктують зображення на сітківку ока. В цьому випадку керуватися VR-гаджети будуть кивками або рухом очей.

Сьогодні технології віртуальної реальності знаходяться на початковій стадії розвитку. Перші VR-гаджети поки не враховують всіх потреб аудиторії. Однак дослідження показують, що користувачі зацікавлені в таких пристроях, так що їх подальший розвиток зумовлений очікуваннями користувачів: розробники і виробники націлені на підвищення чіткості зображення, збільшення потужності пристроїв, випуск якісного контенту з віртуальної складової, якого поки явно недостатньо. В кінцевому рахунку, поява посправжньому комфортних VR-пристроїв залежить від того, як скоро ними почне користуватися масовий споживач. Саме споживчий інтерес стимулює компанії швидко інтегрувати нові технології. Тут є кілька вирішальних чинників: швидкість поширення інформації про новий гаджет в інтернеті, в тому числі через соціальні мережі, а також швидкість зниження вартості гаджетів. Як показали дослідження, масова аудиторія поки не готова до віртуальної реальності, вбудованої в телебачення. Глядачі віддають перевагу звичній режисерській зйомці - вони не готові регулярно надягати VR-шолом, щоб перенестися в реальність, що транслюється телепрограми. Тому на сьогоднішній день ми не робимо ставку на рішення в області VR, продовжуючи орієнтуватися на потреби масового глядача. Однак віртуальна реальність - перспективна сфера. Це розуміють розробники, так що через 3-5 років, швидше за все, вдасться вирішити проблему продуктивності апаратного забезпечення, а також домогтися надвисокої роздільної здатності і високої якості графіки. Нові, більш досконалі, рішення в області VR можуть сприяти різкому зростанню продуктивності процесорів, зокрема графічних [10].

Чи потрібно обмежувати присутність людей в VR (на технологічному або законодавчому рівні)

Обмежувати поширення VR-технологій чи ні, буде залежати від того, яке місце віртуальна реальність займе в нашому житті і наскільки популярним стане використання VR-пристроїв. Поки ставити питання таким чином дещо передчасно. Але превентивні заходи захисту і обмеження присутності людини у віртуальній реальності, звичайно, необхідно продумати. На початковому етапі достатньо задіяти технологічний рівень, але це стане актуальним лише тоді, коли високотехнологічні VR-пристрої, інтегровані в розвинену ІоЕ-мережу, почнуть використовуватися масово [10].

1.3 ІоЕ – приклади сфер впровадження

1.3.1 Загальний огляд

У 21 столітті інформаційні технології розвиваються так швидко, що часто ми не встигаємо встежити за останніми новинками у сфері ІТ [54]. Всього за 8 років (2005-2013 рр.) мобільні пристрої проникли скрізь. Тепер кожен носить інтернет в своїй кишені. ІоТ теж прийшов в повсякденне життя через RFID теги, штрих-коди на практично будь-які товари [13]. Крихітні електронні пристрої на товар передають інформацію за допомогою радіохвиль і застосовується в магазинах для миттєвого зчитування на касі інформації про товар і його кількість на складі. Однак технологія по розробці RDIF давно зробила крок далеко вперед. Щоб скористатися перевагами цієї нової епохи, компанії починають повністю трансформувати свої процеси, операції і бізнес-моделі [29]. Ми вже бачимо появу «розумних» міст, підключених комунальних систем, підключених залізниць, підключених заводів, підключених автомобілів і навіть підключених шахт - і це далеко не повний перелік того, що буде підключено до глобальної мережі. Всі галузі промисловості розглядають Всеохоплюючий Інтернет як революційну технологію, що дозволяє оптимізувати бізнес, вийти на нові ринки і розширити зв'язки з замовниками. У найближчому майбутньому все більше число предметів, від одягу до будівель, зможуть виходити в інтернет. Це значно полегшить і убезпечить наше життя [13].

На автовиставці в Токіо, один з власників електромобіля Toyota показав на своєму смартфоні наступне повідомлення від свого автомобіля: «Зарядка акумулятора буде завершена до 2:15 ранку. Це нормально. Побачимося завтра?».

Сьогодні вже має місце така статистика [13]:

- 51% компаній використовують ІоТ для оптимізації використання активів. Наприклад, для контролю парку.
- 49% використовують ІоТ для скорочення операційних витрат. Наприклад, автоматизація будівництва.
- 46% використовують ІоТ для збільшення рівня безпеки та безпеки громадян і предметів. Наприклад, в камерах спостереження і системах моніторингу трубопроводів.
- 42% оптимізують за допомогою ІоТ ланцюжок поставок товарів і послуг від виробника споживачеві. Наприклад, продаж товарів через вендинг-автомати.
- 41% створюють нові підключення продукти і послуги. Наприклад, рішення для розумного будинку, віддалений моніторинг здоров'я, страхування.
- 40% проектів пов'язано з поліпшенням ефективності і безпеки публічних місць: розумні сміттєві баки, розумне вуличне освітлення, розумні міста.

Серед основних сфер застосування такі [12].

в промисловості:

- «Розумне» місто,
- транспорт,
- важка і легка промисловість,
- будівництво,
- енергетика,
- безпека,
- медицина,
- ритейл.

в повсякденному житті:

- носимі пристрої,
- "розумний будинок.

Більшість потенційних споживачів вже готова до переходу в еру IoT – IoE. На запитання «які б з IoE розумних пристроїв Ви б хотіли мати вже в наступні 5 років?», відповіді розподілились таким чином (опитування GSMA, 2015, [12]):

- розумну побудову техніку – 37%,
- розумні лічильники електричної енергії 25%,
- розумні носимі пристрої (годинники) – 13%,
- розумні авто - 10%,
- розумні медичні прилади – 10%,
- ничого з переліченого – 5%.

Окремі приклади інноваційних IoE пристроїв [13]:

- датчики на тілі тварин;
- кросівки, які переказують дані про успіхи бігуна;
- мікросхеми на таблетці, які передають дані зі шлунка;
- інгалятори для астматиків, які зв'язані з даними про погоду;
- датчики вологості, які відправляють вам SMS - повідомлення, якщо затопило підвал, будинок, квартиру;
 - холодильник, який сам може замовити необхідні продукти в супермаркеті;
 - одяг з мікрочіпом на дитину автоматично повідомить батькам, якщо він виявиться занадто далеко від будинку або раптом стане «підозріло мокрим»;
 - «розумна» зубна щітка, яка повідомляє через інтернет скільки разів ти почистив зуби своєму зубного лікаря і твоєї страхової компанії;
 - «розумна» вилка, яка знає, що ти їси і з якою швидкістю;
 - меблі, які аналізують вашу поставу і передають дані про ваш хребет.

Чисельно ми маємо таку картину [54]. Сьогодні ми маємо близько 50 млрд. підключених пристроїв по IP:

- транспорт - 500 млн пристроїв (аварійне реагування, самодіагностика автомобіля та ін.)
- розумне місто - 2 млрд пристроїв (розумне освітлення, розумні парковки, інтелектуальний транспорт, розумні будівлі і т.д.)
- безпека - 100 млн пристроїв (біометрія, сканери, контроль водних ресурсів, моніторинг навколишнього середовища та ін.)
- електронна охорона здоров'я - 100 млн пристроїв (мобільні лабораторії, хірургічне обладнання, віддалений моніторинг, телемедицина та багато інших)
- промисловість - 1 млрд пристроїв (управління активами, розумні насоси, компресори, клапани, безпека (LAS), Smart-мітки (NFC) і ін.)
- енергетика і ЖКГ - 500 млн - 1 млрд пристроїв (розумні лічильники, польові мережі і т.д.)
- торгівля - 200 млн пристроїв (заправні станції, ігри, соціальні події, автомати з їжею (вендинг), Smart-мітки і т.п.)

Сьогодні за допомогою Розумного виробництва заводи кожні 79 секунд мають можливість виробляти новий мотоцикл. Завдяки бездротовій мережі на заводах можна зменшити час на пошук потрібного контейнера з матеріалами і знизити невиробничі втрати. Наприклад, дохід за десять років від розумних автобусів в Барселоні досяг 28 млн. доларів.

1.3.2 Промисловість



Рис. 1.3.2.1. Стимули інвестицій [50].

«Розумне» устаткування забезпечує більш гнучке управління, поліпшену ефективність, знижений час простоїв, підвищений рівень продуктивності і гнучкість виробництва [50]. Підключення пристроїв вносять істотні зміни в багато галузей нашого життя - наприклад, енергоуправління на рівні домогосподарств або охорону здоров'я і т. п. Виробництво в цьому сенсі не виняток. Якщо обладнання має «зворотний зв'язок» зі своїм виробником, той може виявити дефекти задовго до того, як їх виявить споживач, зажадавши гарантійних компенсацій. Крім того, відомості про практичне застосування виробів можуть стати ключовою складовою вартості самих виробів. Принцип підключеності виходить за межі виробничого майданчика і внутрішніх операцій, тягнучись від машин до їх виробників. Виробничі операції, в основі яких покладено дані, згенеровані в межах заводу, позбавлені істотних можливостей. Обмін даними з виробниками, які знають свої машини краще, ніж хто-небудь ще, створює передумови для істотного підвищення продуктивності обладнання, роблячи його більш «розумним». «Розумне» устаткування забезпечує більш гнучке управління, підвищену ефективність, знижений час простоїв, більший рівень продуктивності і гнучке виробництво, що означає оптимізацію виробництва в цілому. З'єднання машин з їх виробниками і постачальниками засобів управління відкриває такі можливості:

- віддалене випереджаюче техобслуговування, що знижує час простоїв;
- використання діагностичних даних і оперативне вирішення технічних питань, що збільшує середній час роботи машин до ремонту, а також підвищення ефективності заводського обладнання;
- прискорений монтаж, швидке введення в експлуатацію та оперативний процес налагодження на нових та оновлених конвеєрних лініях.

Підключене устаткування залучає до спільної роботи конструкторів машин, постачальників засобів управління і виробників.

1.3.3 «Розумні» дома

За останні кілька років помітно зросла згадка терміну «розумний дім», хоча широкого впровадження автоматизація житлових будинків поки не отримала.

Історія цієї високоінтелектуальної системи почалася в 1961 році, з моменту створення подружжям Джоель і Рут Спіра спеціального пристрою для плавного регулювання світла – диммеру [38]. Цей винахід дав поштовх до утворення відомої сьогодні фірми Lutron Electronics Company, Inc, яка продовжила проектування «розумних будинків». Крім того, ця фірма впровадила нові поняття – світлова зона і сцена. Пізніше, в 1975 році шведська компанія Pico Electronics створила побутову автоматику, яка використовувалася для керування музичним програвачем. Нарешті, коли технології окремих систем «розумного будинку» були зібрані і вдосконалені, був створений і втілений в реальність перший проект – житловий будинок на південному березі Англії. Система автоматики здійснювала контроль над системами освітлення, сигналізацією будинку, жалюзі, опалення та дверима гаража. Крім того, був створений басейн з різними кольорними ефектами. Сучасні системи пішли далеко вперед і можуть контролювати буквально кожен сантиметр будинку і, крім того, системи «розумного будинку» влаштовані таким чином, що власник сам вибирає необхідний йому перелік контрольованих систем і спосіб управління системою.

Сьогодні аналітичні компанії, що виконують маркетингові дослідження в цій області, одноставно оцінюють цей ринок (з урахуванням продажів апаратури, послуг і установки), який швидко розвивається. Якщо в 2014 році, за даними компанії HIS Technology, частка "розумних будинків" в світі не перевищувала 1%, то до 2018 року цей показник збільшиться в три-чотири рази, і до цього часу буде встановлено 45 млн. таких будинків. Правда, вартісна оцінка ринку варіюється від \$12 млрд. в 2018 році (HIS Technology) до \$71 млрд. (Juniper Research - англійська компанія, що вивчає розвиток засобів телекомунікації та їх застосування). До 2020 року, згідно з прогнозами компанії Allied Market Research, світовий ринок "розумних будинків" і будівель досягне \$35,3 млрд. (середньорічні темпи зростання за 2014-2020 роки - 29,5%). Його розвиток частково стимулює зростання попиту на планшетні комп'ютери, які прихильники принципу "зроби сам" (Do It Yourself, DIY) широко використовують як ефективний засіб реалізації "розумного будинку". Зниження цін на пристрої "розумного будинку", увагу і стурбованість споживачів щодо стану навколишнього середовища і безпеки також сприяють збільшенню обсягу їх продажів [115].

Система "розумного будинку" складається з різноманітних апаратних і програмних засобів. У 2014 році багато гігантів, що виробляють побутову техніку, оголосили про своє бажання зайняти ніші ринку, що розвивається відносно недорогих приладів і систем "розумного будинку". Однак вони відчувають конкуренцію з боку операторів телекомунікаційних послуг,

постачальників засобів охорони, електронних приладів і програмного забезпечення, а також енергокомпаній. Це призвело до формування складної екосистеми, до якої увійшли постачальники таких систем і платформ розумного будинку. Крім традиційних постачальників систем повного обслуговування будинку - Crestron, Control4, Gira і Jung - на ринку з'явилися нові великі виробники таких систем - Vivint, ADT, Comcast і Vera Controls в Північній Америці і eQ-3 і SFR в Європі. Комплектне обладнання для систем "розумного будинку" поставляють такі промислові лідери, як Honeywell, Whirlpool, ASSA ABLOY, Somfy, Philips і Sony. Вимога забезпечення взаємозв'язку компонентів системи призвело до появи приладів нового типу і перегляду проектів таких пристроїв, як системи охорони та інтелектуальні термостати. В результаті на ринок вийшли компанії D-Link, Sonos, Belkin, Ecobee, Nest і Numera. Провідні постачальники програмного забезпечення систем розумного будинку - iControl, Alarm.com і AlertMe. Але вони почали відчувати конкуренцію з боку таких великих компаній, як Arris, Amdocs і Technicolor.

Взяти участь в гонці вирішили і постачальники альтернативного програмного забезпечення компанії Apple і Google. У січні 2014 року Google оголосила про укладення угоди на придбання за \$3,2 млрд. компанії Nest Labs [110], яка створила популярний самонавчальний термостат Nest Learning, що автоматично регулює витрату електроенергії в залежності від зміни зовнішніх умов середовища. В кінці року власники термостата вже могли керувати ним за допомогою мовних команд, використовуючи смартфони з ОС iOS або Android. Google також веде розробки домашніх систем охорони [99].

Apple працює над створенням платформи програмного забезпечення для системи управління пристроями "розумного будинку" і веде переговори з групою ритейлерів по використанню її в їх приладах. По гарячих слідах тенденції формування "розумних будинків" компанія на початку 2015 року оголосила про створення системи HomeKit з ОС iOS 8, мову якої розуміють прилади "розумного будинку" будь-якого виробника. Система HomeKit складається з інтелектуального персонального помічника і питально-відповідної системи Siri, що може розуміти природну мову. На виставці побутової електроніки CES 2015 були представлені різноманітні сумісні з HomeKit вироби - від інтелектуальних систем регулювання водопостачання до інтелектуальних дверних замків [80].

Основні вимоги до системи "розумний будинок".

«Розумний будинок» (smart house, далі - SH) - під цим терміном мається на увазі програмно-апаратний комплекс, що дозволяє автоматизувати і спростити управління різними системами, а також іншим обладнанням будинку або квартири. Підсистеми будинку об'єднуються в один єдиний взаємопов'язаний комплекс, що безсумнівно спрощує доступ користувача до управління всією системою (Рис. 1.3.3.1). По суті, SH повинен відповідати способу життя господарів. Він влаштований так, що всі комунальні системи (водопостачання, енергозабезпечення, вентиляція, кондиціонування, опалення

та інші) працюють в комплексі і підпорядковані одній меті - забезпечити мешканцям максимальний комфорт і захищеність. Система SH повинна включати в себе наступні об'єкти автоматизації (Рис. 1.3.3.2).



Рис. 1.3.3.1. Концепція SH.



Рис.1.3.3.2. Структура системи SH.

- керування всією будинковою інфраструктурою, включаючи освітлення, водопостачання та інше;
- централізоване керування системами безпеки і охоронно-пожежної сигналізації, в тому числі контроль доступу, навантажень і аварійних станів;
- клімат-контроль, включаючи управління системою вентиляції і кондиціонування;
- управління зв'язком і комп'ютерною мережею, включаючи бездротову;
- управління аудіовізуальною і розважальною побутовою технікою;
- засоби медичної допомоги та моніторингу активності в будинку.

Функції загального поточного контролю даних і регулювання системи SH виконує спеціальний автоматизований інтелектуальний електронний блок управління вбудованими мікроконтролерами, які здійснюють збір і обробку даних і бездротовий зв'язок з іншими компонентами системи. При необхідності за допомогою інтелектуального блоку можна опускати жалюзі або штори,

включати кондиціонер або систему опалення, підключати пристрої охоронно-пожежної сигналізації, а в разі крайньої небезпеки повідомляти користувача про аварійну ситуацію і вживати заходів щодо запобігання її наслідків. Пристрої, що необхідні для створення SH, можна об'єднати в кілька груп (Рис. 1.3.3.3).



Рис.1.3.3.3. Групи приладів, що входять до системи SH.

Елементна база системи SH

Виділимо основні компоненти централізованої автоматизованої системи SH, що забезпечують її успішне функціонування. Найбільш важливі і необхідні компоненти системи: керуюча програмно-апаратна платформа, виконавчі модулі мікроконтролера, різні датчики (температури, освітлення, вологості), засоби охоронно-пожежної сигналізації.

Програмно-апаратна платформа управління системою

Перспективним електронним блоком управління системою SH вважається програмно-апаратна платформа під торговою маркою Arduino - електронна плата з вільним програмним забезпеченням, виконана на основі зручних у використанні апаратних обчислювальних пристроїв і інтегрованого середовища розробки (Integrated Development Environment, IDE). Базуються платформи Arduino на мікроконтролерах AVR компанії Atmel - ATmega328, ATmega168, ATmega2560, ATmega32U4 з тактовою частотою 16 або 8 МГц. У попередніх версіях застосовувалися мікроконтролери ATmega8, ATmega1280 і інші. Більшість плат програмується через USB [64] (Рис. 1.3.3.4).

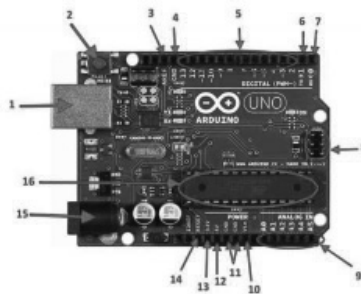


Рис. 1.3.3.4. Плата мікроконтролеру:

- 1) порт для перепрошивання (стандартний USB);
- 2) кнопка апаратного скидання;
- 3) сигнал опорної напруги;
- 4) GND;
- 5) контакти для цифрових сигналів;
- 6) сигнал TX;
- 7) сигнал PX;
- 8) порт для підключення зовнішнього програматору;
- 9) контакти для аналогових сигналів;
- 10) підключення зовнішнього живлення;
- 11) GND;
- 12) +5 В;
- 13) +3,3 В;
- 14) сигнал скидання;
- 15) роз'єм для джерела живлення;
- 16) мікроконтролер.

Особливість Arduino - можливість встановлення спеціальних додаткових плат розширення, так званих шілдів (shields), подібно верствам бутерброда поверх платформи Arduino для додання їй нових можливостей. Так, існують плати розширення для підключення до локальної мережі (Ethernet Shield), для управління потужними моторами (Motor Shield), для отримання координат і часу зі супутників GPS (модуль GPS) і багато інших [60].

Концепція програмування платформи Arduino

Програмування платформи здійснюється через власну спеціальну інтегроване середовище розробки IDE (Рис. 1.3.3.5), що дозволяє складати керуючі програми (скетчі) для плати. В IDE входять текстовий редактор програмного коду, компілятор і модуль для установки нових прошивок плати. Для завантаження програм і встановлення зв'язку середовище розробки підключається до апаратної частини Arduino. IDE написана на мові Java на основі проекту Processing, працює під ОС Windows, Mac OS X і Linux. Мова програмування Arduino - C ++. Програми, написані за допомогою Arduino, обробляються препроцесором, а потім компілюються.



Рис. 1.3.3.5. Оболонка для програмування Arduino.

Виконавчий модуль мікроконтролеру

Виконавчі блоки системи SH в основному побудовані на основі мікроконтролерів AVR з RISC-архітектурою, що працюють узгоджено з дискретним повторювачем або бездротовою системою зв'язку на кристалі [61]. Залежно від побудови системи, складності програмно-апаратного забезпечення і необхідного форм-фактору ARM-мікроконтролер може бути 8- або 32-розрядним. Більш широкі внутрішні шини і досконалі периферійні пристрої 32-розрядного мікроконтролеру забезпечують більш високу швидкість системи управління в порівнянні з 8-розрядним контролером. Для програм з робочим циклом "прискорено працюй, а потім спи" (Run Fast, Then Sleep, RFTS), застосування 32-розрядного пристрою забезпечує більш короткий робочий період і підвищену енергоефективність (Рис. 1.3.3.5). Модулі мікроконтролеру

повинні стояти в кожній кімнаті, працювати від напруги живлення 7-40 В і забезпечувати можливість підключення різних датчиків і кнопок типу "сухий контакт" (герконів, кнопок, датчиків руху). Мікроконтролери, що керують зовнішніми пристроями, здатні виконувати найпростіші дії без участі сервера (включати світло по кнопці, за даними датчика руху, геркону).

Підсистема клімат-контролю

Підсистема управління кліматом підтримує і змінює значення температури і вологості, а також контролює циркуляцію свіжого повітря зовні і всередині приміщення. Функції клімат-контролю системи SH виконуються за допомогою трьох систем: опалення, вентиляції і кондиціонування. Ці інженерні системи в результаті інтеграції працюють як єдине ціле, створюючи комфортні кліматичні умови в приміщенні. Роботу системи забезпечує набір різноманітних датчиків, що фіксують поточний стан мікроклімату в будинку (температуру, вологість, наявність диму, склад і швидкість руху повітря, а при необхідності - освітленості, інсоляції, стану вікон і дверей і т.д.), А також засоби управління - перемикачі та дисплейні панелі. Останні застосовуються для встановлення необхідного режиму роботи кліматичних систем і відображення поточних показників.

Можливе індивідуальне регулювання мікроклімату для кожного мешканця будинку, за рахунок умовного поділу приміщення на зони, для кожної з яких задається необхідний режим (сценарій) і встановлюються відповідні значення керованих і контрольованих параметрів. При необхідності задається порядок зміни параметрів у часі. Той чи інший сценарій, що виконується підсистемою клімат-контролю для кожної зони вибирається автоматично (наприклад, відповідно до дати, дня тижня, часу доби, сезону) або вручну. Крім того, підсистема управління кліматом може контролювати наявність в середовищі шкідливих і небезпечних реагентів: задимленості, чадного газу. Нарешті, підсистема повинна стежити за станом обладнання - кондиціонерів, котлів, засобів опалення та вентиляції.

Дані численних датчиків (наприклад, температури теплоносія в системі опалення, потужності, споживаної тим чи іншим сегментом теплої підлоги, стану вентиляторів і шибєрів вентиляційної системи, режимів роботи кондиціонерів) передаються підключеним до них локальним контролерам (регуляторам). Об'єкти управління системи клімат-контролю - інженерні системи, механізми і пристрої, що впливають на кліматичні параметри. Основні характеристики об'єктів з точки зору управління - коефіцієнт передачі, інерційність і запізнювання передачі даних.

Підсистема охоронно-пожежної сигналізації.

Підсистема охоронно-пожежної сигналізації - складний комплекс технічних засобів, службовців для своєчасного виявлення спалаху і запобігання несанкціонованого проникнення в охоронювану зону. Як правило, пристрій охоронно-пожежної сигналізації входить в комплекс, який об'єднує системи

безпеки і інженерні системи будівлі, забезпечуючи достовірною адресною інформацією засоби оповіщення, пожежогасіння, димовидалення, контролю доступу та ін. Кожен пристрій охоронно-пожежної сигналізації обладнано сповіщувачами, контролюючими різні фізичні параметри середовища. Широко використовуються пасивні і активні ІЧ-сповіщувачі, прилади сповіщення про розбите скло, комбіновані активні пристрої. У системах пожежної сигналізації застосовуються теплові, димові, світлові, іонізаційні, комбіновані і ручні сповіщувачі. Залежно від способів виявлення тривоги і формування сигналів сповіщувачі і системи охоронно-пожежної сигналізації поділяють на наступні типи:

- неадресні з фіксованим порогом чутливості. Сповіщувачі цього типу входять в загальний шлейф пристрою охоронно-пожежної сигналізації. В результаті при спрацьовуванні одного з приладів формується узагальнений сигнал тривоги;
- адресні, дані яких містять адресу приладу охоронно-пожежної сигналізації, що дозволяє визначити зону пожежі з точністю до місця знаходження сповіщувача;
- адресно-аналогові - найбільш інформативні та досконалі. Поточні значення контрольованого параметру разом з адресою сповіщувач передає по шлейфу охоронно-пожежної сигналізації. Такі пристрої використовуються для раннього виявлення тривожної ситуації, отримання даних про необхідність технічного обслуговування приладів внаслідок їх забруднення або інших чинників. Крім того, адресно-аналогові системи дозволяють, не перериваючи роботу підсистеми охоронно-пожежної сигналізації, програмно змінювати фіксований поріг чутливості сповіщувача при необхідності його адаптації до умов експлуатації.

Підсистема регулювання освітлення

Підсистема створює умови комфортного освітлення різних приміщень СН, виходячи з зовнішньої освітленості, потреб користувачів і інших параметрів. Модуль регулювання освітленості СН враховує типові значення освітленості при різних умовах (Табл. 1.3.3.1).

Таблиця 1.3.3.1.

Типові значення освітленості приміщення в залежності від умов середовища.

Умова навколишнього середовища	Рівень освітленості, лк
Ніч	0,001–0,02
Світло місяця	0,02–0,3
Хмарно	5-50
Купчасті хмари	50-500
Слабке сонячне освітлення	100-1000
Сонячне освітлення в літній день	10 ⁶
Освітлення, необхідне при читанні книг і документів	50-60
Освітлення при відтворенні відеозапису	1400

Система регулювання освітлення розрахована на роботу з економічними джерелами світла, в основному переважно напівпровідниковими світлодіодами.

Підключення датчиків до платформи управління системою.

Поняття розумного пилу (smartdust) ввів Крістофер Пістер з Каліфорнійського університету Берклі в 2001 році. Не зовсім звичним для високих технологій терміном «розумний пил» називаються мініатюрні сенсори, які мають можливості обчислень і бездротового зв'язку, а також пам'яттю для зберігання даних і чутливими елементами для вимірювання параметрів навколишнього середовища. Передбачається, що базові елементи «розумного пилу» - моти (англ. mote - порошок), в результаті будуть розміром с зернятка піску або навіть частки пилу.

Кожен мот має власні сенсори, обчислювальний вузол, комунікацію і живлення. Групуючись разом, моти автоматично створюють дуже гнучкі мережі з малим споживанням живлення. Области їх застосування можуть варіюватися від систем управління кліматом до пристроїв для розваги, взаємодіючих з іншими інформаційними пристроями.

«Розумна пил» прекрасно підходить для організації бездротових мереж, в яких вузли зв'язуються один з одним при необхідності. Така мережа має розподілені обчислювальні можливості. Смуга пропускання мережі зростає з ростом її розмірів. Крім власне сенсорів, сенсорні мережі включають в себе і деяку кількість «шлюзів». Останні потрібні для того, щоб збирати, обробляти і направляти далі інформацію з навколишніх їх сенсорів. На першому етапі розвитку концепції «розумного пилу», творці сенсорів посилено прагнули до зменшення їх розмірів. Однак, досвід їх впровадження показав, що мініатюризація не завжди вітається в промисловості. Тому перші зразки «розумного пилу», що були створені корпорацією Intel, представляли собою плати розміром 3 × 3 см.

Зараз створюється друге покоління сенсорів «розумного пилу». В їх основі - 32-бітний процесор XScale, а для стиснення інформації використовується спеціальний процесор, так само як і для забезпечення безпеки. Розміри нових сенсорів менше, ніж у попереднього покоління майже в два рази. У нових сенсорах - велика RAM і FLASH-пам'ять і вони можуть працювати на основі операційної системи Linux. Крім цього, вони мають високошвидкісними можливостями введення інформації, наприклад, з відеокамер.

Окремий напрямок досліджень - питання енергоживлення. Є, наприклад, проекти живлення сенсорів від сонячних батарей розміром 10×10 см. Досліджуються можливості перетворення вібрації механізмів у електроенергію. За допомогою сенсорів нового покоління планується реалізувати ідею «випереджаючих обчислень».

Для підключення різних датчиків до модуля Arduino використовуються існуючі в модулі цифрові і аналогові порти введення-виводу.

Дротові рішення системи SH

Дротові рішення передбачають з'єднання всіх елементів системи SH, будь то датчики, центральний управляючий пристрій, лампа або вентиль батареї, за допомогою дротів. Перевага дротового рішення - практично необмежений функціонал: система відрізняється високою швидкістю роботи, надійністю і відмінною захищеністю від зовнішніх впливів. Ще одним важливим плюсом є абсолютна незалежність у виборі виробників кінцевого обладнання, оскільки система дозволяє інтегрувати практично будь-які пристрої.

Розглянемо приклади датчиків з дротовим підключенням до керуючого модуля Arduino.

Датчик температури і вологості DHT11.

Цифровий датчик температури і вологості DHT11 (Рис. 1.3.3.6) представляє собою складовий пристрій каліброваного цифрового вихідного сигналу, що відповідає контрольованій температурі і вологості (Рис. 1.3.3.7). Відрізняється високою надійністю і довготривалою стабільністю роботи. Датчик містить резистивний сенсор вологості і компоненти структури з від'ємним температурним коефіцієнтом опору (NTC-структури).

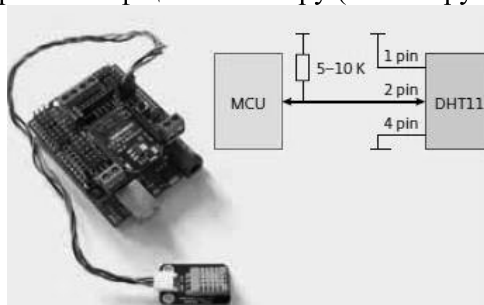


Рис. 1.3.3.6. Модуль Arduino з підключеним датчиком DHT11.

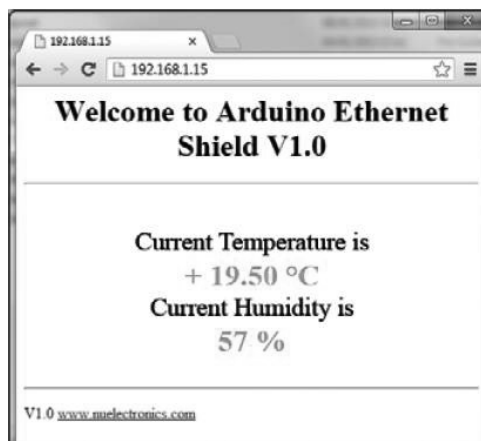


Рис.1.3.3.7. Файл Ethernet екрану системи Arduino з показаннями температури і вологості датчика DHT11.

DHT11 взаємодіє з приймаючою платформою Arduino за власним протоколом:

- мікроконтролер повідомляє датчику про готовність зчитування показань, для чого на деякий час встановлює сигнальну лінію в 0, а потім - в 1;
- датчик підтверджує готовність передати дані, для чого в аналогічному порядку встановлює сигнальну лінію в 0, потім - в 1;
- після цього датчик посилає мікроконтролеру 5 байт даних. Перших два описують температуру, наступних два - вологість, останній - контрольну суму, що підтверджує відсутність помилок зчитування.

Завдяки тому, що датчик виконує вимірювання тільки за запитом, досягається висока енергоефективність: за відсутності комунікацій споживаний струм невеликий.

Датчик має стандартний трьохдротовий інтерфейс, може легко підключатися до пасивної плати розширювача (IO Shield) [34].

Датчик інтенсивності освітлення BH1750.

BH1750 (Рис.1.3.3.8) - мікросхема цифрового датчика зовнішньої освітленості з вбудованим 16-біт АЦП, що підключається до I2C-інтерфейсу. Датчик безпосередньо виводить цифровий сигнал без додаткових розрахунків. Його вихідний сигнал безпосередньо зчитується люксометром. Через інтерфейс I2C можливий вибір адрес двох ведених пристроїв.



Рис.1.3.3.8. Датчик освітленості BH1750.

Датчик придатний для вимірювання інтенсивності фар автомобіля, ліхтарів, фотографічних ламп, цифрових фотоапаратів, настройки ЖК-дисплея і підсвічування клавіатури мобільного телефону [35].

Цифровий датчик руху HC-SR501.

Основне призначення датчиків руху - автоматичне включення або відключення навантаження в заданому інтервалі часу при появі рухомих об'єктів в його зоні чутливості. Він враховує також рівень освітленості.

HC-SR501 - модуль піроелектричного ІЧ-датчика руху (Pyroelectric InfraRed sensor, PIR), зібраного на основі високопродуктивної мікросхеми обробки сигналу датчика BISS0001 з чутливим елементом FIR800 (Рис. 1.3.3.9). При включенні живлення модуля протягом 1-2 с формується зображення кімнати, якщо після цього в приміщенні відбувається рух, спрацьовує пристрій

сигналізації [8]. Модуль може працювати в двох режимах, що встановлюються за допомогою перемички:

- повторного запуску - Н, при якому кожен раз при реєстрації руху вимір відновлюється. При спрацьовуванні датчика кілька разів поспіль на його виході (OUT) зберігається високий логічний рівень;

- одиночного запуску - L: після спрацьовування і реєстрації руху протягом заданого часу датчик відключається, навіть якщо детектор як і раніше реєструє дані і на виході з'являється окремий імпульс. У модулі використовуються два підстроювальних резистора: один - для регулювання чутливості (від 3 до 7 м), інший - завдання часу, протягом якого при виявленні руху на виході буде встановлена логічна одиниця "1" (від 5 до 300 с) (Рис. 1.3.3.9).

Застосування дротової технології завдає певну незручність як проектувальникам систем, так і користувачам, яким доводиться ховати дроти під підлогою, плінтусом або в стінах.

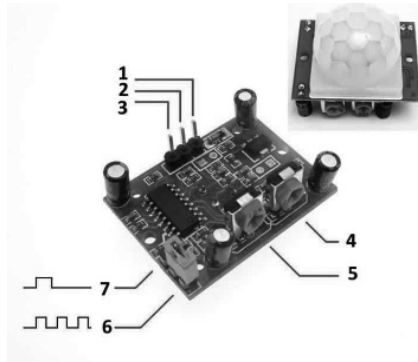


Рис.1.3.3.9. Цифровий ІЧ-датчик руху HC-SR501:

1. Живлення від джерела в діапазоні 5-12 В.
2. Сигнал, що виходить від датчика.
3. GND.
4. Час затримки (утримання логічної одиниці на виході) - від 5 до 300 сек.
5. Чутливість датчика (3-7 метрів).
6. Перемикач в режим «Н» (при серії спрацьовувань встановлюється логічна одиниця).
7. Встановлення режиму «L» (при активації надсилається одиночний імпульс).

Бездротове підключення датчиків

Для створення системи SH з бездротовим підключенням датчиків найкраще підходить система зв'язку стандарту ZigBee. Стандарт орієнтований на створення великих мереж (до 65536 вузлів), що вимагають гарантованої безпечної передачі даних при відносно невеликих швидкостях (250 Кбіт/с) і можливості тривалої роботи пристроїв від автономних джерел живлення (більше одного року від батарей). Серед властивостей ZigBee слід виділити підтримку стандартом складних топологій мереж (рис.1.3.3.10) [104], завдяки якій при відносно малій дальності зв'язку двох сусідніх пристроїв, можна забезпечити більшу зону покриття мережі.

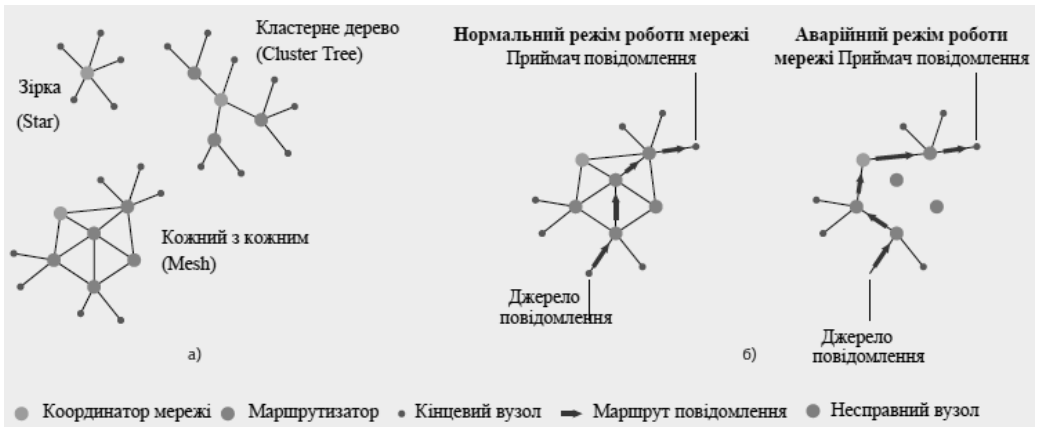


Рис. 1.3.3.10. Топології мереж ZigBee з урахуванням можливості самовідновлення (а) і режими роботи мережі (б).

Інша відмінна риса стандарту - можливість самовідновлення мережі в разі виходу з ладу окремих її вузлів. Ця властивість ґрунтується на тому, що кожен вузол стежить за сусідніми і на основі оцінки потужності прийнятих від них сигналів постійно оновлює маршрутні таблиці. В результаті при зміні просторового розташування сусідів або при виході з мережі одного з пристроїв, обчислюється новий маршрут передачі повідомлення.

Для підключення датчиків до системи моніторингу SH корисно застосовувати спеціальні додаткові розширювальні плати - XBee Shield і модулі XBee компанії Maxstream. Відзначимо, що Maxstream пропонує досить великий перелік модифікацій модулів XBee з різними значеннями вихідної потужності і чутливості приймача, а також з різною конструкцією антени. Вони являють собою досить розвинені пристрої, що працюють під управлінням власного мікроконтролера MC9S08.

В системі SH бездротовий зв'язок, як правило, необхідний коли датчики і виконавчі пристрої знаходяться у віддалених або важкодоступних місцях.

WEB-сервер системи SH

Для віддаленого доступу до інформації системи моніторингу SH її необхідно підключити до WEB-сервера або IP-менеджера. Завдяки поєднанню системи з мережею Інтернет можна контролювати всі системи споживання електроенергії в будинку, встановлювати, де горить світло, яка температура в кімнатах, який прилад знаходиться під напругою, а також фіксувати пересування по будинку. При необхідності за допомогою мобільної телефонії, КПК, iPhone або iPad можна управляти і контролювати всю систему життєзабезпечення житла, навіть перебуваючи на значній відстані від будинку.

Для практичної реалізації недорогого WEB-сервера спільно з модулем Arduino рекомендується використовувати модуль Ethernet Shield (Рис.1.3.3.11).

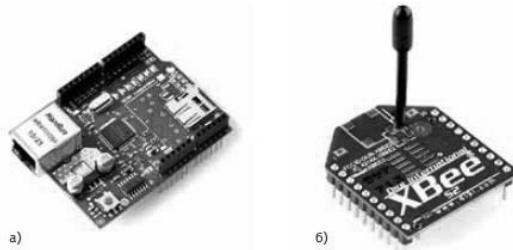


Рис. 1.3.3.11. Модуль Arduino з додатковими платами XBee Shield (a) і Ethernet Shield (б).

Сценарії розумного будинку реалізуються за допомогою експертних систем. Експертні системи - одне з багатьох напрямків в області інноватики та штучного інтелекту, пов'язане зі створенням обчислювальних систем, які вміють самостійно приймати рішення, схожі з думками експертів в заданій предметній галузі. Зазвичай експертні системи розробляються для пошуку рішень задач в спеціалізованих системах вузького спрямування. Експертні системи були першими розробками, які здобули популярність і привернули увагу до галузі штучного інтелекту.

Основними факторами, що впливають на поведінку «розумного будинку» є: час доби, пору року, присутність людини всередині будинку і інформація про кімнату / кімнатах, в якій спостерігається активність. Отримавши ці дані, експертна система, посилаючись на базу правил, оптимально підбере ті чи інші сценарії для роботи своїх складових.

Самоналаштування і самонавчання інтелектуального домашнього простору стає можливим завдяки нейронним мережам. Штучна нейронна мережа - свого роду математична модель, а також її програмне втілення, що побудована за принципом організації та функціонування біологічних нейронних мереж - мереж нервових клітин живого організму. Фактично, нейронна мережа представляє собою систему з'єднаних і взаємодіючих між собою простих процесорів, які зазвичай досить прості, особливо в порівнянні з процесорами, що використовуються в персональних комп'ютерах. Кожен процесор подібної мережі має справу тільки з сигналами, які він періодично отримує, і сигналами, які він періодично посилає іншим процесорам. І, тим не менше, будучи з'єднаними в досить велику мережу з керованою взаємодією, такі локально прості процесори разом здатні виконувати досить складні завдання. Нейронні мережі не програмується в звичному сенсі цього слова, вони навчаються. Можливість навчання - одне з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. В процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. Це означає, що в разі успішного навчання мережа зможе повернути вірний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних і / або «зашумлених», частково перекручених даних.

1.3.4 «Розумні» міста

Загальні підходи

Міста стають епіцентром Всеохоплюючої Інтернету, здатним об'єднати людей, речі, дані і процеси, відкриваючи нові дивовижні можливості. Щодня населення міст збільшується на 180 000 чоловік [19]. Тепер є можливість зробити міста більш безпечними і зручними для життя. Сприяючи зростанню економічної ефективності завдяки скороченню витрат, збільшенню продуктивності праці, створенню нових джерел доходів без підвищення податків, Всеохоплюючий Інтернет здатний протягом наступного десятиліття забезпечити 4,6 трильйона доларів прибутку для організацій державного сектора.

Глобальна урбанізація і пов'язаний з нею безпрецедентний попит на ресурси вимагають нових підходів, включно із застосуванням технології «розумних» міст [43]. І якщо для концептуальної розробки «розумних» міст зроблено чимало, то реальне розгортання «розумних» рішень в містах нерідко наштовхується на бюджетні обмеження і гальмується необхідністю усувати «насушні» проблеми.

При цифровізації міста інформаційні системи і засоби зв'язку повинні відповідати таким критеріям:

- віртуалізація послуг, відсутність необхідності в особистому відвідуванні держустанов;
- збільшення числа призначених для користувача додатків для моментального доступу до послуг, що дозволить громадянам взаємодіяти з держструктурами там і тоді, де і коли їм це зручно;
- розширене впровадження мобільного відеозв'язку в порівнянні з іншими типами передачі даних (мобільна пошта, інтернет-пейджери, веб-сторінки), а також отримання технологічної інформації від різних бездротових датчиків по IP-мережі.

Платформою для надання послуг мешканцям міста, компаніям і держустановам стає мережа Wi-Fi, яка спирається на інфраструктурне обладнання локальних бездротових мереж. Мережа стає безмежною, безшовною і доступною звідусіль, що дозволяє створювати найрізноманітніші сервіси. Мобільність - одна з ключових характеристик, дозволяє забезпечити безперервність зв'язку, особливо в тих випадках, коли неможливо використання проводних мереж передачі даних.

Якщо розробники додатків будуть мати дані, що отримуються за допомогою Всеохоплюючого Інтернету, то «розумні» міста майбутнього зможуть впроваджувати нові сервіси одним натисканням клавіші [9].

В одній тільки Індії в наступному десятилітті планується створити 100 «розумних» метрополій. Ці та сотні інших міст в різних країнах вимагають не тільки будівництва будинків і вулиць, але й пристроїв розгалуженої IP-інфраструктури. Чи зможуть вони собі це дозволити?

Відповідь позитивна по ряду причин.

1. Прокладати комунікації під час будівництва значно дешевше, ніж потім ламати дорожнє покриття і копати кабельні траншеї.

2. Після того, як «розумне» місто почне отримувати дані від мережі пристроїв, підключених до всеосяжного Інтернету, використання цієї інформації для роботи міських сервісів буде коштувати значно менше, ніж можна собі це уявити. Більша її частина взагалі може виявитися безкоштовною.

В основному, це зумовлено винятковою демократичністю розвитку «розумних» міст. Багато перспективних «розумних» міст - такі, наприклад, як Чикаго, - надають розробникам свої дані. Розробники, в свою чергу, створюють такі додатки, які можуть бути використані в будь-якому «розумному» місті світу. Для містобудівників це означає, що надання міських послуг незабаром стане не складніше вибору додатків в магазині. «Ми віддаємо дані безкоштовно, оскільки вони надходять з відкритих джерел, - розповів в Барселоні на Всесвітньому конгресі «розумних» міст Пол Догерті (Paul Doherty), президент і головний виконавчий директор The Digit Group. - Але при цьому кожне місто може створити щось для себе». Таким чином, міська адміністрація, впроваджуючи комплексні міські послуги, не витратить великих коштів на розробку, і це повинно істотно знизити витрати на створення «розумних» міст при одночасному прискоренні розробок.

Важливо і те, що багато що з таких послуг націлене на підвищення ефективності і скорочення витрат. Тому за рахунок утвореної економії вони цілком можуть перейти на самокупність. Адже економія на рівні додатків не виникає просто так. У Європі, наприклад, діє проект Fiware. Його мета - розробка повністю відкритою, публічною архітектури, що не передбачає виплату роялті, і відповідних специфікацій для додатків, що використовуються в «розумних» містах. Організація, що займається його здійсненням, має відділення в 16 містах Європи, а також в Мексиці та Бразилії, і зараз розглядає можливість виходу на північноамериканський ринок. Представляючи Fiware на виставці Smart City Expo, Мауріціо Чеккі (Maurizio Cecchi), що відповідає в компанії Telecom Italia за фінансування науково-дослідної діяльності, сказав: «Ми сприяємо появі нового ринку». Цей ринок обіцяє зробити «розумні» міста доступними всім.

Лідери в сфері містобудування і технологій реалізують інноваційні проекти, щоб міста перестали бути кам'яними джунглями [11]. Завдяки використанню можливостей інформаційно-комунікаційних технологій (ІКТ) перехрестя, зупинки громадського транспорту та основні входи в будівлі перетворилися на місця обміну інформацією. «Інтелектуальні» міста відповідають основним викликам сучасності, в тому числі в сфері сталого розвитку, енергоспоживання, охорони здоров'я, освіти, громадської безпеки та зміни клімату. Багато програм і ініціатив на основі Всеохоплюючої Інтернету вже вирішили ряд проблем, забезпечивши вдосконалені системи обслуговування будівель, зовнішнє освітлення, ефективний рух транспорту, інтерактивні ресурси в школах і містах, динамічне видалення відходів,

дистанційний моніторинг у сфері охорони здоров'я і нові правоохоронні механізми.

Барселона - це одне з найбільш «розумних» міст світу [7], де передові рішення на основі Всеохоплюючої Інтернету вписані в існуючу інфраструктуру: вбудовані датчики на автостоянках, «розумні» автобусні зупинки, Wi-Fi в міському транспорті, вуличні ліхтарі з динамічним керуванням для збереження енергії. Мер Барселони Хав'єр Триас (Xavier Trias) розробив програму «інтелектуального» міста, згідно з якою в наступні десять років столиця Каталонії [11], використовуючи можливості ІоЕ, отримає 3,6 мільярда доларів додаткової вигоди, що перетворить економіку, соціальну сферу та екологію міста. Вже сьогодні мешканці Барселони можуть спілкуватися з представниками державних органів через інформаційні кіоски або мобільні пристрої, попереджати один одного про події або погані дороги, знаходити місця для паркування або інформацію про знижки в магазинах за допомогою мобільних телефонів. Радикальні зміни торкнулися і міського господарства. Наприклад, сміттєві баки, оснащені датчиками, звільняються у міру їх заповнення, а не раз в кілька днів.

Гамбург. Влада Гамбурга просуває стратегію цифровізації [22], що передбачає віртуальне з'єднання міської адміністрації, порту, бізнесу, населення, охорони здоров'я, освіти, громадської безпеки та інших ключових сфер життєдіяльності один з одним. Тепер Гамбург не просто модернізоване «розумне» місто, але й «розумне» і підключене співтовариство (Smart+Connected Community), або, як його ще називають, футуристичне місто на морі (Seatropolis). Інтегрована мережа охоплює установи та організації всього міста, бесшовно поєднуючи людей, процеси, дані і об'єкти єдиним цифровим оверлеєм, накладеним на діючу фізичну інфраструктуру. Продовжуючи розробляти проекти, пов'язані з ІоЕ, увірвався в XXI століття Гамбург вже отримав величезні вигоди від впровадження деяких з них. Так, технологічна модернізація другого за обсягом вантажоперевезень в Європі гамбурзького порту, що займає площу більше 72 кв. км, дала такі результати:

- на 15% скорочено навантаження на транспортні потоки, утворені тисячами судів, легкових і вантажних автомобілів і 85 залізничними коліями;
- на 75% скорочені операційні витрати;
- скорочення кількості серверів з 242 до 48 дозволило на 20% скоротити капіталовкладення.

До 2025 року гамбурзькому порту доведеться подвоїти пропускну здатність до майже 17 млн контейнерів без розширення площ. «Щоб збільшити транспортний потік в порту, ми підключили до системи управління трафіком розвідні мости, - каже Йенс Майєр (Jens Meier), головний виконавчий директор адміністрації гамбурзького порту. - При наближенні судна портал відкривається, а транспортний потік можна направити по іншому маршруту».

«Всеохоплюючий Інтернет - найважливіший фактор реалізації наших подальших планів, - стверджує Себастьян Закса (Sebastian Saxe), головний директор Гамбурзького портового управління по інформатизації. - Нам

необхідно збільшити контейнерну пропускну здатність порту, і головну роль тут відіграють технології».

HafenCity - «місто в місті» на базі Всеохоплюючої Інтернету.

Гамбурзькому керівництву постійно доводиться вирішувати питання, що виникають у зв'язку з успішною реалізацією планів по залученню нових жителів, робітників і туристів (зростання турпотoku становить 30% на рік, вже будується третій термінал для круїзних лайнерів). Крім того, Гамбург реконструює стару, практично невикористану промислову зону, яка лежить навпроти порту, перетворюючи її в «місто в місті» на базі Всеохоплюючої Інтернету. Цей район, що отримав назву HafenCity Hamburg, збільшить площу міста приблизно на 40%, додасть 6 000 жителів, створить 45 000 робочих місць і буде залучати близько 100 000 туристів на день.

Щоб забезпечити ефективність потоку трафіку і полегшити життя водіям, в HafenCity планується установка датчиків на паркувальних місцях. Вони будуть підказувати водіям, завантажившим на свої мобільні пристрої відповідні додатки, де можна припаркуватися. Як вважає Юрген Брюнс-Берентелг (Jurgen Bruns-Berentelg), головний виконавчий директор проекту HafenCity Hamburg, з цими та іншими міськими сервісами на базі Всеохоплюючої Інтернету «Гамбург може надихнути інші німецькі і європейські міста на впровадження аналогічних інновацій та рішуче просування концепції Всеохоплюючої Інтернету і цифровізації бізнесу».

І це лише початок. У Гамбурзі повним ходом реалізуються плани подальшої мережевої експансії в такі сфери, як охорона здоров'я, громадська безпека та освіту. Тут і в інших «розумних» містах на кшталт Копенгагена, Барселони і Чикаго цифрові платформи на базі Всеохоплюючого Інтернету з'єднують торгівлю, виробництво, транспорт і енергетику, допомагаючи людям по-новому жити, працювати, вчитися і розважатися.

Копенгаген. У 2012 році Копенгаген очолив список найбільш екологічно чистих міст (Global Green Economy Index - «Всесвітній індекс зеленої економіки»), а в 2014 році був визнаний «Зеленої столицею Європи». Крім того, Всесвітній Банк назвав Данію найкращою в Європі країною для ведення бізнесу, а журнал Monocle Magazine зарахував Копенгаген до найкомфортніших для проживання міст світу. А ще це місто очолює безліч світових рейтингів як «найщасливішої столиці» [6].

В кінці травня 2014 року Cisco, а також адміністрації Копенгагена і прилеглих до нього комун Альбертслунн і Фредерикссуне уклали партнерські угоди з впровадження програми Всеохоплюючої Інтернету. За допомогою цифрових технологій ці угоди покликані об'єднати в єдине ціле всі міське господарство, починаючи від освітлення і парковок і закінчуючи енергосистемами і об'єктами цивільного призначення.

Копенгаген вступає в епоху цифрової революції ІоЕ слідом за Барселаною, Амстердамом, Гамбургом, Чикаго, Далласом, Сан-Франциско, Сонгдо, Ніццою і Канзас-Сіті. Можливість підключати різні додатки до загальноміської мережевої платформи відкриває нові перспективи для

забезпечення роботи сучасних муніципальних служб, що дозволяє кардинально поліпшити соціальну, екологічну та економічну ситуацію в місцевих громадах.

У Альбертслунне створюється Датська дослідна лабораторія зовнішнього освітлення (Danish Outdoor Light Lab, DOLL), покликана продемонструвати працездатність системи «розумного» освітлення. Близько 40 конкуруючих рішень для зовнішнього освітлення, об'єднані в одну відкриту мережу, забезпечать наймовірні можливості для скорочення фінансових витрат і обсягів споживання електроенергії при одночасному підвищенні громадської безпеки.

В оточеному зеленими полями місті Виндж (40 км від центру Копенгагена в комуні Фредерикссунне) розроблений один з найбільш екологічних та інноваційних в Європі генеральних планів розвитку, в основу якого покладено повна відсутність викидів вуглекислого газу в атмосферу. Для досягнення цієї мети прийнято ряд планів щодо формування інфраструктури, орієнтованої на додатки. Вона зв'яже воедино людей, дані, процеси і фізичні об'єкти - відмінний приклад Всеохоплюючого Інтернету в дії.

А в самому Копенгагені в одну систему будуть об'єднані «розумне» освітлення, паркування, управління водопостачанням, «розумні» електромережі та багато іншого. При цьому повсюдно будуть використовуватися датчики, що поліпшить такі показники, як довговічність і відмовостійкість різних пристроїв і систем, а також якість життя.

Влада Копенгагена і його передмість заявили про намір повністю позбутися від викидів вуглекислого газу до 2025 року. Вирішальну роль в поліпшенні стану навколишнього середовища і в побудові «зеленої» економіки повинні зіграти нові технології.

Нью-Йорк. У Нью-Йорку запущена інтерактивна платформа, яка збирає корисну інформацію, що надходить з відкритих державних джерел, від місцевих підприємств і громадян [11]. Встановлені на автобусних зупинках і залізничних вокзалах «інтелектуальні екрани» об'єднують сенсорну, мовну та аудіо-технології, надаючи широкий спектр локальної інформації, включаючи оповіщення служби громадської безпеки. Завдяки «інтелектуальним екранам» місцеві поліція і пожежна служба мають загальноміську мережу збору інформації, зв'язку і реагування, що дозволяє направляти співробітників саме туди, де потрібна допомога.

Казань. «Майбутнє створюється сьогодні, - вважає Євген Морозов. - Нас уже оточують «розумні» винаходи, здатні змінити наше життя [52]. Причому мова йде не тільки про саме підключене місто Європи - Барселону. В кінці літа, в День міста Казань, в столиці Татарстану був запущений пілотний проект в рамках створення комплексного рішення «Розумне" і безпечне місто Казань»[5]. Слідом за цим в цілях подальшої реалізації цього проекту компанія Cisco підписала дорожню карту щодо взаємодії з Республікою Татарстан [16]». Ну і як логічне продовження 20-22 листопада 2014 року в Казани був проведений перший всеросійський хакатон «Розумне місто» [32].

1.3.5 Нафтогазова галузь

Застосування IoT в нафтогазовій галузі - один з головних потенційних проривів для промисловості [43]. Більш ніж дворазове падіння ціни на нафту з червня 2014 р стало основною темою порядку денного керівників промислових підприємств, і воно ж дає шанс інноваційним компаніям вирватися вперед, застосовуючи нові технології для зниження кінцевої вартості видобутку, переробки і доставки.

Галузеві експерти згодні з тим, що отримання інформації на основі даних є ключовим фактором підвищення операційної ефективності, а аналіз даних - каталізатором № 1 для прискорення прийняття рішень і підвищення їх якості. На сьогодні визначено три головних бізнес-переваги такого підходу:

- прискорення вирішення проблем,
- поліпшення управління процесами,
- вдосконалення техніки безпеки на виробництві.

1.3.6 Телекомунікації. ІоЕ. ІоТ

Cisco уважно аналізує свої власні внутрішні процеси [44]. Їх оптимізація вже дозволила заощадити близько 2 млрд доларів. Cisco всіляко заохочує професійний розвиток своїх інженерів. В компанії був створений інноваційний хаб Think Q baterubator (від слів think - «думати» і «інкубатор»). Його обладнали 3D-принтерами, сенсорами, технікою. Thinkubator дає можливість будь-якому співробітнику компанії заглянути в майбутнє, створювати його власними руками, обмінюватися ідеями з колегами. Cisco заохочує творчі думки не тільки всередині власної компанії, адже інновації неможливо створювати на самоті. Cisco відкриває спільні інноваційні хаби спільно зі своїми партнерами і клієнтам і замовниками, що дозволяє виробляти не тільки те, що цікаво їм самим, але й те, що корисно і необхідно клієнтам. Cisco не боїться ділитися своїми розробками і знаннями: врешті-решт, 19 трлн доларів - така величезна сума, яку жодна компанія не зможе освоїти поодиночі. Справ вистачить на всіх.

Новий світ стане світом, заснованим на прототипуванні і спільної розробці та впровадженні інновацій. Якщо ти створюєш щось нове, у тебе приблизно 5% на успіх. Якщо ти робиш 30 нових речей, можливість успіху зростає до 50%, і так далі. Зрозуміло, що з 50 нових розробок 45-46 будуть провалом просто за законом ймовірності. Тому принцип прототипування і, готовність швидко запускати і припиняти розробку, відсутність страху провалу отримують особливу важливість в цьому новому світі. Cisco щорічно витрачає 6 млрд доларів на дослідження і розвиток розробки - це більше, ніж загальна сукупність суми аналогічних витрат п'яти найближчих конкурентів компанії.

1.3.7 Медіабізнес

Розвиток високошвидкісних мереж доступу в інтернет і лавиноподібне зростання кількості «розумних» пристроїв призводять до еволюції способів доставки контенту споживачеві [10]. Якщо ще 3-5 років тому компанії прагнули розробляти власні програми, то тепер ринок диктує нові правила: фокус змістився в бік більш глибокої інтеграції сервісів і інформації в месенджери з їх чат-ботами. Споживач вже не хоче встановлювати ваш додаток - він хоче отримувати актуальну інформацію саме через той канал, в якому він проводить найбільше часу.

З іншого боку, ІоЕ неминуче веде до необхідності збирати, накопичувати і аналізувати колосальні обсяги персональних даних з «розумних» пристроїв. Для медіа-бізнесу це означатиме перетворення самого контенту: він стане набагато більш персоналізованим і, ймовірно, навіть кастомізуємим для певних груп споживачів. Не виключено, що ця ніша буде повністю зайнята UGC (User-Generated Content, або призначеним для користувача контентом), який вже сьогодні генерується з неймовірною швидкістю і в найближчій перспективі буде передаватися споживачеві в режимі реального часу. Причому контент буде підбиратися не тільки на підставі історії споживання і сформованих переваг користувача, але і з урахуванням його фізичного стану, навантаження протягом дня - параметрів, які можна буде відстежувати онлайн за допомогою «розумних» носимих пристроїв. Наприклад, після важкого дня користувачеві запропонують подивитися легку комедію, а «розумний дім» сам запустить сценарій для максимального розслаблення.

Доступ до даних про глядачів також відкриє можливості для розвитку інструментів аналітики даних: пильне вивчення різних сегментів користувачів дозволить виробити алгоритми для залучення і нацелювання аудиторії. Майбутнє за тими гравцями медіаринку, які навчатися отримувати і аналізувати всі можливі дані про поведінку користувача, виявляти тренди і правильно їх інтерпретувати. Алгоритми вже зараз стають більш витонченими, дозволяючи глибоко аналізувати реакцію аудиторії на контент. Якщо ці дані поєднати з можливостями роботів-журналістів, ми отримаємо автономний ЗМІ, яке майже не вимагає участі людини.

1.3.8 Роздрібна торгівля

Нові можливості ІоЕ, відкривають великі перспективи для трансформації бізнесу, що особливо актуально для роздрібною торгівлі [55]. ІоЕ змушує ритейлерів перебудовувати бізнес. Стрімке зростання підключених пристроїв вимагає нової парадигми - роздрібна торгівля, готова до використання ІоЕ (IoE-Ready Retailer). Ця бізнес-модель значно поліпшить якість обслуговування, підвищить продуктивність праці і ефективність ланцюжка поставок, адже ритейлери зможуть розуміти своїх покупців як ніколи раніше.

Згідно з оцінкою Cisco, протягом наступних десяти років роздрібна торгівля отримає 11% сукупної потенційного прибутку від ІоЕ, поступившись лише обробної промисловості. Cisco вважає, що успіх ритейлерів буде залежати, перш за все, від їх здатності застосовувати технологію для поліпшення «особистісних» і «технологічних» аспектів своєї діяльності і пропонувати покупцям нові підключення послуги.

За даними досліджень Cisco, сучасні споживачі оцінюють, порівнюють і купують товари одним клацанням миші. Чисельність же цифрового покоління людей збільшується з неймовірною швидкістю (на 60% щорічно). Це дає продавцям величезний обсяг цінної інформації, заснованої на оперативних даних, допомагаючи підтримувати запас потрібних товарів і по-новому пропонувати їх. Ось лише деякі з переваг використання ІоЕ в роздрібній торгівлі, виявлені в ході недавнього дослідження консалтингового підрозділу Cisco Consulting Services [55]:

- Інтерактивні дисплеї допомагають вивчати і купувати товари;
- Адаптація ритейлерами маркетингової і рекламної стратегії відповідно до запитів покупців збільшує продажі на 3%;
- відеорішення для спільної роботи дозволяють покупцям отримувати особисті консультації фахівців незалежно від місця розташування і розміру магазину або використовуюваного пристрою;
- Мобільні технології, рішення для дистанційної спільної роботи і BYOD («Принеси своє власне пристрій») сприяють підвищенню продуктивності праці і збільшують прибуток на 9%;
- Скорочення браку товару збільшує продажі на 1%;
- Бездротові технології забезпечують зв'язок з партнерами, дозволяючи підтримувати запас товарів і уникати провалів в поставках;
- Мобільні технології для підключених пристроїв доставляють персоналізовану маркетингову інформацію і інформацію про товар на мобільні пристрої покупців;
- Тільки інноваційні платіжні рішення здатні збільшити прибутковість на 1-2%;
- Скориставшись переваги ІоЕ, звичайний ритейлер зможе за 2 роки збільшити свій прибуток на 12%.

5 основних чинників ефективності ІоЕ, які діють у всіх сегментах. Розглянемо їх стосовно роздрібною торгівлі.

1. Обслуговування покупців. У минулому взаємодія продавців з покупцями відрізнялася знеособленістю і єдиним підходом до всіх. З появою технологічних нововведень на базі ІоЕ, можна в режимі реального часу формувати вузькоспеціалізовані пропозиції, виходячи з місця розташування покупця і конкретних умов. Це дозволяє будувати довірчі відносини з покупцем, де б він не знаходився - вдома, в магазині або де-небудь ще.

2. Інновації. Раніше ритейлери визначали свою збутову політику, керуючись неточними даними продажів і власними інтуїтивними міркуваннями

щодо найбільш продаваного товару. Завдяки прогнозній аналітиці, яка спирається на поведінку окремих покупців, а не на загальні демографічні дані, ІоЕ відкриває можливості для більш глибокого рівня інновацій. Пропозиції, які бачить покупець, відображають його справжні бажання, потреби і переваги.

3. Продуктивність праці. Рітейлер, що користується технологіями ІоЕ, приймає кадрові рішення на динамічній основі, керуючись схемами переміщення покупців в магазині і детальними даними про клієнтів. Результат - підвищення якості обслуговування і скорочення витрат.

4. Ланцюжок поставок. Найчастіше покупці скаржаться на відсутність товару. Сучасні програми ІоЕ (наприклад, Endless Aisle) дозволяють покупцям вибирати з необмеженого асортименту товарів, що знаходиться не тільки в магазині, але і на складі, який тепер доступний в режимі онлайн.

5. Контрольоване використання ресурсів. Високі операційні витрати багато в чому обумовлені постійним використанням систем освітлення, опалення та інших допоміжних засобів. В умовах ІоЕ ритейлер застосовує схеми енергоспоживання, що адаптуються відповідно до потоком покупців і виробничими потребами конкретного магазину.

1.3.9 Сфера послуг

Компанія Uber [44]. Персонал - близько тисячі молодих людей. Не має власних активів. Але надуспішно використовує нові технології, щоб повністю модернізувати, підірвати індустрію таксі. І зараз приблизна ринкова вартість цієї компанії, у якої немає реальних, дійсних активів, - 44 млрд доларів. Для порівняння: компанії United Airlines, UPS, Hertz, з усіма їх авіалайнерами, нерухомістю і автомобілями, в сукупності не володіють такою ринковою вартістю, як Uber.

1.3.10 Охорона здоров'я

У сфері охорони здоров'я ІоЕ дозволив лікарям проводити персоналізоване лікування, отримуючи відомості про пацієнта в реальному часі і надаючи послуги дистанційно [11]. У 2008 році віддалені села провінції Сичуань (Китай) були зруйновані в результаті сильного землетрусу. Через відсутність великих медичних центрів постраждалим доводилося витратити на дорогу години і навіть дні, щоб проконсультуватися з лікарем. Тим часом багато хто з них не могли кинути роботу і сім'ї. Мережеві технології дозволили місцевим поліклінікам обмінюватися даними і взаємодіяти з колегами з міських лікарень, використовуючи систему телеприсутності. І тепер пацієнти в провінції Сичуань можуть отримувати дистанційні консультації та лікування, економлячи на дорожніх витратах.

1.4 ІоЕ в навчальних програмах

«Працювати в сфері ІТ не тільки цікаво, але й перспективно [52]. За даними Світового банку, в найближчі десять років кількість вакансій, пов'язаних з інформаційно-комунікаційними технологіями, складе в усьому світі два мільйони, а щоб підключити непідключене, в період з 2014-го по 2022-й рік щорічно буде вимагатися близько 220 тисяч нових інженерів. Всеохоплюючий Інтернет докорінно змінить наше життя, і вчитися жити в цьому новому світі потрібно вже сьогодні. ».

(Керівник програми Мережевої академії Cisco в Росії Дмитро Разумовський).

В умовах загальної цифрофізації першочерговою необхідністю є розвиток галузі інформаційних технологій [54]. Для реалізації даної мети важливу роль відіграє підготовка висококваліфікованих ІТ-фахівців. Наприклад, в Казахстані запущена програма Country Digitization Acceleration (прискорення цифровізації країни). Компанія Cisco недавно ввела в свою програму Мережевої академії вступний курс по всеосяжного Інтернету [52]. Іншою формою навчання та обміну досвідом є проведення тематичних форумів ІоТ, ІоЕ.

Щоб прискорити процес формування Інтернету речей [66], Cisco провело Перший Всесвітній форум, присвячений цій темі. Internet of Things World Forum зібрав теоретиків і практиків бізнесу, представників державних установ та навчальних закладів. На форумі обмінялися думками лідери в різних галузях і новатори, які прагнуть створити систему спільної роботи, щоб тим самим стимулювати повсюдне поширення інновацій.

Перед Всесвітнім форумом з Інтернету речей поставлені такі завдання:

- вирішення соціальних, політичних, технологічних і природоохоронних питань, пов'язаних з Інтернетом речей;
- популяризація міжгалузевих, міждисциплінарних інновацій та передового досвіду;
- визначення необхідних стандартів і рішень для підтримки безпеки і загальних правил (політик), задоволення нормативних вимог і забезпечення сумісності;
- рішення загальних міжгалузевих питань і завдань і формування єдиної технологічної платформи для Інтернету речей;
- прискорена підготовка ринку до Інтернету речей.

Висновки до розділу 1.

1. В роботі надані визначення, встановлений зв'язок із суміжними поняттями, прослідкована історія розвитку, виявлені основні тенденції та сформовані прогнози подальшого розвитку **Інтернету Всього (ІоЕ)**.

2. Важливим аспектом ІоЕ, який відрізняє ІоЕ від ІоТ, є "мережевий ефект". ІоТ - лише складова частина ІоЕ. ІоЕ ґрунтується на зв'язках між людьми, процесами, даними і речами, причому мова не йде про кожен з цих вимірів окремо. Кожен підсилює можливості трьох інших, і справжня сила ІоЕ проявляється на їх перетині.

3. Перспективи розвитку Інтернету.

- ІоЕ вдихне життя в усе, що тільки можна собі уявити, підключаючи неживий світ до Інтернету..
- Всеохоплюючий Інтернет буде сприяти розвитку економіки.
- Всеохоплюючий Інтернет стане більш персональним і прогнозованим.
- Всеохоплюючий Інтернет сприятиме схрещуванню технологій з гуманітарними науками.

4. За оцінками компанії Cisco, 99% фізичних об'єктів, які можуть стати частиною ІоЕ, до Мережі поки що не підключені. У наступному десятилітті завдяки підключенню того, що ще не підключено, ІоЕ може забезпечити приватному сектору світової економіки прибуток, що обчислюється 14,4 трлн. доларів США.

5. Три ключові чинники, що сприяють прийняттю ІоЕ бізнесом: зростання темпів інновацій (36% опитаних), необхідність задовольнити зацікавленість споживачів в нових методах взаємодії (35%), необхідність автоматизувати бізнес-процеси (35%).

6. Головні технологічні фактори впровадження ІоЕ: нові типи пристроїв (33%), зростаючий обсяг даних (32%), хмарні технології (31%).

7. Щоб отримати максимальну користь з економіки ІоЕ, компанії повинні взяти на озброєння стратегію, яка передбачає:

- інвестування в високоякісні технологічні інфраструктури та інструменти,
- повсюдне впровадження методів бізнесу, що дозволяють всім співробітникам вносити особистий вклад у загальну справу,
- розробку ефективних методів управління інформацією.

8. Головним фактором, що визначає ефективність використання можливостей ІоЕ, є якість технологічної інфраструктури та відповідних інструментів.

9. Компаніям слід зосередитися на тих можливостях ІоЕ, які найбільш значимі для їх галузей.

Для виробничих підприємств це - багатовимірний аналіз даних в реальному часі, спільна робота з інтеграцією відеозв'язку, віддалений контроль фізичних активів.

Для енергетичних компаній - інтеграція даних лічильників, можливість знаходити потрібних фахівців, попереджуючий аналіз.

Для компаній роздрібною торгівлі - попереджуючий аналіз і візуалізація даних, BYOD (Bring Your Own Device), взаємодія з замовниками шляхом застосування мультимедійних технологій, мобільні платежі і віддалений моніторинг споживачів.

10. Фірми і компанії середнього розміру з країн з ринком, що формується, мають відмінні і при цьому зростаючі шанси скласти гідну конкуренцію нинішнім лідерам.

11. Для компаній важливо не упустити момент зміни технологій. В результаті цифрової революції в найближчі п'ять років в 12 галузях зникне 40% компаній. Що стосується ТОП-500 світових корпорацій, то тут в найближчі десять років очікується зникнення 35% компаній, які пропустять або не впишуться в цей цифровий перехід.

12. В першу чергу, необхідно забезпечити інформаційну безпеку.

13. Зміни в світі цифрових технологій відбуваються все швидше. Невиробничі компанії: раніше - 3-5 років, тепер - 1-1,5 років. Промисловість: раніше - 10-12 років, тепер - 2 роки.

14. За консервативними оцінками, в світовому масштабі потенційна економічна вигода ІоЕ в найближче десятиліття може скласти \$ 19 трлн. З них: \$ 14,4 трлн - приватний бізнес, \$ 4,6 трлн - держсектор.

15. Основні можливості для реалізації ІоТ вже є: новий інтернет-протокол - IPv6; мікрочіпи з наднизьким споживанням енергії; бездротові модулі.

16. Чотири технологічні опори ІоЕ: мобільні технології, хмарні технології, великі дані, речі.

17. Основні сфери застосування ІоЕ в промисловості: «розумне» місто, транспорт, важка і легка промисловість, будівництво, енергетика, безпека, медицина, ритейл; в повсякденному житті: носимі пристрої, "розумний будинок.

18. Сьогодні ми маємо близько 50 млрд. підключених пристроїв: транспорт - 500 млн пристроїв (аварійне реагування, самодіагностика автомобіля та ін.); розумне місто - 2 млрд пристроїв (розумне освітлення, розумні парковки, інтелектуальний транспорт, розумні будівлі і т.д.); безпека - 100 млн пристроїв (біометрія, сканери, контроль водних ресурсів, моніторинг навколишнього середовища та ін.); електронна охорона здоров'я - 100 млн пристроїв (мобільні лабораторії, хірургічне обладнання, віддалений моніторинг, телемедицина та багато інших); промисловість - 1 млрд пристроїв (управління активами, розумні насоси, компресори, клапани, безпека (Ias), smart-мітки (nfc) і ін.); енергетика і жкг - 500 млн - 1 млрд пристроїв (розумні лічильники, польові мережі і т.д.); торгівля - 200 млн пристроїв (заправні станції, ігри, соціальні події, автомати з їжею (вендинг), smart-мітки і т.п.)

2. ВБУДОВАНІ ТА МОБІЛЬНІ СИСТЕМИ МОНІТОРИНГУ НЕБЕЗПЕЧНИХ СИТУАЦІЙ ЯК СКЛАДОВА ІОЕ

2.1 Програмне забезпечення виявлення небезпечних для користувачів ситуацій на основі аналізу даних вбудованих сенсорів мобільних пристроїв

В даному підрозділі розглянуті результати розробки системи для виявлення та попередження небезпечних для користувачів ситуацій на основі аналізу даних вбудованих сенсорів мобільних пристроїв. Було розроблено серверне програмне забезпечення для аналізу даних ІоЕ, веб-інтерфейс для керування системою, та клієнтський мобільний додаток для збору і відправки даних з сенсорів та сповіщення про можливі небезпечні ситуації. Також був визначений та реалізований алгоритм виявлення небезпечних ситуацій для користувача смартфона. Були вдосконалені підходи щодо збільшення частоти зняття даних з сенсорів та обходження обмежень діапазонів припустимих величин, які вимірюють сенсори, поліпшена якість отриманих значень з сенсорів смартфона.

Результати роботи можуть бути використані в будь-яких аналітичних системах ІоЕ, які потребують збільшення точності отримання даних та обходження обмежень діапазонів можливих значень. Серверна частина розроблялася з використанням мови програмування Java та інтерфейсу Java Servlet API. Для реалізації клієнтської частини були використані наступні технології: HTML, CSS, JavaScript, AJAX, JSP. Для зберігання інформації, було обрано систему керування базами даних PostgreSQL.

2.1.1 Вступ

Визначення поточної фізичної активності людини є актуальним напрямом на сьогодні. Ця інформація має широкий діапазон застосувань, таких як відстеження фізичної форми і стану здоров'я користувача та надання контекстно-залежної реклами, що виводиться в мобільних додатках в залежності від попередніх дій користувача. Для виявлення фізичної активності людини використовуються деякі сенсори, які повідомляють про стан зовнішнього середовища. Такі сенсори є в більшості сучасних мобільних пристроїв.

Мобільні пристрої стали частиною повсякденного життя для багатьох людей: користувачі мають та використовують їх протягом дня. Для роботи деяких додатків необхідно встановити поточну фізичну активність користувача. Для цього смартфони оснащені відповідними сенсорами: акселерометр, гіроскоп, датчик освітлення, датчик приближення та GPS-навігатор, які і надають необхідну інформацію.

Компанія Google розробила інструмент для виявлення поточної активності та положення користувача, такі як стояння на місці, ходіння, біг, знаходження в автомобілі чи на велосипеді та інші. Інструмент автоматично визначає активність, періодично зчитуючи значення сенсорів і обробляючи їх за допомогою моделей машинного навчання. Щоб оптимізувати ресурси програмний інтерфейс може призупинити надання інформації про діяльність користувача (якщо пристрій протягом деякого часу знаходиться в нерухомості) та продовжити роботу (коли він виявляє рух) [58].

Для реабілітації та діагностики стану здоров'я також важливе розуміння активності та рухів пацієнтів. Існує потреба ідентифікувати події, які можуть виникнути з людиною або з телефоном, такі як: серцевий напад, втрата телефону та інше. Подібні технології вже існують: вчені з Данії розробили нейронну мережу Corti Signal, яка відстежує найменші сигнали для діагностики серцевого нападу. Планується, що нейронну мережу Corti Signal використовуватимуть в якості помічника диспетчерів служби медичної допомоги. Corti Signal в режимі реального часу аналізує інформацію і буквально за секунди визначає діагноз. Штучний інтелект буде не тільки слухати та аналізувати вербальну інформацію, але й відстежувати дихання потерпілого. Ефективність штучного інтелекту датських вчених вже підтверджують ряд екстрених ситуацій [67].

Отже, для ідентифікації небезпечних подій потрібні методи машинного навчання та інші нетривіальні технології. Тому актуальною стає задача використання будь-яких доступних засобів, зокрема сенсорів мобільних пристроїв, для ефективної ідентифікації ситуацій, в яких опинилась людина або її смартфон. Багато розробок та досліджень були присвячені виявленню певних подій, але не було реалізовано деякої загальної системи, яка ці дані обробляє та надає необхідну інформацію про можливу небезпеку.

Основні невирішені проблеми існуючих розробок полягають в обмеженнях та неточних значеннях, які мобільний додаток отримує з сенсорів

смартфону. Такі фактори, як діапазони можливих значень сенсорів, навантаження на пристрій, занадто великий інтервал часу передачі даних та складова шуму в значеннях сенсорів ускладнюють роботу системи і знижують ймовірність їх коректної обробки. Тому необхідно застосувати певні аналітичні методи для обходження цих обмежень та поліпшити якість отримання даних.

Мета роботи – підвищення якості моніторингу станів користувача, які є загрозовими для його здоров'я за допомогою бюджетних мобільних пристроїв.

Об'єкт дослідження – відстеження поточного стану користувача смартфона на основі аналізу даних, отриманих з вбудованих сенсорів мобільних пристроїв.

Предмет дослідження – методи побудови програмного забезпечення моніторингу станів користувача, які є загрозовими для його здоров'я за допомогою бюджетних мобільних пристроїв.

Завдання, які потрібно виконати для досягнення мети:

1. Задіяти певні аналітичні методи для обходження обмежень діапазону можливих значень;
2. Застосувати чисельні методи для зменшення часового інтервалу отримання даних з сенсорів смартфона;
3. Використати методи статистичної обробки даних для їх фільтрації від шуму.
4. Розробити програмне забезпечення.

Шлях досягнення мети – реалізація системи виявлення небезпечних ситуацій із використанням сучасних досліджень та розробок виявлення деякої фізичної активності користувача смартфона.

Потрібна система, яка може аналізувати дані з сенсорів мобільного пристрою та на їх основі виявляти події, які відбуваються з ним. Для цього потрібно реалізувати програму для мобільного пристрою під керуванням ОС Android, яка буде отримувати та передавати дані з сенсорів, та веб-сервісу, котрий і буде аналізувати ці дані та надавати відповіді. Технологія може бути застосована і під iOS, але для широких верст населення Android є більш доступною за ціновим фактором.

Кінцева ціль даної роботи є реалізація системи, котра буде складатися з двох компонентів – мобільного додатку, який буде надавати інформацію, отриману з сенсорів мобільного пристрою, та веб-сервісу, який ці дані і буде отримувати. Як результат застосунок повинен визначати, чи є певна загроза здоров'ю користувачу смартфона. У разі визначення небезпечної ситуації програма повинна відповідно відреагувати та надати всю необхідну інформацію про подію.

Наукова новизна

1. Вдосконалені підходи щодо збільшення частоти зняття даних з сенсорів за допомогою інтерполяції та апроксимації;
2. Вдосконалені підходи щодо обходження обмежень діапазонів припустимих величин, які вимірюють сенсори за допомогою аналітичної обробки даних;

3. Поліпшена якість отриманих значень з сенсорів смартфона за допомогою використання методів статистичної обробки даних.

Практична значущість роботи

Результати роботи можуть бути використані в будь-яких аналітичних системах, які потребують збільшення точності отримання даних та обходження обмежень діапазонів можливих значень. Прикладами можуть бути такі системи, метою роботи яких є моніторинг та вимірювання змін характеристик певних явищ або подій.

Реалізована система відрізняється від інших отриманням більш об'єктивної інформації про стан користувача смартфона шляхом обробки та аналізу більшої кількості вихідних даних, що надходять з сенсорів мобільного пристрою, використовуючи існуючі розробки та дослідження.

Таким чином, створенні розробки можуть бути застосовані для будь-яких вимірювальних приладах та систем, де точність отриманих даних є головним фактором для роботи з ними.

2.1.2 Аналіз роботи сенсорів та створення прототипу системи

Сенсори, що доступні для мобільних пристроїв, в більшості випадків можна поділити на три категорії: руху, положення та навколишнього середовища [46]. Ці сенсори можуть бути досить різноманітними:

- Акселерометр.
- Гіроскоп.
- GPS-навігатор.
- Датчик освітлення.
- Датчик магнітних полів.
- Барометр.
- Датчик піднесення телефону до голови.
- Датчик температури апарату.
- Датчик температури оточуючого середовища.
- Вимірювач відносної вологості.

У багатьох сучасних мобільних пристроїв вже вбудовані такі базові сенсори, як акселерометр, гіроскоп та GPS-навігатор. В більшості випадків, для визначення характеристик пристрою в просторі, використовуються саме ці сенсори.

Акселерометр вимірює проекцію повного прискорення. Електронні акселерометри часто вбудовуються в мобільні пристрої (зокрема, в смартфони) і застосовуються в якості крокомірів, датчиків для визначення положення в просторі, автоматичного повороту дисплея та інших цілей.

Гіроскоп визначає орієнтацію і нахил пристрою в просторі. В багатьох випадках гіроскоп використовують разом із акселерометром для відстеження змін в рухах. В електронних пристроях програмне забезпечення, що

використовується разом з гіроскопом, здатне швидко реагувати на переміщення пристрою в просторі і приймати відповідні рішення.

GPS-навігатор визначає місцеположення мобільного пристрою на Землі, використовуючи для цього дані, отримані з навігаційних супутників. За допомогою належних додатків телефон з GPS-навігатором може замінити переносний зовнішній GPS.

Мобільні пристрої використовуючи ці три основних сенсори мають змогу визначити більшість подій, які відбуваються із ними. Сенсори, такі як датчики наближення, освітлення та інші, можуть надати додаткову інформацію про стан мобільного пристрою та збільшити шанс на коректне виявлення події.

2.1.2.1 Аналіз обмежень мобільного пристрою

Розробляючи додатки для мобільних пристроїв необхідно враховувати їх апаратні характеристики, які, порівнюючи із апаратними характеристиками сучасних стаціонарних комп'ютерів, менш продуктивні та можуть вимагати більше часу на обробку деяких операцій. Два основних фактори, це продуктивність роботи процесора та швидкість передачі даних через мережу.

Перевантажуючи мобільний пристрій зайвою роботою є ризик отримати некоректні дані, зависання, або аварійне завершення роботи програми. Саме тому було б доцільно покласти основну роботу з обробки даних на деякий комп'ютер, в той час як мобільні пристрої будуть лише відправляти необхідні дані та отримувати потрібну інформацію.

Мінімальна функціональність мобільного додатка повинна складатися з двох частин: отримання даних від сенсорів та їх передача через мережу. Для більшості смартфонів отримання даних зі всіх сенсорів в середньому займає 0.3 секунди. В залежності від типу передачі даних, розміру інформації та якості зв'язку з мережею відправлення даних може займати різний час.

Таким чином, проаналізувавши ці обмеження необхідно написати такий мобільний додаток, який буде:

- реалізовувати мінімальний необхідний функціонал;
- використовувати мінімум апаратних ресурсів пристрою;
- оптимізований для отримання даних з сенсорів;
- формувати мінімальну структуру даних для передачі через мережу.

2.1.2.2 Проектування та розробка прототипу мобільного додатку

Визначення алгоритму роботи додатку

Основна задача мобільного додатку – це збір даних з сенсорів та їх подальша передача деякому сервісу, який і буде виконувати обробку даних. Виконуючи ці дії необхідно враховувати час, який займатиме кожна операція, та встановити певний часовий інтервал між ними. Отримавши дані з сенсорів їх треба зберегти в певний формат, який не займатиме багато ресурсів та буде зручним для передачі через мережу. Дані, що були збереженні у відповідному

форматі, відправлятимуться на задану адресу, де будуть проаналізовані програмним забезпеченням. Для забезпечення необхідної послідовності передачі даних та запобігання некоректній роботі мобільного додатку, попередні дії необхідно виконувати із заданим інтервалом часу, щоб уникнути перенавантаження щодо роботи з сенсорами та передачі даних через мережу. Заданий інтервал часу є загальний середній час отримання даних з сенсорів та їх передачі через мережу.

Алгоритм такого мобільного додатка буде складатися із таких кроків:

1. Отримання даних з сенсорів.
2. Запис даних у зручний формат.
3. Відправлення даних на задану адресу.
4. Переведення потоку виконання програми у «режим сну» на встановлений час, після чого повернення до першого кроку.

Вибір технологій передачі даних

Після того, як додаток отримав значення з сенсорів, їх необхідно зберегти в певному спеціальному форматі, який буде мати мінімальний розмір, і в той же час матиме достатню кількість інформації для встановлення від якого сенсора були отримані значення. Також необхідно визначити куди саме ці дані будуть надсилатися.

Для зберігання даних та формування зворотних відповідей для додатку було обрано текстовий формат передачі обміну JSON. Основні переваги цього формату, це стислий розмір, зрозумілий текст повідомлення (як для комп'ютера, так і для людини), а також придатність для реалізації складних структур даних. Кожне поле в цій структурі даних відповідає певному типу датчика, що доступний в смартфоні.

Дані можуть відправлятися по одиночно, або відразу набором із декількома отриманими значеннями. Для реалізації мобільного додатку буде використовуватись саме другий варіант. Дані для передачі отримуватимуться з трьох основних сенсорів мобільного пристрою: акселерометра, гіроскопа та GPS-навігатора. Отримані та упаковані в JSON-формат дані будуть мати такий вигляд (Рис. 2.1.1.1):

```
{
  "Acc_X": 0.0,
  "Acc_Y": 9.6172,
  "Acc_Z": 0.8192,
  "Gyr_X": 1.3811,
  "Gyr_Y": 5.7134,
  "Gyr_Z": 3.9483,
  "longtitude": 50.4506697,
  "latitude": 30.5231078
}
```

Рис. 2.1.1.1. Вигляд даних у JSON-форматі.

Структура даних, яка була отримана після серіалізації зчитуваних з сенсорів значень, може бути відправлена через мережу на вказану заздалегідь адресу сервісу, який матиме можливість отримати значення сенсорів та виконати їх необхідну обробку.

Для відправлення даних у JSON-форматі необхідно визначити яким чином вони будуть надсилатися, а саме – який мережевий протокол передачі даних буде використаний. Так як JSON-формат в більшості випадків призначений для обміну даними між веб-браузером та сервером, можна використати для цього протокол HTTP. Але, враховуючи те, що дані будуть надсилатися на певний сервіс, до того ж швидкість передачі є важливим фактором, виникає потреба використання оптимального рішення (Рис. 2.1.1.2).



Рис. 2.1.1.2. Модель OSI та TCP/IP.

Протоколи HTTP та TCP працюють на різних функціональних рівнях: HTTP знаходиться на прикладному рівні, і використовується для передавання гіпертексту, XML-файлів, потокового відео й аудіо. TCP є протоколом транспортного рівня, головною задачею якого є передача сегментів інформації від джерела до місця призначення, яка надійшла з прикладного рівня. Таким чином, протокол HTTP використовує TCP як протокол транспортного рівня. Ще один протокол, який можна використати, це UDP, який передає дані якнайшвидше, але без перевірки на правильність передавання, контролю первантажень і гарантій доставки.

Заради уникнення зайвої роботи, збільшення швидкості передачі даних та можливості створювати надійні канали було обрано протокол TCP. В даному випадку обмін даними поділено на три чітко виражені фази: організація з'єднання, передавання даних і розривання з'єднання.

Реалізація клієнт-серверної архітектури

Перед тим як створити повноцінну систему необхідно розробити деякий її примітивний прототип, щоб ознайомитись із особливостями реалізації та використати отримані розробки в майбутньому. Маючи алгоритм роботи мобільного додатку та визначившись із необхідними технологіями для передачі даних можна розробити клієнт-серверну архітектуру. Для цього необхідно реалізувати деякий додаток для комп'ютера (виконуючий роль веб-сервісу), який буде отримувати запити на з'єднання та отримувати дані, а також сам мобільний додаток для операційної системи Android, який буде надавати дані з сенсорів. Обидва додатки будуть реалізовані використовуючи мову програмування Java.

Веб-сервіс буде очікувати запити на здійснення з'єднання, встановлювати канали зв'язку та друкувати отримані дані у консоль. Обробка та аналіз цих даних будуть розглянуті під час проектування та розробки основної системи. Мобільний додаток буде складатися з одного екрану та однієї кнопки, яка після натискання за вказаним в коді IP-адресом відправить запит на з'єднання та встановить канал зв'язку. Використовуючи цей канал додаток з певним інтервалом часу буде отримувати дані з сенсорів, упаковувати їх у текстовий JSON-формат та надсилати до веб-сервісу. Повторне натискання кнопки зупинить потік надання даних та закриє канал зв'язку між додатком та веб-сервісом (Рис.2.1.1.3).

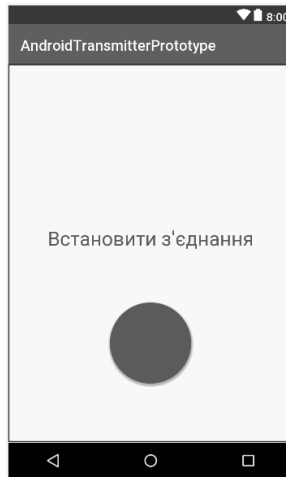


Рис.2.1.1.3. Екран прототипу мобільного додатку.

Спершу необхідно запусити веб-сервіс, котрий буде отримувати з'єднання. Для цього буде використовуватися сокет - комунікаційний інтерфейс взаємодіючих процесів. Для створення серверного сокета його потрібно прив'язати до певного порту операційної системи, після чого процес, який буде очікувати запити на з'єднання, перейде в режим очікування. Після створення нового каналу зв'язку процес створить для нього окремий потік виконання

програми, який буде обробляти вхідні дані, а після закінчення сеансу роботи і розриву з'єднання потік завершить свою роботу. Таким чином, веб-сервіс зможе працювати з декількома мобільними пристроями одночасно.

Після запуску веб-сервісу мобільний додаток може встановити з ним канал зв'язку та передавати дані (Рис. 2.1.1.4). На початку роботи всі необхідні сенсори реєструються у системі та при кожній зміні значення сенсора додаток зберігає його значення в відповідних ділянках пам'яті (змінних).

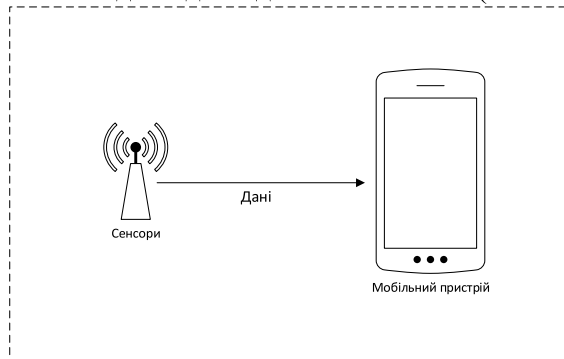


Рис. 2.1.1.4. Отримання даних з сенсорів.

Після натискання кнопки, котра відповідає за встановлення з'єднання, додаток спробує підключитись за вказаним IP-адресом та портом до веб-сервісу і, в разі успіху, створить надійний канал зв'язку за протоколом TCP, після чого почне формувати JSON-об'єкти із значеннями сенсорів та надсилати їх до серверу. В результаті роботи веб-сервісу будуть отримані дані від мобільного додатку, які були передані через встановлений канал зв'язку (Рис. 2.1.1.5, Рис. 2.1.1.6).

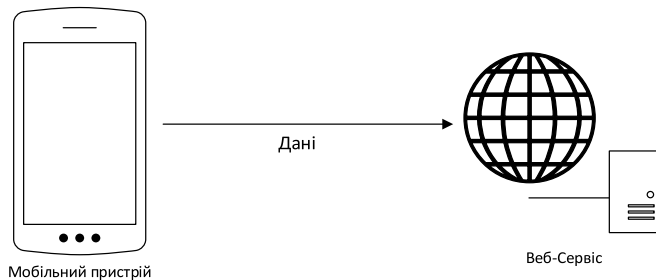


Рис. 2.1.1.5. Передача даних.

Середній час, із яким мобільний пристрій збирав всі значення з сенсорів та формував JSON-об'єкт, становив 0.3 секунди а надсилання даних до веб-сервісу в середньому займало 0.5 секунди. Таким чином в мобільному додатку можна використовувати два потоки роботи: один буде займатися зчитуванням та збереженням даних з сенсорів, в той час як другий буде надсилати відомі дані до веб-сервісу. Після того, як всі необхідні дані будуть надіслані, мобільний додаток зупиняє передачу даних та закриває канал зв'язку.

Таким чином, була реалізована система, яка виконує більшість необхідних дій, зокрема такі як отримання даних з сенсорів, їх формування в зручний формат, створення надійного каналу зв'язку для передачі інформації та її надсилання до веб-сервісу.

```
Receiver x
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" "--javaagent:D:\Programming\IntelliJ IDEA Community Edition 2018.3.3\
Awaiting...
Accepted
{"Acc_X": "-4,7333", "Acc_Y": "1,9681", "Acc_Z": "7,9632", "Gyr_X": "-1,9181", "Gyr_Y": "-1,8816", "Gyr_Z": "-0,2628",
{"Acc_X": "2,5906", "Acc_Y": "4,561", "Acc_Z": "18,8257", "Gyr_X": "4,9517", "Gyr_Y": "-3,7799", "Gyr_Z": "-0,0124",
{"Acc_X": "4,2162", "Acc_Y": "-3,6966", "Acc_Z": "-0,4333", "Gyr_X": "9,9584", "Gyr_Y": "0,4087", "Gyr_Z": "-2,7821",
{"Acc_X": "9,584", "Acc_Y": "-0,8331", "Acc_Z": "-6,8091", "Gyr_X": "0,1963", "Gyr_Y": "5,5912", "Gyr_Z": "-4,2128",
{"Acc_X": "3,5866", "Acc_Y": "-1,8435", "Acc_Z": "-4,7955", "Gyr_X": "-8,5485", "Gyr_Y": "8,9255", "Gyr_Z": "-8,6581",
{"Acc_X": "-1,5801", "Acc_Y": "-7,252", "Acc_Z": "8,3247", "Gyr_X": "-6,3125", "Gyr_Y": "1,2353", "Gyr_Z": "-6,0823",
Ended
Process finished with exit code 0
```

Рис. 2.1.1.6. Отриманні дані з сенсорів.

Використання поштового сервісу в якості приймача

Інший підхід, який можна використати для передачі даних, це запис даних у звичайний лист та його відправлення на задану електронну адресу. Цей метод є універсальним, оскільки більшість систем підтримують роботу із поштовим сервісом та можуть використовувати його в якості каналу зв'язку. Один із найбільш поширених поштових сервісів, Gmail, який підтримується компанією Google, може бути використаний для реалізації ще одного прототипу мобільного додатку для операційної системи Android. Для цього, можна використати бібліотеку Gmail API, яка має змогу читати та відправляти повідомлення, керувати чернетками та вкладеннями, шукати повідомлення, працювати з мітками, налаштовувати сповіщення та керувати налаштуваннями поштового сервісу. Бібліотека може бути використана багатьма мовами програмування, в тому числі й мовою Java [113].

Дані, отримані з сенсорів мобільного пристрою, можна одним натисканням кнопки відправити на задану електронну адресу, використовуючи бібліотеку Gmail API та Google акаунт. Мобільний додаток знову буде зчитувати дані з сенсорів, зберігати їх відомі значення та очікувати натискання кнопки, яке сформує листа та відправить його на вказану в коді електронну адресу. Крім самої адреси, в коді також повинні бути вказані логін і пароль для авторизації у поштовому сервісі, який використовуватиметься для відправлення листа (Рис. 2.1.1.7). Натискання на кнопку задіє функцію, яка виведе на екран дані, що були отримані з сенсорів, після чого збереже цю інформацію в текстовому форматі у листі, який згодом буде відправлений на вказану електронну адресу (Рис.2.2.1.8). Перед тим як відправити листа додаток, використовуючи мережу інтернет, повинен авторизуватися в поштовому сервісі, вказавши логін та пароль деякого Google акаунту. Після успішної авторизації додаток має змогу відправити листа. Цей етап займає 3-4 секунди (Рис. 2.1.1.9, Рис. 2.1.1.10).



Рис. 2.1.1.7. Вигляд додатку на початку.

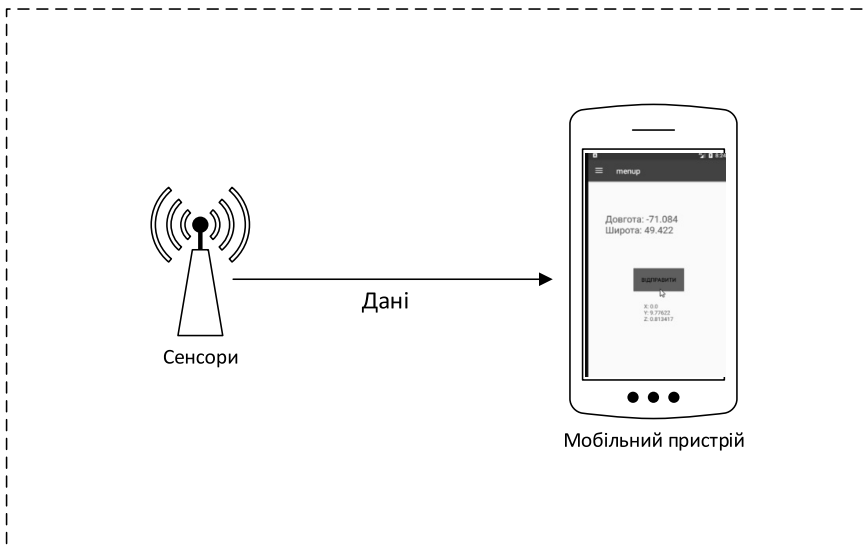


Рис.2.1.1.8. Отримання даних з сенсорів мобільного пристрою.

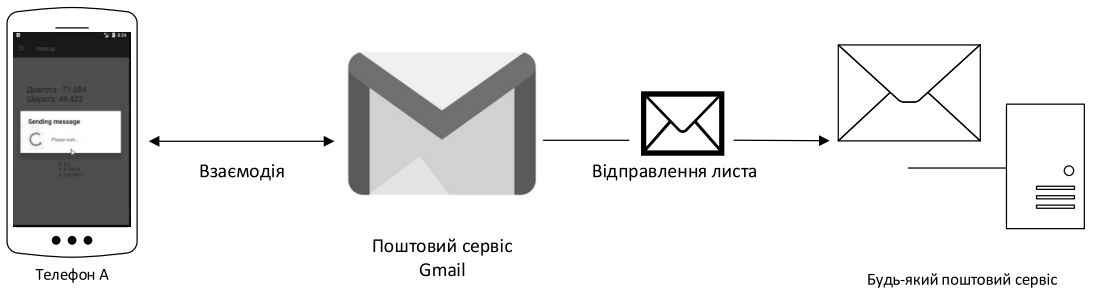


Рис. 2.1.1.9. Принцип роботи.



Рис. 2.1.1.10. Програма відправляє листа.

Якщо лист було успішно відправлено, програма повідомляє про це користувачу, а через деякий час на вказаній пошті повинен з'явитися новий лист із даними, які були отримані з сенсорів мобільного пристрою (Рис. 2.1.1.11).



Рис. 2.1.1.11. Програма успішно відправляє листа на пошту.

Надалі ці дані можуть бути прочитані системою, котра і виконає їх подальшу обробку. Таким чином, був реалізований другий прототип мобільного додатку, який використовує поштовий сервіс як канал зв'язку передачі даних. Головним недоліком такого підходу є довгий час відправки даних, причиною чого є обов'язкова авторизація в системі Google, але, через свою унікальність, вартий уваги для використання в проектах для інших цілей.

2.1.3 Проектування та розробка

Розглянувши основні типи сенсорів, проаналізувавши їх обмеження та роботу з ними, створивши прототип мобільного додатку та встановивши основні вимоги було отримано представлення майбутньої системи. Розроблений в попередньому розділі прототип виконує своє основне призначення – отримує значення вбудованих сенсорів, та надсилає їх до веб-сервісу, тим самим надаючи інформацію про свій стан.

Дані, які мобільний додаток надсилає до веб-сервісу, необхідно певним чином аналізувати та на основі цього аналізу визначати можливі небезпечні події, після чого, за попередньо вказаними діями, попереджати про можливу небезпеку. Перед початком розробки системи необхідно визначити алгоритм її роботи, яким чином веб-сервіс повинен виявляти небезпечні події та обрати технології й інструменти для її реалізації. Також необхідно визначити яким чином буде зберігатися отримана з мобільних пристроїв інформація у веб-сервісу, та особливість її обробки.

2.1.3.1 Визначення алгоритму роботи системи

Три основні задачі, які повинна вирішувати система, це:

1. Надавати значення сенсорів веб-сервісу.
2. Виявляти небезпечні події.
3. Повідомляти про можливу небезпеку.

Надаючи інформацію, яка була отримана з сенсорів смартфона, мобільний додаток таким чином повідомляє про свій стан веб-сервісу. Розроблений в попередньому розділі прототип вирішує цю задачу. Веб-сервісу, який буде отримувати ці дані, необхідно зберігати отриману інформацію, після чого, використовуючи визначений алгоритм, проаналізувати зміни значень сенсорів за певний діапазон часу та з'ясувати, чи відбувається деяка небезпечна ситуація. Визначивши, що є можливість виникнення небезпечної події із користувачем смартфона, веб-сервіс повинен повідомити про це користувачів системи, використовуючи для цього мобільний додаток, або за необхідністю служби безпеки. Таким чином, мобільний додаток може як надсилати інформацію про стан поточного смартфона, так і отримувати попередження про можливі загрози іншого мобільного пристрою. Щоб отримувати сповіщення про можливу небезпечну подію для користувача іншого смартфона, його необхідно зареєструвати у веб-сервісі та вказати необхідний мобільний пристрій, за станом якого необхідно стежити.

Таким чином, алгоритм основної системи, яка буде складатися з мобільного додатку та веб-сервісу, буде складатися із таких кроків:

1. Відправлення даних
 - 1.1. Отримати дані з сенсора.
 - 1.2. Упакувати дані у зручний формат JSON для передачі через мережу.
 - 1.3. Відправити пакет даних на задану адресу за протоколом TCP.

- 1.4. Перевести потік виконання у «режим сну» на встановлений час, після чого перейти до першого кроку.
2. Аналіз даних
 - 2.1. Отримати дані з мобільного пристрою за протоколом TCP.
 - 2.2. Зберегти отримані дані.
 - 2.3. Проаналізувати дані за деякий час на виявлення небезпечних подій.
 - 2.3.1. При виявленні можливої небезпечної події відправити цю інформацію на зареєстровані пристрої, після чого повернутися до першого кроку.
 - 2.3.2. Якщо результати обробки даних вказують на те, що небезпечних подій не було зафіксовано, повернутися до першого кроку.
3. Отримання мобільними пристроями інформації про подію
 - 3.1. Отримати повідомлення про вияв події та сповістити про це користувача з одночасним виведенням назви та детальної інформації події на екран.
 - 3.2. Очікувати команди користувача.

2.1.3.2 Проектування архітектури та засоби реалізації системи

Для розробки системи буде використовуватися клієнт-серверна архітектура, де клієнтом буде мобільний додаток, тоді як деякий сервер буде надавати доступ до веб-сервісу, котрий буде отримувати запити на з'єднання, зберігати та обробляти надіслану інформацію.

Клієнтська частина – мобільний додаток – буде розроблятися для смартфонів під керуванням операційної системи Android, з використанням мови програмування Java та технології JSON для передавання інформації між мобільним додатком та веб-сервісом.

Веб-сервіс також буде реалізований з використанням мови програмування Java та технологій JSP, JavaScript, JSON та програмним інтерфейсом Java Servlet API. Для побудови веб-інтерфейсу буде використана мова розмітки HTML та каскадні таблиці стилів CSS, а для взаємодії будуть використовуватися веб-технології JavaScript, JQuery та AJAX. Таким чином, для взаємодії із веб-інтерфейсом користувачу не потрібно жодного програмного забезпечення крім браузера.

Для зберігання даних буде використана система управління базами даних PostgreSQL. Основні дані, що будуть зберігатися на сервері, це дані з сенсорів мобільного пристрою та інформація про те, яким додаткам будуть надсилатися сповіщення про можливі небезпечні ситуації.

Для створення надійного каналу зв'язку між мобільним додатком та веб-сервісом буде використаний програмний інтерфейс сокетів та протокол TCP.

На початку роботи системи необхідно запустити веб-сервіс, який буде отримувати запити на з'єднання. Після цього користувач запускає мобільний додаток та обирає один з двох режимів його роботи:

- «Відправник» - мобільний додаток буде отримувати дані із всіх доступних (або заданих в налаштуваннях) сенсорів, та із заданим інтервалом буде передавати їх у зручному форматі до веб-сервісу.
- «Отримувач» - в цьому режимі мобільний додаток буде лише отримувати інформацію, яку йому буде надсилати веб-сервіс, і сповіщати про це користувача.

Для прикладу маємо три мобільних пристрої (смартфони) із встановленим мобільним додатком і веб-сервіс (Рис. 2.1.3.1):

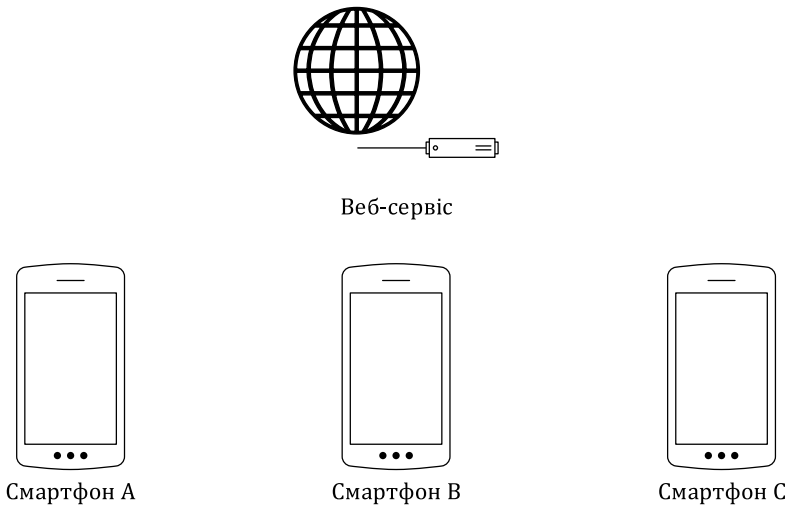


Рис. 2.1.3.1. Основні компоненти системи.

У прикладі смартфон А буде використовувати режим «Відправник», інші два – «Отримувач» (Рис. 2.1.3.2):

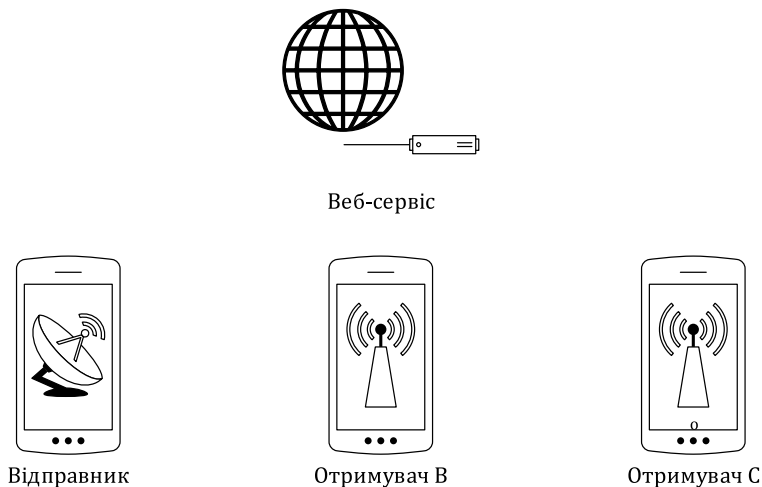


Рис. 2.1.3.2. Смартфонам задані режими.

Після того, як мобільні пристрої визначили свої режими роботи, необхідно встановити зв'язок між цими смартфонами та веб-сервісом. Для встановлення зв'язку між мобільним додатком та веб-сервісом, необхідно вказати IP-адресу серверу та порт, на якому веб-сервіс очікує запитів на встановлення каналів зв'язку. Номер порту буде узгоджений ще на етапі розробки, але IP-адреса буде змінюватися з використанням різних серверів. Для цього у веб-сервісу є можливість виведення на екран своєї IP-адреси. Так як процедура ведення IP-адреси на мобільному додатку вручну може бути громіздкою, а іноді і з великим шансом виникнення помилки, веб-сервіс також надає QR-код із закодованим адресом. Мобільний додаток може зчитати цю інформацію та миттєво підключитися до веб-сервісу (Рис. 2.1.3.3).



Рис. 2.1.3.3. Зчитування QR-коду мобільним додатком.

Процедура зв'язування між мобільним додатком та веб-сервісом майже ідентична процедурі встановлення зв'язку між деякими пристроями за допомогою технології Bluetooth: пристрої об'єднуються своїми даними (IP-адресом, MAC-адресом, і т.д.), перевіряють коректність цих даних, і, нарешті, встановлюють між собою прямий зв'язок за протоколом TCP (Рис. 2.1.3.4):

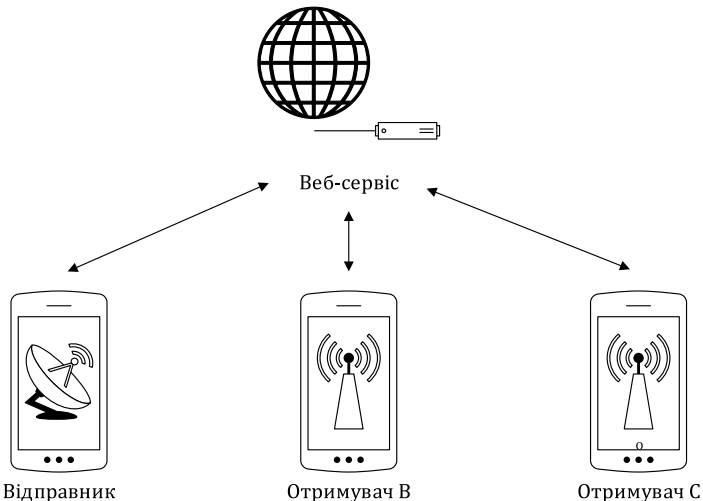


Рис. 2.1.3.4. Мобільні додатки встановлюють зв'язку із веб-сервісом.

Для зчитування QR-коду необхідно, щоб мобільний пристрій мав камеру. У разі її відсутності необхідно вводити ці дані вручну. Після встановлення з'єднання веб-сервіс отримує список підключених мобільних пристроїв та їх режими роботи. Як було зазначено вище, більшість мобільних пристроїв мають свої обмеження у продуктивності та швидкості роботи, тому значну роботу доведеться робити веб-сервісу, а саме:

- Обробляти отриману інформацію.
- Сповіщати про небезпечні ситуації.
- Встановлювати, яким підключеним мобільним пристроям необхідно відправляти цю інформацію.
- Контролювати встановлені зв'язки із мобільними пристроями.

Таким чином, після звільнення мобільного пристрою від зайвих задач він зосереджується на своїх основних функціях:

- Отримувати та відправляти дані з сенсорів (для «Відправника»).
- Отримувати дані з веб-сервісу (для «Отримувача»).

Веб-сервіс буде використовувати два окремих потоки роботи: один буде отримувати запити на з'єднання, в той час як інший буде виконувати безпосередню обробку даних. Після отримання від мобільного додатку даних веб-сервіс зберігає їх в базі даних. В цілях економії місця та запобігання перенавантаження бази даних було вирішено зберігати інформацію, яка була отримана, в діапазоні 2-х хвилин, яких в багатьох випадках достатньо для визначення небезпечної події. Таким чином, якщо нема підозр на небезпечну ситуацію – значення сенсорів, які були отримані більше двох хвилин назад, видаляються з бази даних заради звільнення місця та отримання нових (Рис. 2.1.3.5).

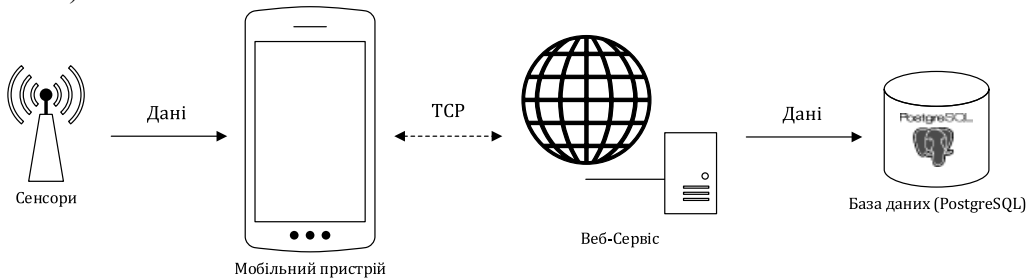


Рис. 2.1.3.5. Передача та збереження даних.

Якщо була встановлена можливість деякої небезпечної події, веб-сервіс намагається надати всім можливим пристроям, що зареєстровані на отримання інформації, сповіщення користувача про можливу небезпеку. Такими пристроями можуть бути інші мобільні пристрої, або навіть розумний годинник (Рис. 2.1.3.6).

Ці мобільні пристрої можуть встановлювати необхідний рівень оповіщень (від простих, таких як текстове сповіщення у рядку статусу смартфона, до

більш складних, таких як використання гучного звукового сигналу, миготіння екрану та зв'язком з іншими пристроями).

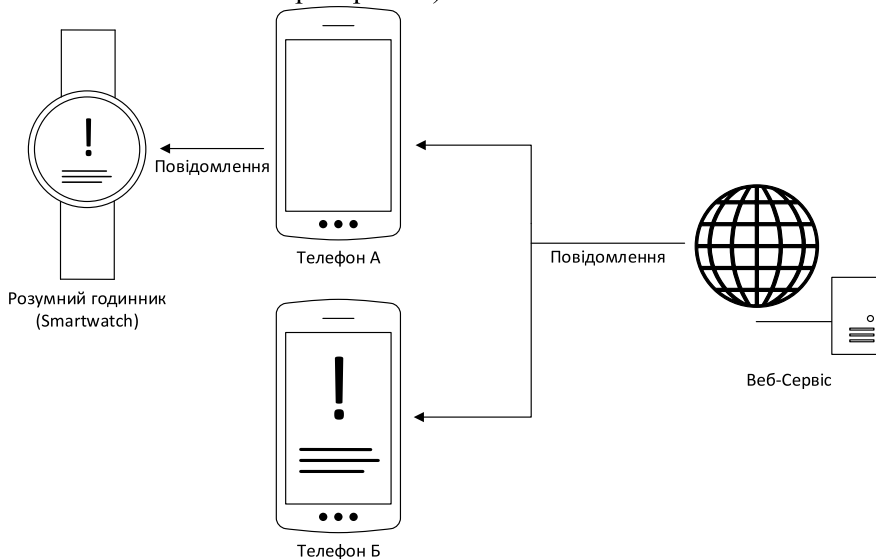


Рис. 2.1.3.6. Відповідь веб-сервісу.

Веб-сервіс, як і мобільний додаток, надає можливість керувати встановленими з'єднаннями, і, в разі потреби, може перервати зв'язок. Таким чином, веб-сервіс може керувати з'єднаннями між мобільними додатками та сервісом, визначати які мобільні пристрої будуть отримувати необхідну інформацію.

Для формування даних у JSON-формат на стороні клієнта, були використані системні інструменти для розробки Android. Веб-сервіс не має подібних вбудованих засобів, тому для їх реалізації необхідно використати сторонню бібліотеку. Для цього було задіяно програмний інтерфейс Gson. Gson – бібліотека, розроблена компанією Google, яка може використовуватися для перетворення об'єктів Java у їх JSON-репрезентацію. Вона також може бути використана для перетворення об'єкту, який знаходиться в текстовому JSON-форматі, в еквівалентний Java-об'єкт. Gson може працювати з довільними Java-об'єктами, включаючи вже існуючі об'єкти, у яких немає вихідного коду [82].

Для того, щоб веб-сервіс мав змогу сформувати QR-код IP-адреси серверу та забезпечити мобільний додаток можливістю його зчитування необхідно також застосувати сторонній програмний інтерфейс. Таким інтерфейсом буде ZXing. ZXing - це відкрита, багатоформатна бібліотека обробки штрих-кодів, реалізована за допомогою мови програмування Java з портами для інших мов. Бібліотека може використовуватися як у веб-застосунках, так і на мобільних платформах [123].

Ці інструменти значно допомогли та спростили як процес розробки системи, так і її використання (Рис. 2.1.3.7).

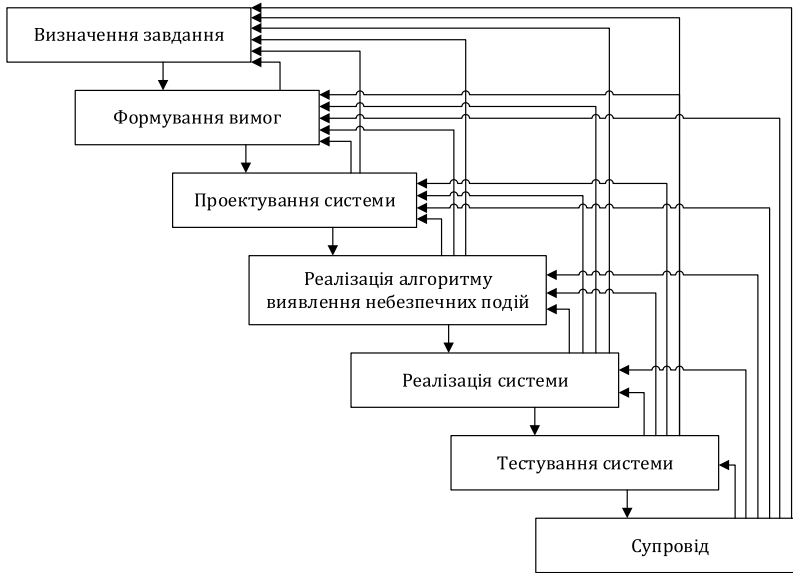


Рис. 2.1.3.7. Життєвий цикл програмного забезпечення.

2.1.3.3 Алгоритм виявлення небезпечних подій

Головною складовою системи є метод визначення небезпечних подій, який використовує для цього аналіз отриманих з сенсорів мобільного пристрою даних. Більшість таких небезпечних подій характеризуються втратою свідомості, яка спричинена або викликана падінням чи зіткненням з іншим об'єктом.

Використання акселерометра є одним з найбільш широко використовуваних методів, що застосовуються для вимірювання та моніторингу фізичної активності людини. Хоча поведінка та рухи людини можуть бути складними та різноманітними і через те, що різні дії можуть виконуватися як послідовно, так і одночасно, існують помітні особливості для деяких явищ, таких як падіння. Отже, саме акселерометр, котрий отримує значення лінійного прискорення, буде виявляти, чи виникає певна небезпечна ситуація.

Загалом, існують два методи виявлення падіння: за допомогою машинного навчання та аналітичний. Для більш точного виявлення таких подій, як падіння, може бути використаний метод машинного навчання, який потребує деякого часу, збору інформації користування та виявлення послідовності зміни значень, при яких може виникнути небезпечна подія для різних людей, але, для більшості випадків, може бути використаний аналітичний метод.

Акселерометр вимірює прискорення у m/c^2 (метрах на секунду в квадраті) за трьома координатами, та надає їх поточні значення, що можна вказати як $A = (A_x, A_y, A_z)$, де A_x, A_y, A_z - значення прискорень на осях абсцис, ординат та аплікат відповідно (Рис. 2.1.3.8).

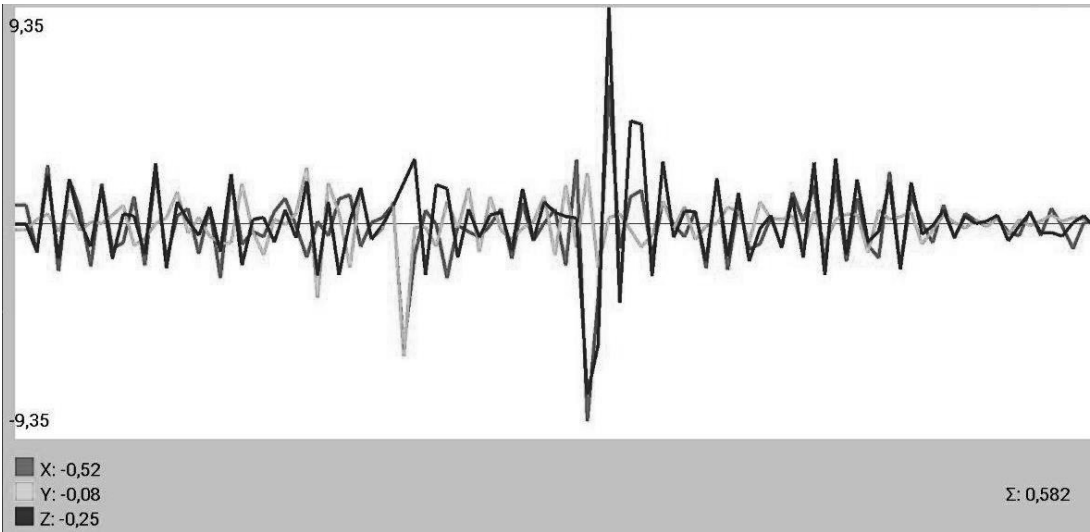


Рис. 2.1.3.8. Графік отриманих з акселерометра значень.

Кожне отримане значення фіксує прискорення тільки за своїм напрямом, тому необхідно привести дані трьох координат до певного спільного значення та отримати загальний графік прискорень. Для цього, потрібно отримати корінь суми квадратів всіх трьох значень координат:

$$g = \sqrt{A_x^2 + A_y^2 + A_z^2}$$

Таким чином, буде побудовано загальний графік значень прискорення мобільного пристрою. Результати аналізу змін цих значень і виявлять можливе падіння мобільного пристрою.

Щоб визначити чи відбувається падіння, необхідно визначити три характерні ознаки зміни значень, а саме:

1. **Вільне падіння.** Короткий період вільного падіння завжди відбувається на початку падіння, що може свідчити про його початок. Тривалість стану невагомості залежить від висоти падіння.

2. **Момент удару.** Як тільки стан невагомості минув, людський організм може зазнати удару об землю або інші перешкоди. Момент часу, коли два фізичні тіла стикаються та виникає зіткнення, призводить до різкої інверсії напрямку вектора прискорення, яке помітно на графіку.

3. **Нерухоме положення.** Після падіння, людина, зазвичай, не може відразу піднятись. Виключення становлять випадки, коли наслідки падіння не завдали значної шкоди здоров'ю людини. Встановлений акселерометром короткий чи тривалий час нерухомості після можливого падіння може свідчити про вірогідну втрату свідомості людини. На графіку це виглядає як майже рівна лінія (Рис. 2.1.3.9).



Рис. 2.1.3.9. Зафіксовані ознаки можливого падіння.

Якщо було зафіксовано всі три характерні ознаки падіння, система робить висновок, що виникла деяка небезпечна подія, що призвела до падіння або зіткнення із значною силою з іншим фізичним об'єктом, яка могла супроводжуватись втратою свідомості користувача. Отже головною задачею веб-сервісу буде виявлення саме цих ознак.

Зміни значень, що вказують на можливе падіння, добре помітні на графіку прискорення, але для виявлення їх системою, необхідно визначити три пороги значень, за допомогою яких і будуть виявлятися зазначені ознаки.

Для порогів **вільного падіння** та **моменту удару** будуть використовуватися зміни T_f та T_s відповідно. Час падіння зазвичай триває від 0.4 до 0.8 секунди, тобто час між **вільним падінням** та **моментом удару** повинен становити не менше 800 мілісекунд. Це буде третім порогом, який буде визначений змінною T_d .

Отже, сервіс виявить можливе падіння, якщо були виконані умови в наступній послідовності:

1. Значення акселерометра були нижче нижнього порогу T_f :

$$\sqrt{A_x^2 + A_y^2 + A_z^2} < T_f$$

2. Значення акселерометра були вище верхнього порогу T_s :

$$\sqrt{A_x^2 + A_y^2 + A_z^2} > T_s$$

3. Час між вільним падінням (t_f) та моментом удару (t_s) менше за T_d :

$$t_s - t_f < T_d$$

Значення порогів T_f та T_s залежать головним чином від положення мобільного пристрою: у кармані, у руці, в сумці, рюкзаку, на ремні.

вразливими для помилок є положення мобільного пристрою у руці, оскільки значення, отримані з акселерометра, можуть бути хаотичними через реакцію користувача на падіння. Аналіз даних для подібних положень мобільного пристрою може виявитись складнішим за інші. Використовуючи метод опорних векторів [70], були визначені наступні значення порогів T_f та T_s для різних положень мобільного пристрою (Табл. 2.1.3.1):

Таблиця 2.1.3.1.

Встановленні значення порогів.

Позиція	На ремні	У кишені	У нагрудному кармані	У руці
T_f	0.52g	0.42g	0.63g	0.53g
T_s	2.02g	2.43g	2.48g	2.82g
T_d	800 мілісекунд			

Використовуючи ці значення, система може встановити різні порогови для відповідних положень. Але для цього також необхідно встановити, у якому положенні знаходиться мобільний пристрій. У повсякденному житті люди носять свої мобільні телефони у різних положеннях. Дослідження вказують на те, що 60% чоловіків носять телефон в кишені, а 61% жінок - у сумці. Наступними найбільш поширеними місцями для носіння мобільного пристрою є затиски для ременів та нагрудні кармани [83]. Акселерометр може виявити більшість положень мобільного пристрою та дії користувача, але для більш точніших висновків необхідно також використати значення інших сенсорів, таких як сенсори освітлення, приближення та гіроскоп (Рис. 2.1.3.10).

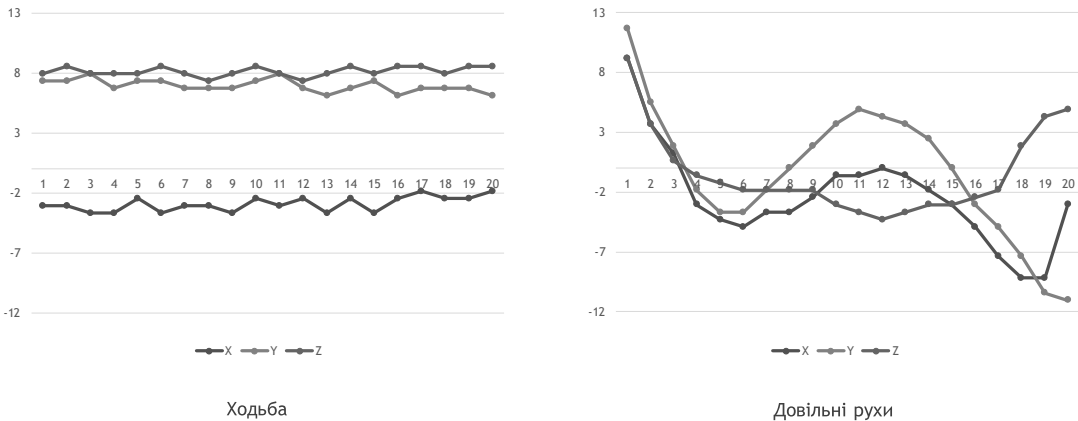


Рис. 2.1.3.10. Значення акселерометра для різних положень та дій.

Значення акселерометра можуть визначити знаходиться користувач в нерухомому положенні, чи він пересувається. Також можна визначити кут нахилу мобільного пристрою, що також вказує, чи знаходиться смартфон у кишені або ж на ремні.

Прискорення руху, яке вимірює гіроскоп, може вказати, чи знаходиться мобільний пристрій у кишені або у руці. Маючи телефон у руці, значення гіроскопа зазвичай більш хаотичні, ніж у кишені.

Сенсори освітлення та приближення можуть визначити, чи знаходиться мобільний пристрій у кишені, сумці або у руці. Якщо значення освітлення досить низьке, телефон може бути у кишені, рюкзаку або у сумці, в інших випадках – на поясі або у руці.

Таким чином, можна визначити положення мобільного пристрою та пересування користувача. Використовуючи методи машинного навчання, також можливо поліпшити процес визначення положення смартфона та збільшити ймовірність коректної обробки.

Отже, отримавши дані від сенсорів мобільного пристрою, веб-сервіс виконує обробку даних на виявлення небезпечних подій із користувачем наступним чином:

1. Визначення положення та пересування користувача;
2. Встановлення значень порогів відповідно до положення смартфона (Табл. 2.1.3.1);
3. Перевірка отриманих від акселерометра значень на наявність характерних ознак падіння (вільне падіння та момент удару) та можливу втрату свідомості (стан нерухомості після виявлення можливого падіння).
 - 3.1. Якщо ознаки були виявленні – надіслати необхідну інформацію зареєстрованим мобільним додаткам та повідомити про можливу небезпеку для користувача;
 - 3.2. Якщо ознак не було виявлено – повернутися до першого кроку.

2.1.3.4 Реалізація та принцип роботи системи

Система складається з двох компонентів: веб-сервісу та мобільного додатку. Для роботи з веб-сервісом потрібен лише веб-браузер, для мобільного додатку – мобільний пристрій під керуванням операційної системи Android.

2.1.3.4.1 Робота із веб-сервісом

Задачами веб-сервісу є отримання запитів на встановлення з'єднання між мобільними пристроями та сервером, реєстрація необхідних мобільних пристроїв на отримання інформації про стан користувача іншого смартфона, обробка даних на виявлення можливих небезпечних подій та надання відповідей мобільним додаткам. Кодова назва веб-сервісу – Wallott (англ. Web Allottee – Веб-отримувач).

Графічний інтерфейс користувача складається з двох частин:

1. **Керування веб-сервісом.** Тут виконується запуск та зупинка роботи системи та надання QR-коду із даними про IP-адресу.

2. **Керування встановленими з'єднаннями.** Основна частина взаємодії між користувачем та системою виконується в цьому вікні. Тут відображаються під'єднанні мобільні пристрої, їх режими та стан. Також тут встановлюються які мобільні додатки із режимом «отримувач» будуть отримувати інформацію про стан інших мобільних пристроїв та є можливість, за необхідністю, закінчити процес передачі даних на окремих смартфонах (Рис. 2.1.3.11).

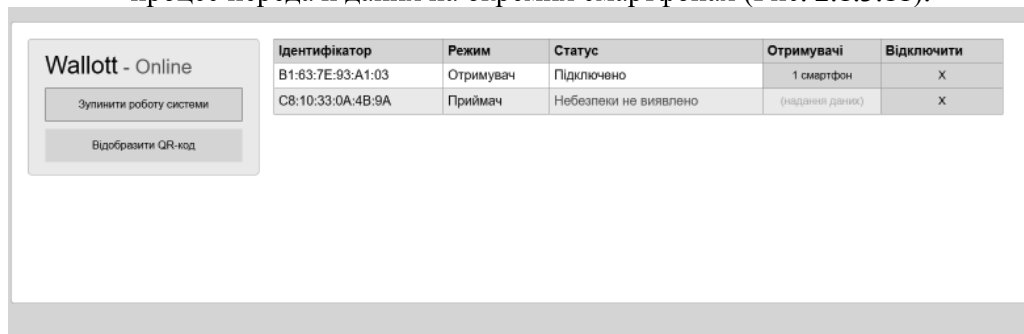


Рис. 2.1.3.11. Вигляд інтерфейса веб-сервісу.

Робота системи починається з запуском веб-сервісу, що виконується в панелі керування сервісом, після чого система очікує запити на з'єднання та отримання даних для їх подальшої обробки.

2.1.3.4.2 Робота із мобільним додатком

Мобільний додаток надає дані з сенсорів та отримує інформацію з веб-сервісу про можливу небезпеку для користувача. Для цього, використовується один із двох режимів - отримувач або відправник. Кодова назва мобільного додатку – Rectran (англ. Receiver and Transceiver – Приймач та відправник).

Основні екрани мобільного додатку:

1. **Вибір режиму та налаштування.** В цьому вікні мобільний додаток встановлює, чи буде він надсилати дані з сенсорів для обробки, або отримувати інформацію про стан деякого іншого смартфона. Також тут встановлюється з'єднання з сервером;
2. **Стан роботи додатку.** Надається інформація про роботу системи;
3. **Попередження про небезпеку.** Надається інформація про можливу небезпечну подію.

Після того, як веб-сервіс був запущений, мобільний додаток може підключитись до нього, використовуючи для цього камеру та зчитувач QR-коду. У разі відсутності на мобільному пристрої камери користувач вводить IP-адресу сервера вручну. Як тільки було встановлено з'єднання між мобільним додатком та сервером користувач обирає один із режимів роботи додатку - приймач або відправник, та перенаправляється на екран стану роботи системи,

де, в залежності від обраного режиму роботи, відображається стан користувача іншого мобільного пристрою або стан наданих даних до веб-сервісу. Якщо мобільний додаток використовує режим «отримувач», та веб-сервіс визначив деяку небезпечну подію, додаток сповістить про це користувача та надасть необхідну інформацію, таку як час та місце виникнення події (Рис. 2.1.3.12).

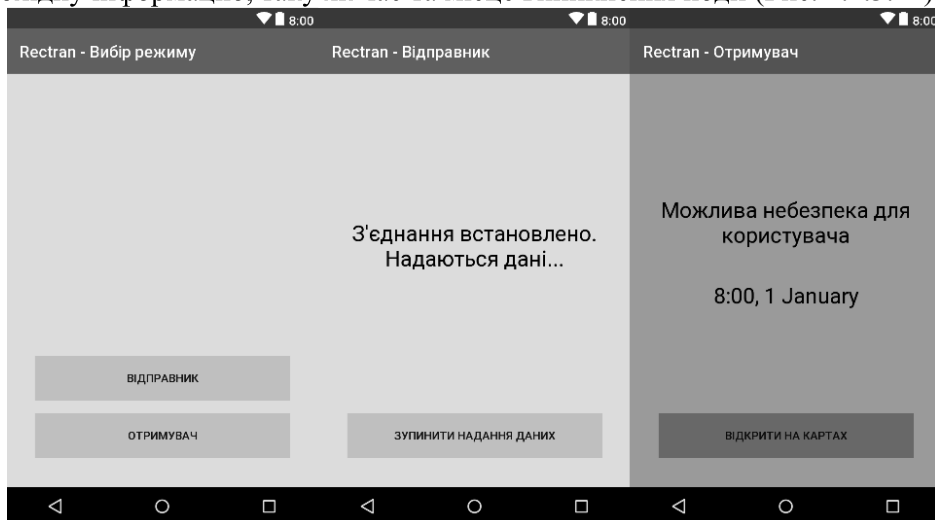


Рис. 2.1.3.12. Вигляд мобільного додатку.

2.1.4 Додаткові можливості покращення роботи системи

Мобільні пристрої переважно працюють в умовах, які супроводжуються шумом та деяким впливом, який не має відношення до дій користувача, що може ускладнити процес визначення події. Сенсори, котрі надають значення, мають певний діапазон можливих значень, який обмежує отримання можливої інформації. Також, через обмежені можливості мобільних пристроїв, дані з сенсорів можуть отримуватись із занадто великим інтервалом, що може вплинути на коректність виявлення небезпечних подій. Тому необхідно застосувати певні методи очищення даних та обійти обмеження сенсорів.

2.2 Очищення даних від шуму

Показання акселерометра і гіроскопа на мобільних пристроях містять складову шуму. Наприклад, похибка показань акселерометра в середньому досягає 0.05g.

На величину вихідного сигналу акселерометра впливають такі фактори:

- температура навколишнього середовища (температурні похибки);
- місце кріплення акселерометра;
- зовнішні магнітні поля (похибки від магнітного поля);
- вібрація і кутові коливання підстави (вібраційні похибки);
- частотні характеристики акселерометра (частотні похибки).

За допомогою спеціального програмного забезпечення отримуємо тестові дані з трьохосового акселерометра ADXL345 [59]. Дані являють собою файли у яких агреговано записи значень акселерометра - значення X,Y,Z координат та дельта часу запису. Приблизна частота зчитування даних: 50Гц (кожні 20 мс). Дані не були попередньо профільТРовані стандартними програмними засобами смартфона. Файл 1 містить 1015 записів координат акселерометра під час виконання різних дій, котрі не повторювались, на протязі 13с 800мс. Файл 2 містить записи координат акселерометра під час ходьби, що повторювалась на протязі 54с 960мс.



Рис. 2.2.1.1. Файл 1. Дані акселерометра під час довільних рухів.

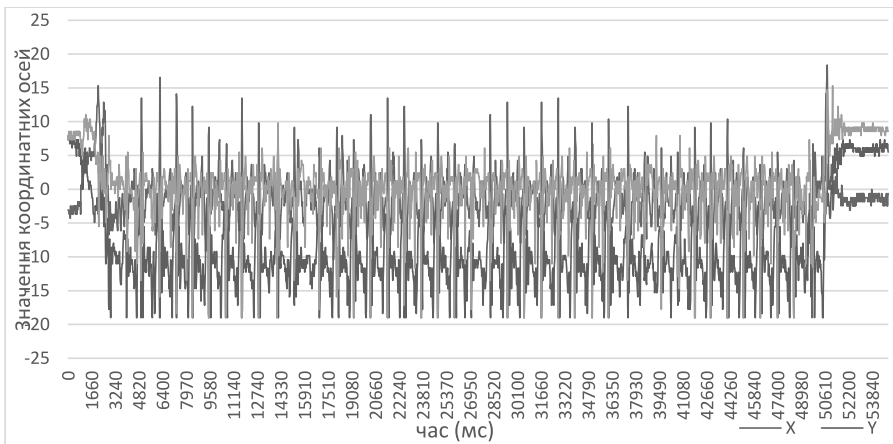


Рис. 2.2.1.2. Файл 2. Дані акселерометра під час ходьби.

2.2.1 Рухоме середнє

Метод середніх значень (Moving average, MA) один з найпростіших методів фільтрації шуму. Алгоритм дуже простий: на кожному кроці k , значення V_k вираховується, як середнє з n попередніх значень акселерометра, тобто:

$$V_k = \frac{\sum_{i=0}^{n-1} a_{k-i}}{n}$$

Для значень сигналу мають номер $k < n - 1$ значення.

$$V_k = \frac{\sum_{i=0}^k a_{k-i}}{k}$$

Такий метод дає при середніх значеннях n непогане згладжування, але має один істотний недолік - велику затримку в значеннях. Переваги методу: простота та гарне згладжування (Рис. 2.2.1.3, Рис. 2.2.1.4).

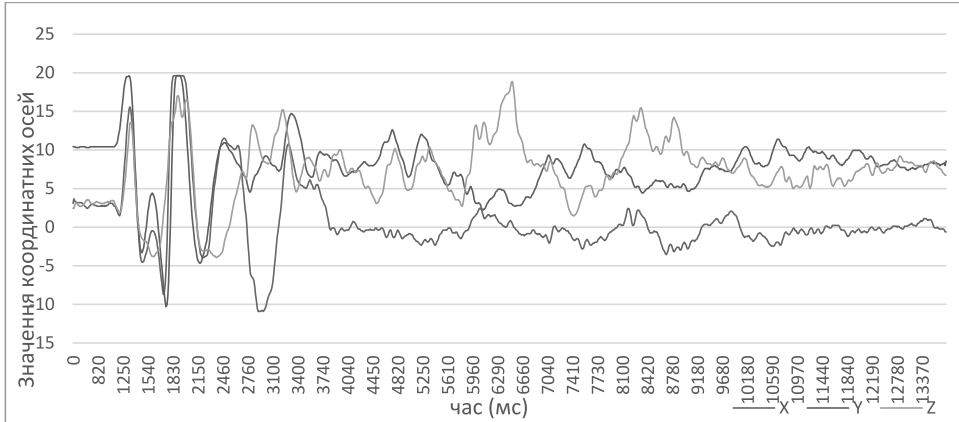


Рис. 2.2.1.3. Графік рухів після використання методу рухомого середнього.

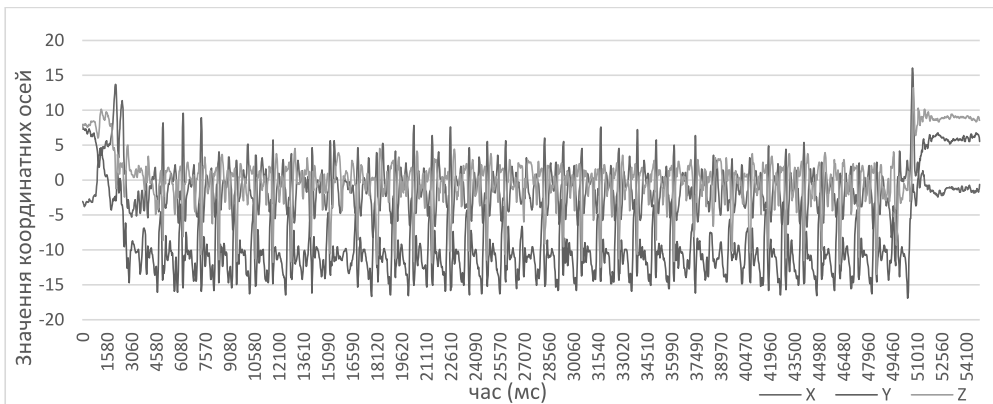


Рис. 2.2.1.4. Графік ходіння після використання методу рухомого середнього.

В цьому випадку метод дозволив полегшити обробку екстремумів функції - різких змін траєкторії (наприклад, ходіння) та моментів шумів - тряски телефону під впливом зовнішніх факторів, що в свою чергу дозволило більш впевнено, ніж в невідфільтрованому випадку, ідентифікувати різні етапи процесу ходьби.

2.2.2 Фільтри нижніх частот

Фільтри нижніх частот - це група фільтрів, основною особливістю яких є здатність фільтрувати сигнали вище зазначеної частоти, тобто такі фільтри пропускають сигнали низької частоти, що дозволяє позбутися від шумових перешкод сигналу. Фільтр нижніх частот пригнічує сигнали вище деякої критичної частоти w і пропускає сигнали нижче цієї частоти.

Найпростіший фільтр нижніх частот визначається наступною формулою:

$$O_n = O_{n-1} + a (I_n - O_{n-1}),$$

де O_n - вихідне значення сигналу (відфільтроване), I_n - вхідне значення (нефільтроване), a - коефіцієнт фільтрації, який приймає значення від 0 до 1. При a дорівнює 1, вихідні значення збігаються з вхідними.

Цей метод є класичним методом очищення сигналів від шуму. Він використовується в багатьох задачах і показує задовільні результати. Також існують і інші підходи до вирішення даної проблеми (наприклад, фільтр Калмана, та інші).

Для тих же даних було взято декілька різних коефіцієнтів: 0.1, 0.2 та 0.5. Розглянемо дані файлу 1, зібрані під час виконання різних рухів при різних коефіцієнтах фільтрації (Рис. 2.2.2.1, Рис. 2.2.2.2, Рис. 2.2.2.3):

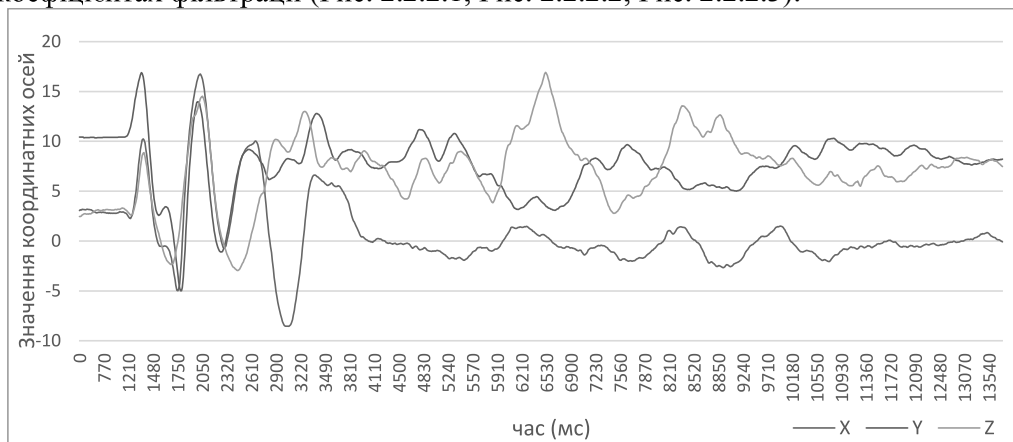


Рис. 2.2.2.1. Графік рухів після використання методу з коефіцієнтом 0.1.



Рис. 2.2.2.2. Графік рухів після використання методу з коефіцієнтом 0.2.

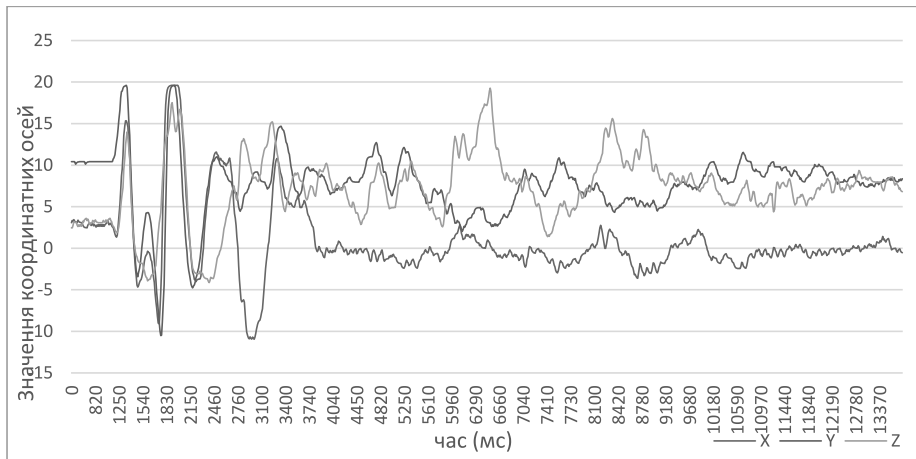


Рис. 2.2.2.3. Графік рухів після використання методу з коефіцієнтом 0.5.

Очевидно, що від зміни коефіцієнту фільтрації можна бачити помітну різницю в зменшенні шумів на графіку, але потрібно врахувати те, що хоч графік зі зменшенням коефіцієнту й стає більш плавним, але при цьому втрачає різкі зміни в даних, що може призвести до часткової втрати можливості аналізу даної ділянки.

В результаті надмірного зменшення коефіцієнту фільтрації (використання коефіцієнту в діапазоні 0.1 - 0.3 дає найкращий результат по згладжуванню екстремумів) алгоритм точно може визначити як саме здійснилася зміна напрямку руху телефону. Але, в той же час, даний діапазон коефіцієнтів (найбільш відчутно в околі 0.1) згладжує ділянки з шумами до рівня стану спокою. Тобто алгоритм може хибно обробити ділянки з швидкою зміною напрямку руху за знаходженням телефону в статичному положенні.

Тому найбільш вдалим рішенням буде використання середнього значення коефіцієнту фільтрації в проміжку від 0.0 - 1.0, що дасть помірне згладжування, але й збереже критичні деталі даних.

2.2.3 Аналіз результатів очищення даних від шуму

При використанні методу рухомого середнього з кроком в 5 елементів було отримано помітне покращення в згладжуванні функцій.

Використання фільтру нижніх частот так само дає більш гладкий результат для екстремумів та «шумів» функції, в залежності від зменшення коефіцієнту. Отриманий в результаті сигнал досить гладкий, але, так само як і при згладжуванні методом змінного середнього, присутня деяка затримка, особливо при різкому коливанні значень.

Покращення, описані вище, дозволили побачити на графіку деталі, які було неможливо визначити через надмірну кількість шумів.

В графіку ходіння можна прослідкувати та точно визначити, коли саме людина змінює ногу при ходьбі. На очищених даних також добре видно межі періоду ходьби - кількість мілісекунд в періоді. В свою чергу визначений темп ходьби дозволяє визначити приблизний вік людини та як саме людина рухається: біжить, підскакує, спокійно йде, та ін.

Також можна помітити згладжування функції на моментах екстремумів функції – моментів різкої зміни напрямку руху телефону та зменшенні кількості шумів, що зовсім не було видно до цього. Це дало можливість точно побачити межі періодичності дій та візуально краще сприймати вид діяльності, що виконується.

2.3 Розширення можливостей сенсорів

Для всіх сенсорів існують діапазони можливих значень:

Акселерометр: для трьох координат від -10.0 до ~10.0 м/ с²;

Сенсор магнітного поля: для трьох координат від -1000.0 до 1000.0 мкТл.;

Сенсор температури: від -273.1 до 100 градусів за Цельсієм;

Сенсор наближення: від 0 до 10 см.;

Сенсор натиску: від 0 до 1100 гПа.;

Сенсор освітлення: від 0 до 40 000 лк.;

Сенсор відносної вологості: від 0 до 100%.

Інтервал часу, за який мобільний пристрій може оптимально отримати один пакет даних від всіх сенсорів, не повинен бути меншим ніж 0.3 секунди, інакше є ризик отримати некоректні дані, зависання або аварійне завершення роботи програми для збору даних.

Для передачі даних через мережу, інтервал часу також не повинен бути меншим ніж 3 секунди, інакше знову виникає ризик отримати некоректні дані, зависання, аварійне завершення роботи програми також виникає ризик для Веб-Сервісу, який буде отримувати дані з різним інтервалом часу та може некоректно їх аналізувати та робити неправильні виводи.

2.3.1 Розширення діапазону максимально можливих значень

Як було розглянуто вище, всі сенсори мають обмеження на максимально можливе значення. В нашому випадку це склало суттєву проблему при аналізі прискорень. Справа в тому, що максимально можливе прискорення, яке фіксує акселерометр смартфона дорівнює $\pm 10 \text{ м/с}^2$, що відповідає приблизно перенавантаженню всього в $\pm 1\text{G}$ ($9,81 \text{ м/с}^2$). В той же час людський організм спроможний витримувати перенавантаження, наприклад при аварії, або при виконанні літаком фігур вищого пілотажу на рівні 4-7 G без втрати свідомості пілотом, для добре тренованого космонавта ця величина досягає 8-10 G, а звичайна людина може втратити свідомість, або отримати значні травми при перенавантаженнях вже від 1 до 3 G. Але такі ж значення перенавантаження людина може отримати і в звичайному житті.

Наприклад, дуже важливо оцінити значення можливого перенавантаження, яке отримає людина внаслідок нещасного випадку, наприклад, падіння з вікна багатоповерхівки.

Використовуючи формулу:

$$S = S_0 + V t + \frac{g t^2}{2},$$

можна обчислити час падіння, швидкість падіння та прискорення гальмування об землю.

За тією ж формулою можна обчислити прискорення приземлення з розрахунку, що швидкість, яку набрало тіло за час польоту, потрібно погасити на дистанції 1-1.5 метри (виходячи з росту людини).

В будь-якому випадку виникає потреба розширити діапазон прискорень, які вимірює акселерометр. Це можна зробити за допомогою додаткових аналітичних операцій.

Якщо прискорення кожні 0.1 секунд зростали на 5 м/с^2 , дійшли до обмеження, потім були «на упорі» 0.3 секунд і потім знов з той же швидкістю знизились до нуля, то це означає, що максимальне прискорення дорівнювало 20 м/с^2 . Тобто використовуючи рівняння

$$\frac{Y_3 - Y_1}{Y_2 - Y_1} = \frac{X_3 - X_1}{X_2 - X_1}$$

можна знайти значення, які були поза межею допустимих значень.

Наприклад, маємо деяку таблицю прискорень (Табл. 2.3.1.1). Використовуючи рівняння, можна знайти значення сенсора (Y_3) на моменті часу **0.3** ($X_3 = 0.3$). Для цього використаємо координати відомих значень на моменті часу **0.1** ($X_1 = 0.1$, $Y_1 = 5$) та **0.2** ($X_2 = 0.2$, $Y_2 = 10$):

$$\frac{Y_3 - 5}{10 - 5} = \frac{0.3 - 0.1}{0.2 - 0.1}, \quad \frac{Y_3 - 5}{5} = 2, \quad Y_3 = 15$$

Таким чином, ми отримали деяку пряму з лівого боку. Використавши це рівняння для пошуку значення на моменті часу **0.5** та координати відомих значень на моментах часу **0.6** та **0.7**, отримаємо другу пряму з правого боку, після чого знайшовши перетин цих двох прямих можна побудувати наступний

графік (Рис. 2.3.1.1). Також можливо отримати інші значення та на їх основі побудувати графік (Рис. 2.3.1.2):

Таблиця 2.3.1.1

Значення прискорень.

Час, сек	Прискорення за сенсором	Прискорення за аналітикою	Події
0	0	0	
0.1	5	5	
0.2	10	10	
0.3	10	15	Зашкалювання сенсору
0.4	10	20	Зашкалювання сенсору
0.5	10	15	Зашкалювання сенсору
0.6	10	10	
0.7	5	5	
0.8	0	0	

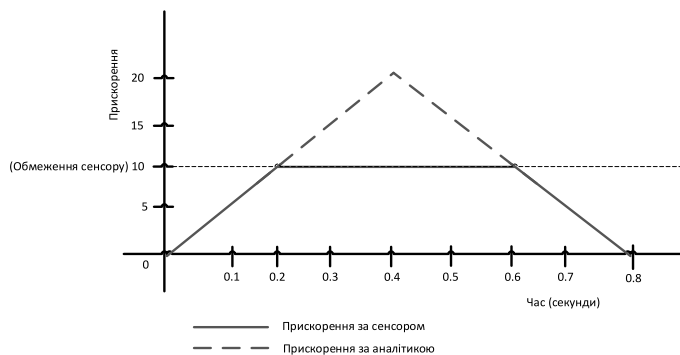


Рис. 2.3.1.1. Графік прискорень.

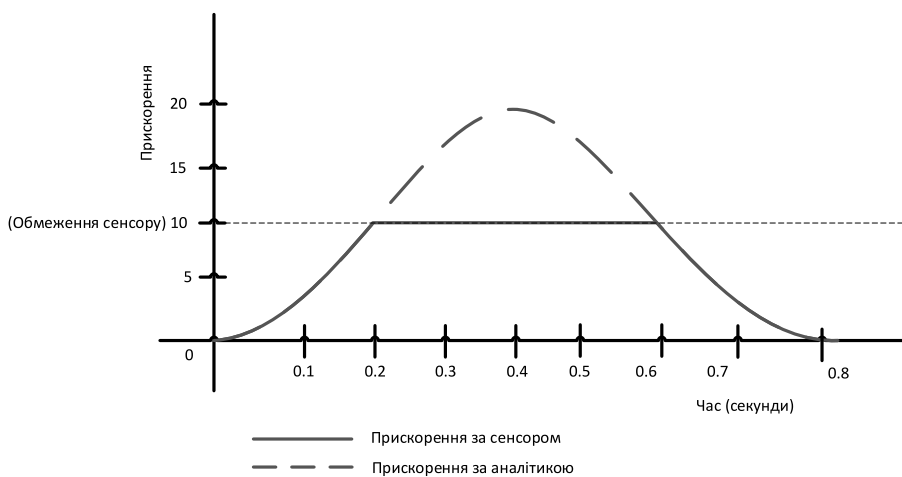


Рис. 2.3.1.2. Графік прискорень.

Таблиця 2.3.2.2

Значення прискорень.

Час, сек.	Прискорення за сенсором	Прискорення за аналітикою	Події
0	0	0	
0.1	3.2	3.2	
0.2	10	10.1	Зашкалювання сенсору
0.3	10	16.8	Зашкалювання сенсору
0.4	10	19.9	Зашкалювання сенсору
0.5	10	17.1	Зашкалювання сенсору
0.6	9.7	9.7	
0.7	2.6	2.6	
0.8	0	0	

Таким чином в роботі запропоновано підхід розширення діапазону чуттєвості сенсорів за допомогою аналітичної обробки даних. Це може допомогти в уникненні обмежень з діапазоном максимально можливих даних з сенсорів та в збільшенні їх інформативності.

2.3.2 Збільшення частоти зняття даних з сенсорів

Як було розглянуто вище, інтервал часу, з яким мобільний пристрій може оптимально отримати один пакет даних від всіх сенсорів, не повинен бути меншим ніж 0.3 секунди, інакше є ризик отримати некоректні дані, зависання або аварійне завершення роботи програми для збору даних, але може виникнути потреба отримувати значення з сенсорів частіше, наприклад, кожні 0.1 секунди. Для отримання такого результату, можна використати інтерполяцію між достовірно відомими значеннями. Наприклад для того, щоб визначити значення, які були між відомими моментами часу, можна використати **лінійну інтерполяцію**. Нехай відомо три значення, які були отриманні протягом однієї секунди (Рис. 2.3.2.1):

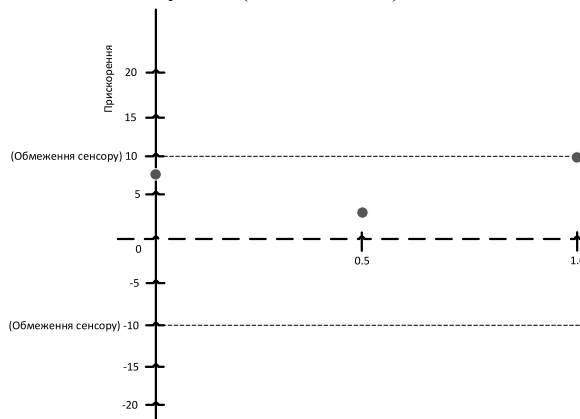


Рис. 2.3.2.1. Графік прискорень.

Табл. 2.3.2.3

Відомі значення прискорень.

Час (X)	0.0	0.5	1.0
Значення (Y)	7.5	2.5	9.5

Використовуючи формулу:

$$Y_2 = \frac{(X_2 - X_0) * (Y_1 - Y_0)}{(X_1 - X_0)} + Y_0$$

Можна отримати значення будь-якого моменту часу, між двома відомими значеннями. Наприклад, за попередніми даними, ми можемо отримати значення (Y_2) на моменті часу **0.25** ($X_2 = 0.25$), використовуючи координати відомих значень на моментах часу **0.0** ($X_0 = 0.0$, $Y_0 = 7.5$) та **0.5** ($X_1 = 0.5$, $Y_1 = 2.5$).

$$Y_2 = \frac{(0.25 - 0.0) * (2.5 - 7.5)}{(0.5 - 0)} + 7.5 = 5$$

Використавши даний метод для пошуку значення на моменті часу **0.75**, будуть відомі вже наступні значення (Табл. 2.3.4.4):

Таблиця. 2.3.2.4

Відомі значення прискорень.

Час (X)	0.0	0.25	0.5	0.75	1.0
Значення (Y)	7.5	5	2.5	6	9.5

Рекурсивно використовуючи даний метод можна отримати значення сенсору на ще менших діапазонах часу та побудувати наступний графік (Рис. 2.3.2.2):

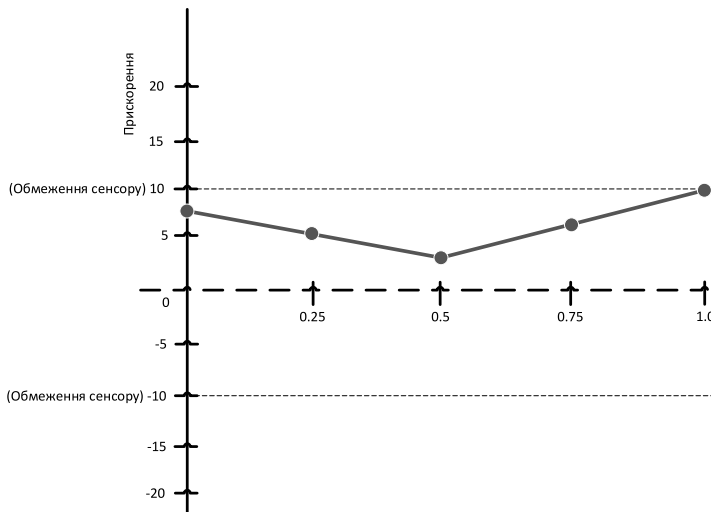


Рис. 2.3.2.2. Новий графік прискорень.

Також можна використати **сплайн-інтерполяцію**.

Для значень лінійного прискорення, які були отримані кожні 0.5 секунд, побудуємо наступний графік (Рис. 2.3.2.3):

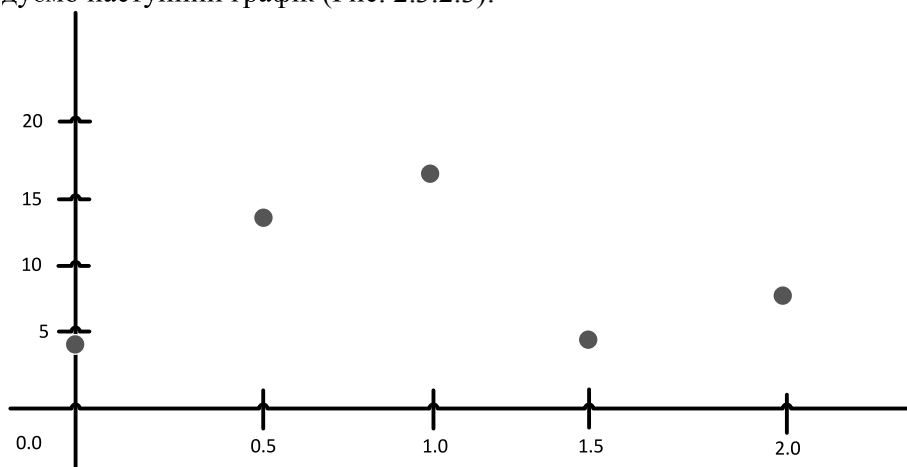


Рис. 2.3.2.3. Значення сенсорів з інтервалом 0.5 сек.

Після чого, використовуючи сплайн-інтерполяцію між достовірно відомими значеннями, побудувати графік:

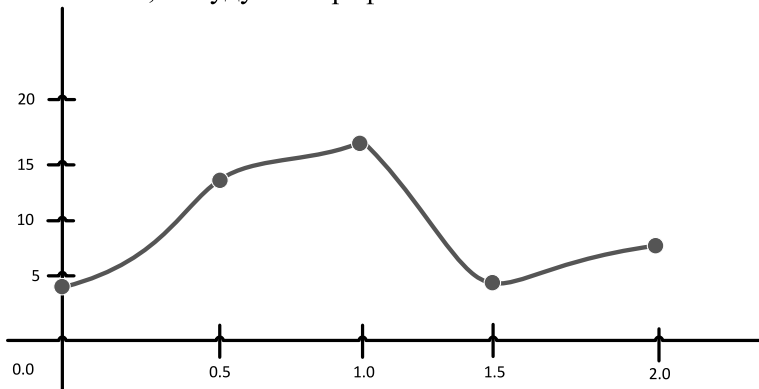


Рис. 2.3.2.4. Значення сенсорів після використання сплайн-інтерполяції.

Але може виникнути проблема, коли в такому графіку може «сховатися» різка зміна значень або, навпаки, з'явиться викид значень там, де його насправді не було. Наприклад, якщо зменшити крок збору даних з сенсорів з 0.5 секунд до 0.25 (що не рекомендується, через проблему описану вище), можна помітити, що виникають розриви значень (Рис. 2.3.2.5):

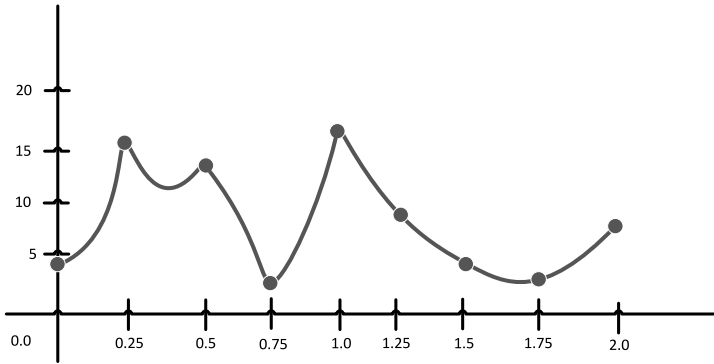


Рис. 2.3.2.5. Розриви значень.

Розглянемо ситуацію з падінням телефону. З реального мобільного пристрою з інтервалом 0.5 секунд були отримані деякі значення (Табл. 2.3.2.5).

Таблиця 2.3.2.5

Значення прискорень.

Час, сек	Прискорення за сенсором
0.0	0
0.5	0.6
1.0	6.8
1.5	-10
2.0	1.2
2.5	0.2

Перенесемо ці значення на графік (Рис. 2.3.2.6):

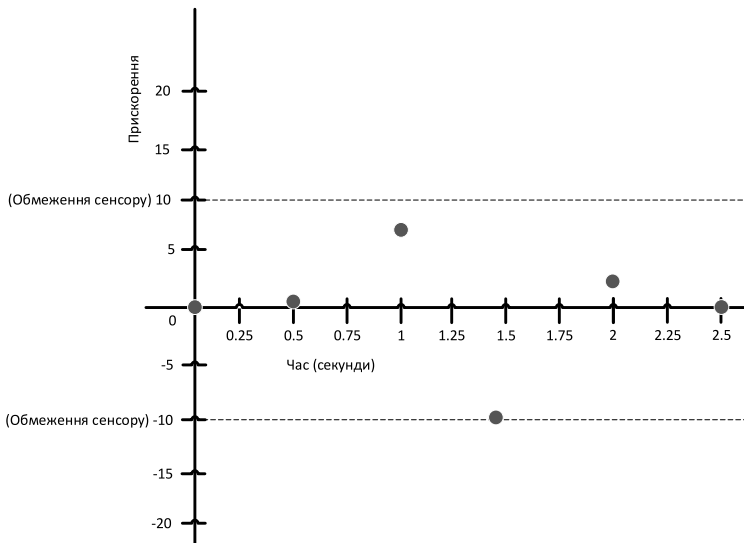


Рис. 2.3.2.6. Значення сенсорів з інтервалом 0.5 сек.

Після чого, використавши сплайн-інтерполяцію, можна отримати наступний графік (Рис. 2.3.2.7):

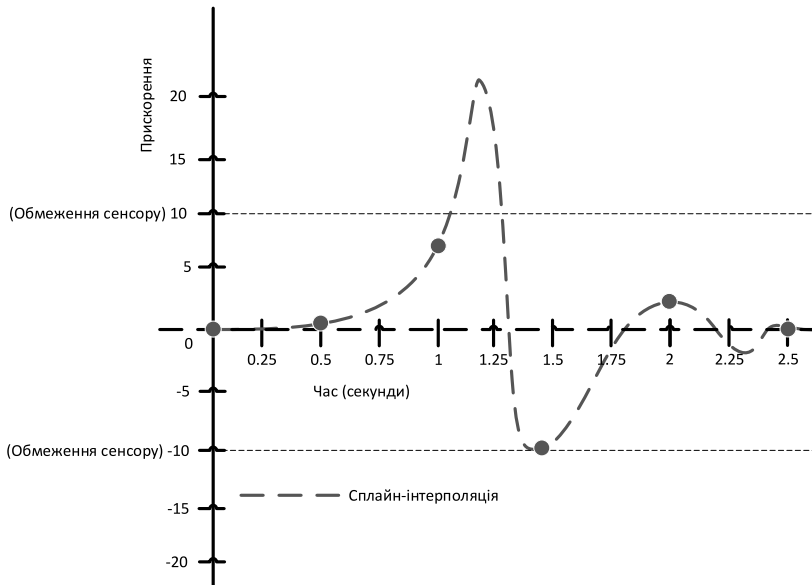


Рис. 2.3.2.7. Графік побудований сплайн-інтерполяцією.

Як бачимо сплайн інтерполяція може провокувати викиди значень, які сильно відрізняються від реальних відносно плавних змін значень. Тому, незважаючи на простоту методу, лінійна інтерполяція найчастіше є найкращим рішенням для визначення значень між відомими моментами часу.

Таким чином, можливо прибрати обмеження збору даних та отримати всі необхідні значення, але виникає навантаження на пристрій, який виконує інтерполяцію, що може спричинити уповільнення роботи системи.

Загалом можна сказати, що для збору даних частіше ніж кожні 0.5 секунд можна використовувати інтерполяцію, що може значно допомогти в отриманні більш інформативних даних, але ризик отримати некоректні значення все одно залишається.

Висновки до розділів 2.1-2.3

1. У розділах були розглянуті основні сенсори мобільних пристроїв, проаналізовано роботу з ними, визначені часові обмеження збору значень, обрані оптимальні технології для передачі даних та встановлені вимоги до програмного забезпечення. Було спроектовано та розроблено прототип системи, який відповідає клієнт-серверній архітектурі й встановленим вимогам, та виконує більшість необхідних функцій, таких як: отримання даних з сенсорів, їх формування в зручний формат, створення надійного каналу зв'язку для передачі інформації та її надсилання до веб-сервісу. Також було створено мобільний додаток, який працює із поштовим сервісом в якості приймача. Для

реалізації цих прототипів були використані технології Java, JSON, поштовий сервіс Gmail та бібліотека Gmail API.

2. Було визначено загальний алгоритм роботи системи, розглянуто основні задачі компонентів системи, таких як: мобільний додаток та веб-сервіс. Були обумовлені необхідні дії в разі виявлення можливої небезпеки. Також було спроектовано архітектуру системи, були розглянуті та обрані інструменти для реалізації системи. Розробка веб-сервісу була виконана із використанням мов програмування Java, JavaScript та технологій AJAX, JSON. За допомогою інструментів HTML, CSS був побудований веб-інтерфейс для взаємодії між користувачем та веб-сервісом. Були розглянуті характерні ознаки виявлення падіння та визначено алгоритм, за яким система зможе виявити небезпечні події. Також були розглянуті можливості виявлення положення мобільного пристрою та пересування користувача. Був розглянутий принцип роботи системи а також використання веб-сервісу й мобільного додатку.

3. Були проаналізовані обмеження сенсорів, діапазони їх можливих значень та чинники, що можуть вплинути на їх роботу та завадити процесу визначення небезпечних подій. Розглянуті методи очищення даних від шуму можуть допомогти отримати більш точні дані та виявити особливості змін значень, що не були помітні до фільтрації даних. Аналітична обробка даних та використання методів інтерполяції можуть допомогти обійти обмеження як діапазону можливих значень сенсорів, так і розмір кроку, з яким дані можуть бути зібрані зі всіх сенсорів. Таким чином, для покращення роботи системи були запропоновані методи збільшення точності надання інформації, що може допомогти з визначенням деяких особливостей наданих сенсорами значень та збільшити ймовірність коректного визначення небезпечних подій.

4. Небезпечні події, які можуть виникнути із людиною, досить різноманітні, тому визначити кожен з них та встановити, на основі яких показників вони виникають та навчити систему розпізнавати кожен з них може виявитись складною задачею. Більш раціональним буде виявлення спільних ознак безпеки, на основі яких і будуть робитися твердження, чи є загроза для здоров'я людини. В багатьох випадках основні такі ознаки – це падіння, момент удару та втрата свідомості.

Для виявлення цих ознак необхідні сенсори, якими обладнано більшість сучасних смартфонів. Багато досліджень та розробок було присвячено саме виявленню цих ознак безпеки, але не було розроблено загальної системи, яка ці значення аналізує та сповіщає про можливу небезпеку користувача мобільного пристрою. Тому із використанням сучасних інструментів розробки було спроектовано та реалізовано систему, яка за допомогою даних вбудованих сенсорів розпізнає ознаки безпеки та надає інформацію про можливу загрозу здоров'ю користувача.

Наступним кроком у розвитку системи буде застосування таких технологій, як машинне навчання та нейронні мережі для аналізу отриманих даних та виявлення небезпечних подій.

2.4 Програмне забезпечення аналізу та прогнозування стану людини під час тренувань за допомогою сенсорів мобільних пристроїв

2.4.1 Вступ

Одним із популярних видів спорту є біг. На ринку існує велика кількість мобільних застосувань для бігу: «Strava», «Nike Run Club», «Polar Beat», «Garmin Connect», «Runtastic», «Map My Run». Більшість з них дозволяють відслідковувати пробіжки та надають багато даних та аналітики. Проте жодне з цих застосувань не надає можливості побачити динаміку тренувань і прогрес бігуна. Саме на цьому аспекті було вирішено сконцентруватись під час написання даної роботи. Після дослідження потреб аматорських та професіональних бігунів, було визначено питання, на які було б цікаво отримати відповіді:

- На якому саме рівні підготовки зараз знаходиться спортсмен?
- Наскільки добре та ефективно поставлений процес тренувань?
- За який час спортсмен може досягти певних результатів, враховуючи його поточний рівень підготовки?

Для того, щоб відповісти на ці питання потрібно проаналізувати дані про пробіжки користувача за останній час. Більшість бігунів зараз вже використовують застосування, які відстежують біг завдяки вбудованим сенсорам мобільних пристроїв. Деякі з програм перелічених вище мають можливість експорту даних. Проте деякі з них належать компаніям-виробникам спортивного екіпірування, наприклад: «Nike Run Club» — Nike Inc., «Runtastic» — Adidas AG, «Map My Run» — Under Armour Inc, «Polar Beat» — Polar Electro, «Garmin Connect» — Garmin Ltd. Це значною мірою впливає на аудиторію користувачів — шанувальники конкретного бренду користуються його застосуванням. Проте виділяється одне застосування — «Strava», яке не належить жодному бренду і є популярним серед спортсменів різного рівня, від любителів до професіоналів. Саме його було вирішено використовувати в рамках дослідження, щоб реалізувати можливість експорту даних. Маючи достатню кількість даних про пробіжки спортсмена за останній час можливо отримати відповіді на усі запитання поставлені вище.

Мета і задачі дослідження

Метою роботи є оцінити ефективність бігових тренувань та визначити прогрес рівня підготовки, шляхом аналізу пробіжок користувача. А також спрогнозувати результати фінішування на певній дистанції, враховуючи поточний рівень підготовки спортсмену.

2.4.2 Постановка задачі

Досягнення мети включало розв'язання таких задач:

- 1) Вивчення існуючих програмних систем для бігу з метою виявлення їх функціональних можливостей.

2) Дослідження існуючих способів оцінки ефективності бігових тренувань.

3) Дослідження існуючих способів прогнозування результатів забігу.

4) Реалізація авторизації користувача у системі та експорту даних про пробіжки із системи «Strava».

5) Обробка експортованих даних (GPX файлів) з метою представлення у вигляді об'єктних моделей для подальшого опрацювання.

6) Реалізація обчислення основних характеристик кожної пробіжки (дистанція; тривалість; швидкість; темп; набір висоти; мінімальне, середнє та максимальне серцебиття).

7) Реалізація визначення зон серцебиття для кожної пробіжки.

8) Реалізація алгоритму прогнозування результатів на конкретній дистанції, враховуючи поточний рівень підготовки спортсмену.

9) Реалізація визначення ефективності тренувань та прогресу завдяки аналізу пробіжок за останній період

10) Представлення прогресу тренувань у вигляді графіка.

Об'єкт дослідження - є процес бігових тренувань спортсмену, дані про який будуть експортовані з сторонньої системи «Strava».

Предмет дослідження - метод бігових тренувань для визначення ефективності та прогресу, а також модель прогнозування результатів на конкретній дистанції.

Методи дослідження:

- Порівняння — для встановлення відмінностей однієї пробіжки від іншої.
- Розрахунок — для визначення чисельних характеристик кожної пробіжки.
- Аналіз — для вивчення процесу тренувань. Даний метод передбачав розбиття загального процесу на окремі складові — пробіжки, для їх подальшої обробки.
- Синтез — з'єднання окремих складових (пробіжок) для визначення прогресу на ефективності процесу тренувань.

Новизна одержаних результатів

Запропонований метод визначення рівня підготовки спортсмену та оцінки ефективності та прогресу тренувань на основі використання показників VDOT та VO2Max.

Запропонований алгоритм аналізу бігових тренувань та прогнозування результатів.

Практичне значення одержаних результатів

Була виконана програмна реалізація вищеперерахованих методів аналізу та прогнозування. Було розроблене програмне забезпечення, що може використовуватись аматорами та професійними бігунами для оцінки ефективності процесу тренувань, для ознайомлення з прогресом рівня підготовки, для підбору відповідного темпу для участі на змаганнях.

2.4.3 Аналіз об'єкту дослідження

Формат GPX

Для експорту даних про пробіжки буде використовуватись застосування для бігу Strava, яке надає можливість через API отримати список пробіжок користувача у вигляді GPX файлів. GPX (GPS eXchange Format) — це текстовий формат зберігання і обміну даними GPS, заснований на XML. GPX є вільним форматом і може бути використаний без будь-яких ліцензійних відрахувань. Формат дозволяє зберігати інформацію про орієнтири (waypoints), маршрути (routes) і треки (tracklogs). Для кожної точки зберігаються її довгота, широта і висота над рівнем моря (якщо є інформація про висоту). Для точок треку зберігається також час проходження точки. XML-схема передбачає також зберігання довільної інформації користувачів по кожній точці, обов'язковими є тільки довгота і широта. Прикладом для користувача інформації може бути висота місцевості (ele), швидкість, частота пульсу (hr), число кроків, темп обертання педалей (каденс) (cad), температура і т.п. Фрагмент GPX файлу, в якому описано одну із точок маршруту, експортованого з Strava:

```
<trkpt lat="50.4602820" lon="30.5230720">
<ele>107.0</ele>
<time>2019-04-07T06:09:21Z</time>
<extensions>
<gpstpx:TrackPointExtension>
<gpstpx:hr>160</gpstpx:hr>
</gpstpx:TrackPointExtension>
</extensions>
</trkpt>
```

Для того, щоб відобразити характеристики пробіжки потрібно розпарсити даний GPX файл. Це можна реалізувати самостійно, використовуючи клас “NSXMLParser”, який входить до фреймвору Foundation. Або використати готову реалізацію, наприклад “CoreGPX” (<https://github.com/vincentneo/CoreGPX>). В програмній реалізації використано саме цей метод.

Дистанція пробіжки

Для того, щоб отримати дистанцію потрібно розрахувати відстань між усіма точками маршруту, використовуючи їх координати (latitude, longitude). А потім додати отримані відстані.

$$Total\ distance = \sum_{i=1}^n distance(x_{i-1}, x_i)$$

Для того, щоб розрахувати відстань між двома точками, які мають координати (lat1, lon1) та (lat2, lon2) можна використати формулу Veronika Hubeny:

$$Distance = \sqrt{latdiff^2 * meridian^2 + londiff^2 * n^2 * (\cos latave)^2}$$

Де:

$$latdiff = (lat1 - lat2) * \frac{\pi}{180}$$

$$londiff = (lon1 - lon2) * \frac{\pi}{180}$$

$$meridian = \frac{a * f2}{w2 * \sqrt[2]{w2}}$$

$$a = 6378137.0$$

$$b = 6356752.314245$$

$$f2 = \frac{b^2}{a^2}$$

$$w2 = 1 - (\sin latave)^2 * (1.0 - f2)$$

$$latave = 0.5 * (lat1 + lat2) * \frac{\pi}{180}$$

$$n = \frac{a}{\sqrt[2]{w2}}$$

Тривалість пробіжки

Для того, щоб отримати тривалість потрібно розрахувати різницю у часі між усіма точками маршруту, а потім додати отримані інтервали.

$$Total\ duration = \sum_{i=1}^n time\ interval(x_{i-1}, x_i)$$

Пульс

Для того, щоб отримати значення мінімального, середнього та максимального пульсу, треба проаналізувати пульс у кожній точці маршруту. А потім знайти відповідне значення. Для визначення середнього пульсу використовується середнє арифметичне усіх значень.

Набір висоти

Існує декілька способів розрахунку набору висоти:

1 метод — сукупний (кумулятивний) набір висоти — передбачає сумування лише додатної різниці висоти на кожному відрізку (враховуючи усі точки відрізка). Такий метод не враховує втрату висоти (elevation loss).

2 метод — передбачає врахування як набору (elevation gain) так і втрати висоти (elevation loss).

3 метод — різниця висоти між початковою і кінцевою точкою маршруту — цей метод є найменш точним.

1 метод є найбільш розповсюдженим і найбільш використовуваним, тому саме він буде використаний в дипломній роботі.

Зони пульсу (серцебиття)

ЧСС — частота серцевих скорочень, показник того, скільки ударів робить серце в хвилину. Багато тренувальних планів зараз відштовхуються від зон інтенсивності, які визначаються відносно ЧСС. Такі зони розраховують індивідуально. В ідеальному варіанті, розрахунок «своїх» зон визначить індивідуальний тест з функціональної діагностики серця з консультацією кардіолога. Найменше, що можна зробити - розрахувати їх самостійно. У розрахунках відштовхуються від пульсу в стані спокою і МЧСС. Існує декілька способів для визначення МЧСС (максимальної частоти серцевих скорочень).

Загальна формула для чоловіків і жінок:

$$\text{МЧСС} = 220 - \text{вік}$$

Більш точна індивідуальна формула:

$$\text{Для чоловіків: МЧСС}_{\text{для чоловіків}} = 214 - (0,8 * \text{вік})$$

$$\text{Для жінок: МЧСС}_{\text{для жінок}} = 209 - (0,9 * \text{вік})$$

Найбільш вірний спосіб, з найточнішими показниками, проте вимагає часу і сил — обчислення МЧСС біговим тестом.

Знаючи МЧСС можна розрахувати зони пульсу — які представлені у табл. 2.4.3.1.

Таблиця 2.4.3.1.

Зони пульсу.

	Назва зони	Діапазон	Рівень інтенсивності
Z1	Спокійна (Endurance)	50-60 %	Дуже низький
Z2	Середня (Moderate)	60-70 %	Низький
Z3	Ритмічна (Tempo)	70-80 %	Середній
Z4	Порогова (Threshold)	80-90 %	Інтенсивний
Z5	Анаеробна (Anaerobic)	90-100 %	Максимальний

Показник МСК (VO2max)

Термін «максимальне споживання кисню» вперше був описаний і використаний Hill і Herbst в 1920-х роках. Основні положення теорії VO2max говорили:

- Існує верхня межа споживання кисню,
- Існує природна різниця в значеннях VO2max,
- Високий VO2max необхідний для успішної участі в забігах на середні і довгі дистанції,
- VO2max обмежений здатністю серцево-судинної системи переносити кисень до м'язів.

Показник VO2max характеризує максимальну кількість використовуюваного кисню, і розраховується шляхом віднімання кількості видихнути кисню з кількості поглиненого кисню. Оскільки VO2max використовується для кількісного опису ємності аеробного системи, показник знаходиться під впливом великої кількості факторів на довгому шляху кисню від навколишнього середовища до мітохондрій в м'язах.

Показник VDOT

Щоб точно визначити показник МСК (VO_{2Max}) потрібно пройти обстеження в лабораторії, що може виявитись недешево. Проте існує інший показник, який називають псевдо-МСК — VDOT.

VDOT — це значення розраховане на основі результатів змагань. Даний показник був придуманий Джеком Деніелсом і Джимом Гілбертом у 1970 роках.

Формула для визначення VDOT враховує:

- Залежність споживання кисню від швидкості бігу.
- Залежність швидкості при максимальному споживанні кисню від тривалості забігу.

На основі цих формул були представлені таблиці VDOT. В табл. 2.4.3.2 приведено скорочену версію таблиць VDOT.

Таблиця 2.4.3.2.

Значення VDOT для результатів змагань на стандартних дистанціях

VDOT	1500 м	Миля	3000 м	2 миля	5000 м	10000 м	15000 м	Полу-марафон	Марафон
30	0:08:30	0:09:11	0:17:56	0:19:19	0:30:40	1:03:46	1:38:14	2:21:04	4:49:17
31	0:08:15	0:08:55	0:17:27	0:18:48	0:29:51	1:02:03	1:35:36	2:17:21	4:41:57
32	0:08:02	0:08:41	0:16:59	0:18:18	0:29:05	1:00:26	1:33:07	2:13:49	4:34:59
33	0:07:49	0:08:27	0:16:33	0:17:50	0:28:21	0:58:54	1:30:45	2:10:27	4:28:22
34	0:07:37	0:08:14	0:16:09	0:17:24	0:27:39	0:57:26	1:28:30	2:07:16	4:22:03
35	0:07:25	0:08:01	0:15:45	0:16:58	0:27:00	0:56:03	1:26:22	2:04:13	4:16:03
36	0:07:14	0:07:49	0:15:23	0:16:34	0:26:22	0:54:44	1:24:20	2:01:19	4:10:19
37	0:07:04	0:07:38	0:15:01	0:16:11	0:25:46	0:53:29	1:22:24	1:58:34	4:04:50
38	0:06:54	0:07:27	0:14:41	0:15:49	0:25:12	0:52:17	1:20:33	1:55:55	3:59:35
39	0:06:44	0:07:17	0:14:21	0:15:29	0:24:39	0:51:09	1:18:47	1:53:24	3:54:34
40	0:06:35	0:07:07	0:14:03	0:15:08	0:24:08	0:50:03	1:17:06	1:50:59	3:49:45

Визначення рівня підготовки

Вважається, що для кожної захопленої бігом людини, показник VO_{2max} (або VDOT у Деніелса) фактично визначає його талант або потенціал. Величина VO_{2max} визначає максимальне споживання кисню (МПК), і це один з найбільш часто використовуваних показників для відстеження прогресу в тренуваннях. Таким чином, у кожного бігуна є базове значення VDOT - одне число, за допомогою якого можна легко порівнювати форму різних бігунів або форму одного бігуна в різні моменти часу. Ця система також виявилася ідеальною для визначення рівня інтенсивності тренування, так як інтервальні і порогові вправи і навіть легкий тривалий біг і біг в марафонському темпі можна проводити при певних частках (відсотках) VDOT бігуна.

Прогнозування результатів забігу

Таблиці VDOT допомагають розраховувати і передбачати результати на будь-яких дистанціях, якщо відомий результат хоча б на одній.

2.4.4 Використані програмні засоби

На даний момент існує багато мобільних операційних систем. Найбільш популярними серед них є: iOS (Apple), Android (Google). Вибір операційної системи залежить від потреб користувача. Вибір мови програмування залежить від поставленої задачі. Вибір середовища програмування залежить від потреб програміста і, можливо, від його особистих переконань. Зараз існує багато середовищ програмування для будь-якої мови:

- C/C++:
 - Microsoft Visual Studio (Microsoft);
 - C++ Builder (Embarcadero Technologies);
 - NetBeans IDE (NetBeans Community);
 - QT;
- Java:
 - IntelliJ IDEA (JetBrains);
 - Eclipse (Eclipse Foundation);
 - NetBeans IDE (NetBeans Community);
- Objective-C, Swift:
 - Xcode (Apple);
 - AppCode (JetBrains).

Операційна система. В якості операційної системи, для якої була реалізована програма, обрана операційна система для смартфонів та електронних планшетів, що розробляється і випускається американською компанією Apple - iOS.

Головними перевагами даної ОС є:

- Оптимізованість — досягається завдяки тому, що розробники заздалегідь знають апаратні характеристики групи пристроїв, на яких буде працювати iOS;
- Безпечність та надійність. Для мобільних пристроїв, вироблених фірмою Apple, не існує вірусів. Так як на пристрій застосування може потрапити лише через магазин застосувань AppStore, а в AppStore застосування потрапляє тільки після перевірки Apple.

Середовище програмування. Xcode — це інтегроване середовище розробки програмного забезпечення під OS X та iOS, розроблене корпорацією Apple. Xcode включає в себе більшу частину документації розробника від Apple і Interface Builder - застосування, що використовується для створення графічних інтерфейсів. Пакет Xcode включає в себе змінену версію вільного набору компіляторів GNU Compiler Collection і підтримує мови C, C ++, Objective-C, Swift, Java, AppleScript, Python і Ruby з різними моделями програмування, включаючи (але не обмежуючись) Cocoa, Carbon і Java.

Swift — це відкрита мультипарадигмальна об'єктно-орієнтована мова програмування загального призначення. Мова створена компанією Apple в першу чергу для розробників iOS та MAC OS X.

Розробка поточного варіанту мови Swift почалася в 2010 році Крісом Латтнером, керівником відділу розробки інструментів для створення програмного забезпечення Apple і одним з основних розробників LLVM. Swift запозичив ідеї з «Objective-C, Rust, Haskell, Ruby, Python, C #, CLU, і ще з стількох багатьох мов, що важко перелічити». Swift працює з фреймворками Cocoa, Cocoa Touch і є сумісним із основною кодовою базою Apple, написаною на мові Objective-C.

Objective-C — це компільована об'єктно-орієнтована мова програмування, що використовується корпорацією Apple. Objective-C побудований на основі мови Сі та парадигм Smalltalk. Зокрема, об'єктна модель побудована в стилі Smalltalk - тобто об'єктам надсилаються повідомлення. Мова Objective-C є над множиною мови С, тому С-код повністю зрозумілий компілятору Objective-C.

2.4.5 Вимоги до апаратного та програмного забезпечення

Для роботи з даним програмним продуктом необхідно мати наступне програмне та апаратне забезпечення:

1) Будь-який мобільний пристрій компанії Apple:

iPhone 5s,
 iPhone 6, iPhone 6 Plus,
 iPhone 6s, iPhone 6s Plus;
 iPhone SE;
 iPhone 7, iPhone 7 Plus;
 iPhone 8, iPhone 8 Plus;
 iPhone X;
 iPhone XS, iPhone XS Max;
 iPhone XR.

2) Операційна система: iOS 11.0 та вище.

2.4.6 Створення та налагодження програми

Створення та налагодження програми виконувалося в IDE Xcode — це інтегроване середовище розробки програмного забезпечення виробництва компанії Apple. В ході розробки використано концепцію MVC (Model View Controller). MVC поділяє систему на три частини: модель даних, вигляд даних та керування. Застосовується для відокремлення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача. За представлення в системі відповідають Storyboard файли. Storyboard — це візуальне представлення користувацького інтерфейсу програми, яке показує екрани контенту, які називаються сценами і переходи між ними (Рис. 2.4.6.1).

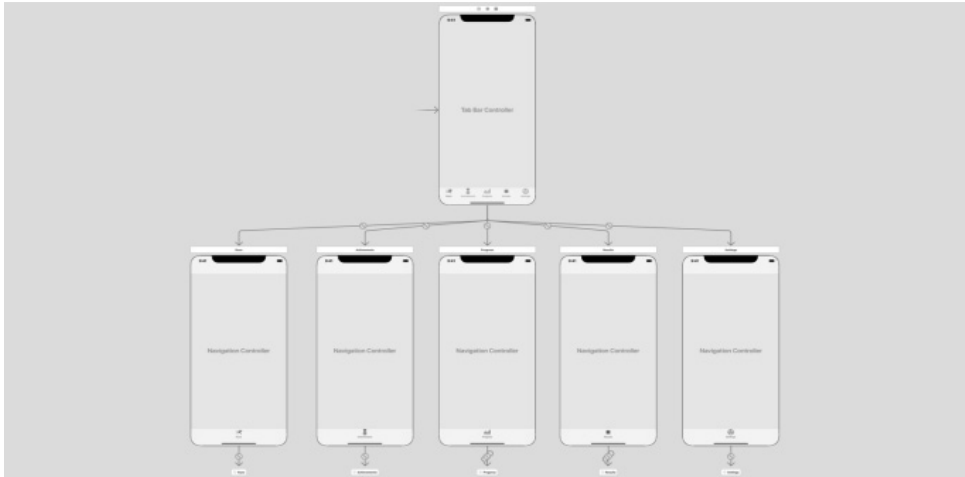


Рис. 2.4.6.1. Файл Main.storyboard.

На Рис. 2.4.6.1 представлено файл Main.storyboard в якому міститься головне меню застосування у вигляді Tab Bar. Кожен пункт меню має власну іконку та назву. А для представлення змісту кожного пункту використовується окремий Storyboard файл:

- «Пробіжки» («Runs») — Runs.storyboard — рисунок 2.4.6.2;
- «Досягнення» («Achievements») — Achievements.storyboard — рисунок 2.4.6.3;
- «Прогрес» («Progress») — Progress.storyboard — рисунок 2.4.6.4;
- «Результати» («Results») — Results.storyboard — рисунок 2.4.6.5;
- «Налаштування» («Settings») — Settings.storyboard — рисунок 2.4.6.6.

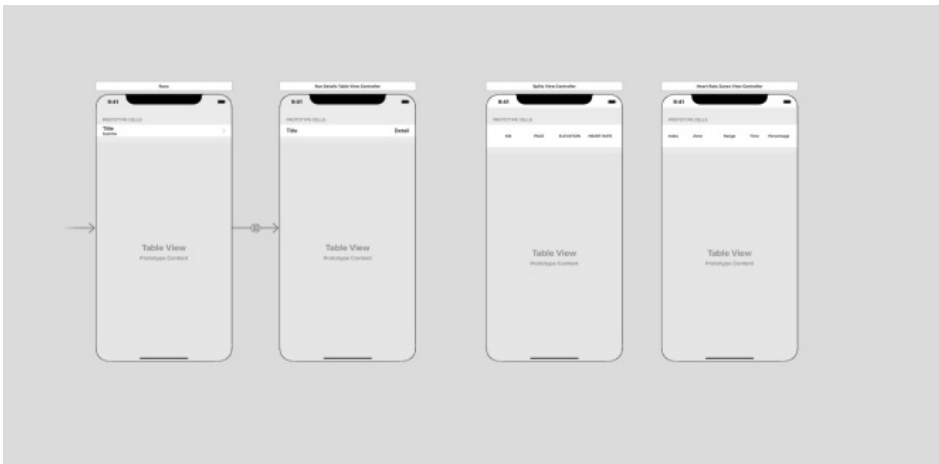


Рис. 2.4.6.2. Файл Runs.storyboard.

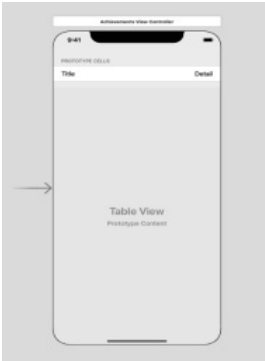


Рис. 2.4.6.3.

Файл Achievements.storyboard.

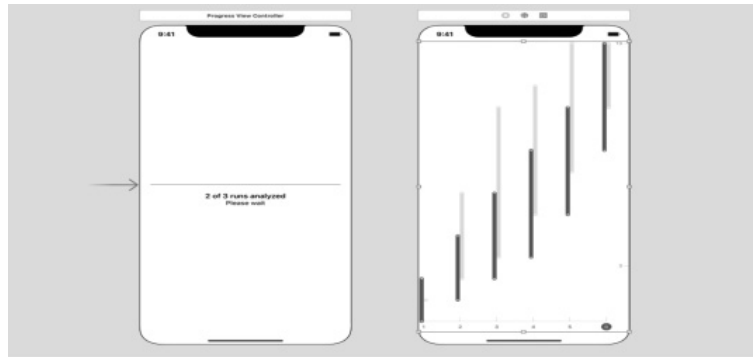


Рис. 2.4.6.4.

Файл Progress.storyboard.

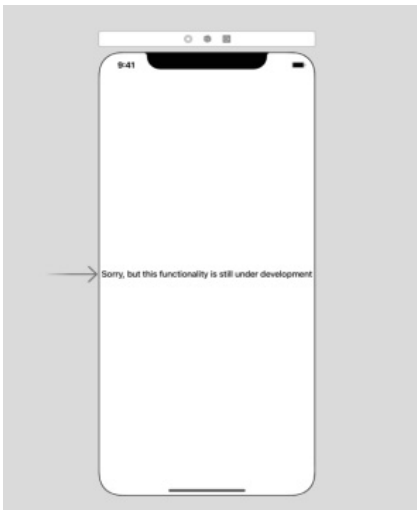


Рис. 2.4.6.5.Файл Results.storyboard.

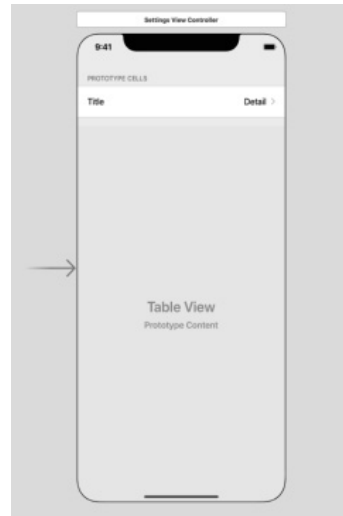


Рис. 2.4.6.6. Файл Settings.storyboard.

Для реалізації проекту було використано наступні фреймворки:

- UIKit Framework — бібліотека, що забезпечує ключову інфраструктуру, необхідну для побудови та управління iOS-застосуванням. Ця структура забезпечує вікно, архітектуру представлення, необхідну для управління користувацьким інтерфейсом застосування; інфраструктуру обробки подій, необхідну для реагування на дії користувача, а також модель застосування, необхідну для запуску головного циклу програми та взаємодії з системою.
- Alamofire — бібліотека для роботи з HTTP-networking.
- CoreGPX — бібліотека для парсингу та створення GPX файлів місцезнаходження. Це Open-source бібліотека, яка розповсюджується за MIT License.

Розробка ієрархії класів

Архітектура застосування була спроектована згідно патерну MVC (представлено на UML-діаграмі пакетів — рис. 2.4.6.7).

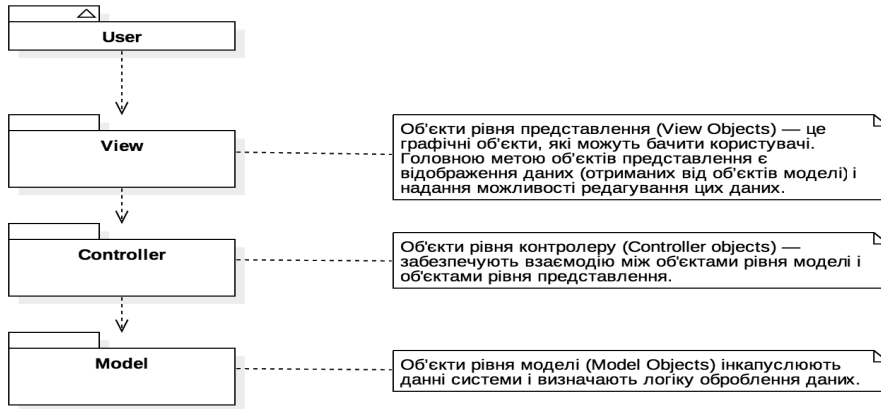


Рис. 2.4.6.7. UML-діаграма пакетів.

На рис. 2.4.6.8, 2.4.6.9 та 2.4.6.10 зображено діаграми класів рівня контролеру. Усі класи цього рівня є підкласами *UIViewController* (забезпечується фреймворком UIKit Framework).

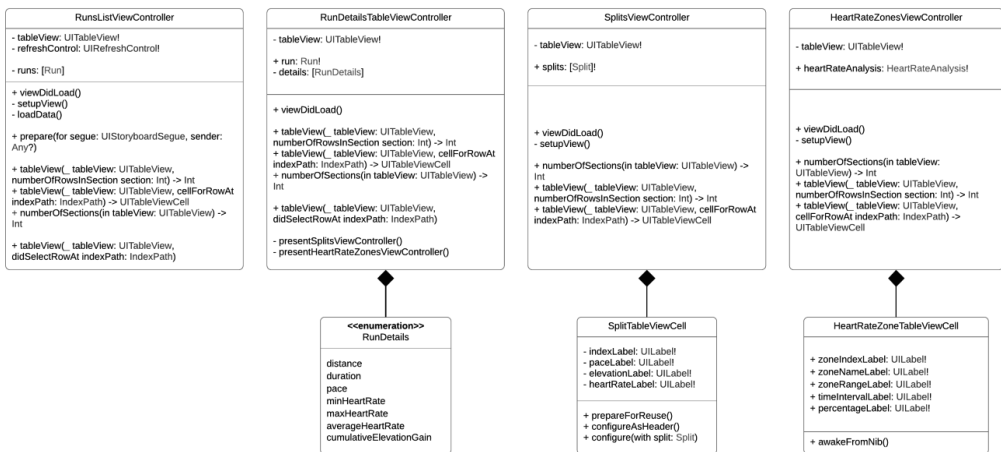


Рис. 2.4.6.8. UML діаграма класів рівня контролеру (1).

Клас *RunsListViewController* (сцена «Список пробіжок») — використовується для представлення списку усіх пробіжок користувача у вигляді таблиці (*UITableView*), які були експортовані із Strava. Для відображення кожної окремої пробіжки використовується стандартна комірка (*UITableViewCell*), на якій відображається загальна інформація: назва і дата. Після натискання на комірку користувач буде перенаправлений на сцену «Деталі пробіжки» (*RunDetailsTableViewController*).

Клас *RunDetailsTableViewCellController* (сцена «Деталі пробіжки») — використовується для представлення детальної інформації про пробіжку у вигляді таблиці (*UITableView*):

- дистанція; тривалість; темп; мінімальний пульс; середній пульс; максимальний пульс; набір висоти.

Для представлення кожної характеристики використовується окрема стандартна комірка (*UITableViewCell*). Також на даній сцені реалізована можливість перейти до сцен «Відрізки» (*SplitsViewController*) та «Зони пульсу» (*HeartRateZonesViewController*), шляхом натискання на відповідну комірку таблиці.

Клас *SplitsViewController* (сцена «Відрізки») — використовується для представлення інформації по кожному кілометровому відрізку пробіжки. Список відрізків представлено у вигляді таблиці (*UITableView*), а в якості комірок використовуються об'єкти класу *SplitTableViewCell*. У кожній комірці відображається порядковий номер відрізка, темп, набір висоти, середній пульс.

Клас *HeartRateZonesViewController* (сцена «Зони пульсу») — використовується для представлення зон пульсу для поточної пробіжки. Список доступних зон пульсу описаний в enumeration *HeartRateZone*. Зони пульсу відображаються у вигляді таблиці (*UITableView*), для кожної зони використовується окрема комірка — *HeartRateZoneTableViewCell*. У кожній комірці відображається:

- порядковий номер зони пульсу;
- назва зони пульсу;
- діапазон значень для поточного користувача (мінімальне та максимальне);
- час, який користувач провів у цій зоні;
- відсоток — відношення часу, який користувач провів у цій зоні, до загальної тривалості пробіжки.

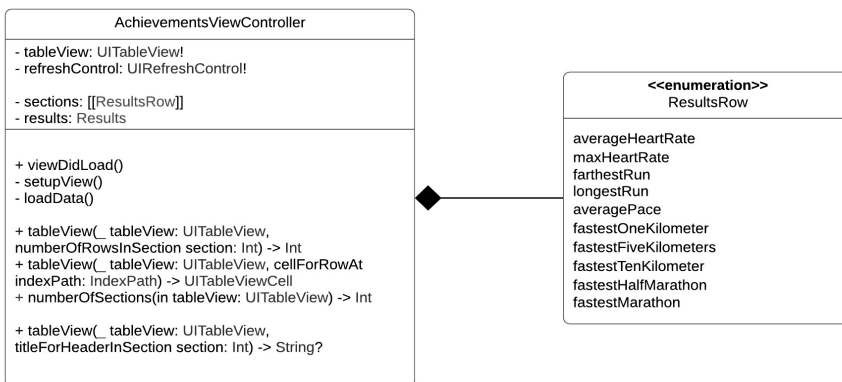


Рис. 2.4.6.9. UML діаграма класів рівня контролера (2).

Клас *AchievementsViewController* (сцена «Досягнення») — використовується для представлення досягнень користувача. Інформація представляється у вигляді таблиці (*UITableView*). Для відображення кожного досягнення використовується окрема комірка — об'єкт стандартного класу *UITableViewCell*. Усі дані, що відображаються на цій сцені розраховуються шляхом аналізу усіх пробіжок користувача:

- середній пульс;
- максимальний пульс;
- найдовша пробіжка;
- найтриваліша пробіжка;
- середній темп;
- найшвидший 1 км;
- найшвидші 5 км;
- найшвидші 10 км;
- найшвидший напівмарафон;
- найшвидший марафон.

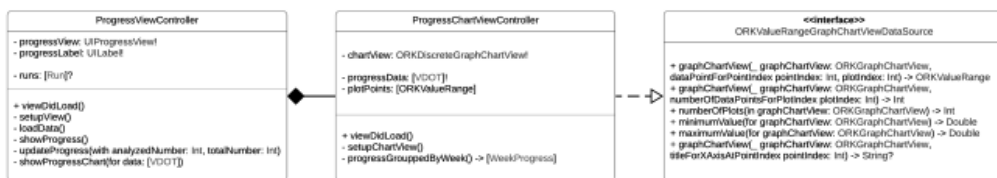


Рис. 2.4.6.10. UML діаграма класів рівня контролеру (3).

Клас *ProgressViewController* (сцена «Очікування аналізу даних») — використовується для представлення прогресу аналізу даних. Дана сцена потрібна, щоб надати користувачу видимість процесу аналізу та показати, скільки пробіжок ще залишається обробити. Після завершення аналізу користувач буде автоматично перенаправлений на сцену «Прогрес» (*ProgressChartViewController*).

Клас *ProgressChartViewController* (сцена «Прогрес») — використовується для представлення прогресу рівня підготовки бігуна. Для представлення графіку використовується бібліотека *ResearchKit*, яка містить модуль *Charts*. На сцені відображаються 2 графіки: для показника VDOT і VO2Max. Показник VDOT розраховується згідно таблиць відомого тренера з бігу Jack Daniels. Показник VO2Max експортується з застосування Apple Health. Він буде наявний лише в тому випадку, якщо користувач має Apple Watch. Якщо таких даних не буде, то графік для VO2Max не буде відображатись. Усі дані згруповані по тижням. Для кожного тижня відображається мінімальний і максимальний показник за умови наявності декількох тренувань на цьому тижні.

На рисунках Рис. 2.4.6.11 та Рис. 2.4.6.12 зображено діаграми класів рівня моделі. Кожен з яких інкапсулює дані певної сутності в системі.

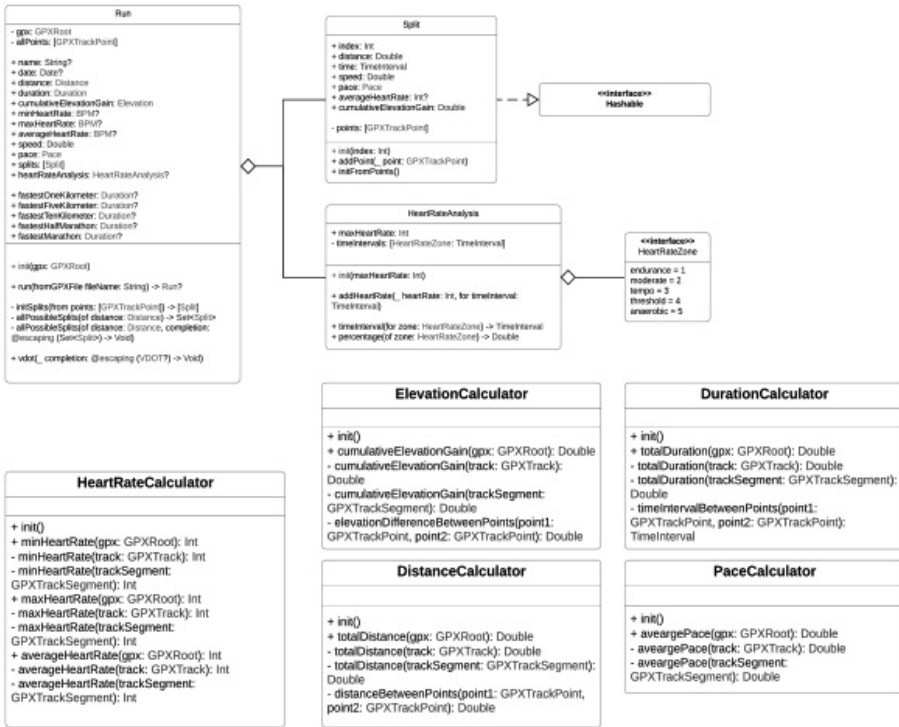


Рис. 2.4.6.11. UML діаграма класів рівня моделі (1).

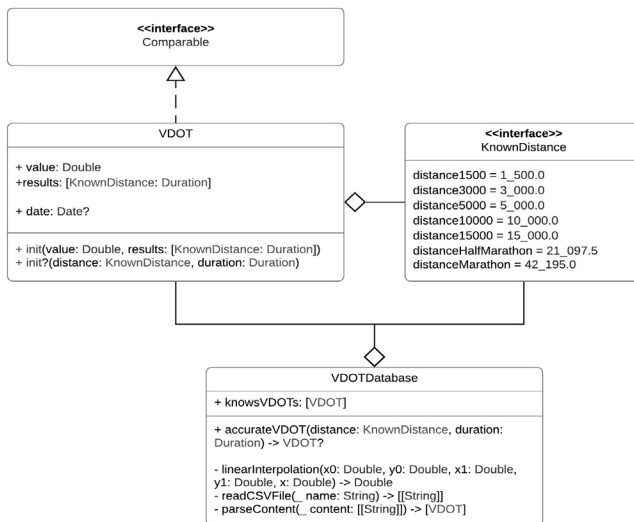


Рис. 2.4.6.12. UML діаграма класів рівня моделі (1).

2.4.7 Опис програми та її алгоритмів

Діаграма прецедентів або діаграма варіантів використання — це UML-діаграма, на якій зображено відношення між акторами та прецедентами в системі. Діаграми прецедентів відображають елементи моделі варіантів використання. Функції, наявні в розробленому застосуванні представлені на рисунку 2.4.7.1. Їх опис наведено нижче.

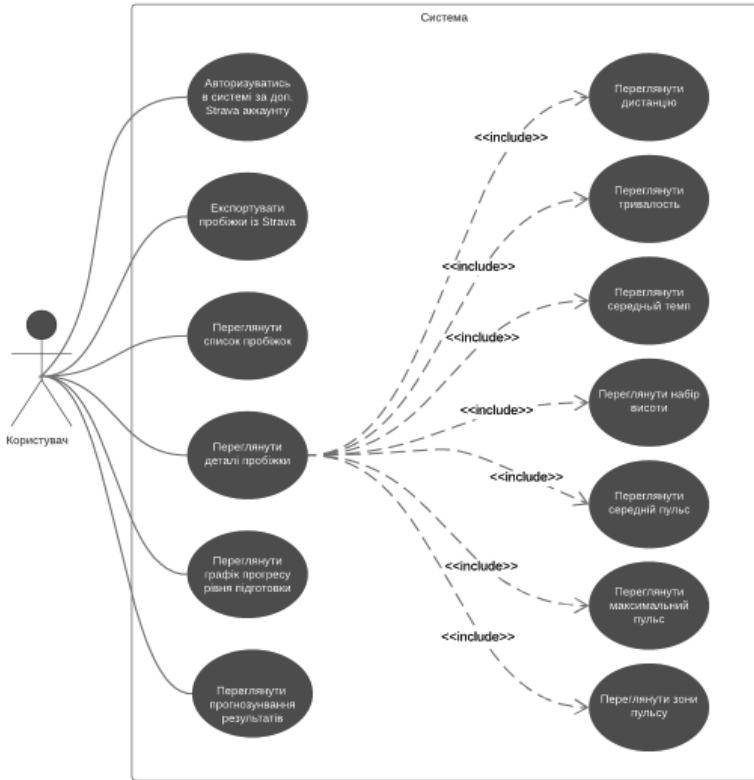


Рис. 2.4.7.1. UML діаграма варіантів використання.

Користувач має наступні можливості для використання:

- 1) Авторизуватись в системі за допомогою Strava акаунту.
- 2) Експортувати усі пробіжки із Strava.
- 3) Переглянути список пробіжок (експортованих з Strava).
- 4) Обрати пробіжку зі списку.
- 5) Переглянути деталі обраної пробіжки.
 - a. Дистанція.
 - b. Тривалість.
 - c. Швидкість.
 - d. Середній темп.
 - e. Набір висоти.

- f. Середній пульс.
- g. Максимальний пульс.
- 6) Переглянути зони пульсу для обраної пробіжки.
- 7) Переглянути графік прогресу рівня підготовки.
- 8) Ознайомитись з прогнозами результату на обраній дистанції.

2.4.8 Інструкція користувача

На даному етапі застосування ще не завантажено до AppStore, тому щоб його запустити потрібно виконати ряд інструкцій.

- 1) Клонувати репозиторій проекту, що розміщується на GitHub. Для цього в терміналі потрібно ввести наступну команду:

```
git clone https://github.com/mishamoe/ranalyze.git
```

- 2) Встановити CocoaPods. Для цього в терміналі потрібно ввести наступну команду:

```
sudo gem install cocoapods
```

- 3) Перейти в директорію проекту.

- 4) Встановити залежності проекту (Pods). Для цього в терміналі потрібно ввести наступну команду:

```
pod install
```

- 5) Завантажити Xcode з AppStore (якщо він ще не встановлений).

- 6) Відкрити проект — файл *Ranalyze.xcworkspace*.

- 7) Обрати будь-який iOS симулятор та натиснути Run (⌘R).

- 8) Зачекати доки завантажиться симулятор і відкриється застосування.

Після першого запуску застосування користувачу буде запропоновано авторизуватись в системі за допомогою Strava аккаунту (Рис. 2.4.8.1). Потрібно натиснути кнопку «Continue». Відкриється веб-сторінка авторизації в Strava через емейл та пароль. Також Strava надає можливість авторизуватись через Facebook або Google аккаунт (Рис. 2.4.8.2).

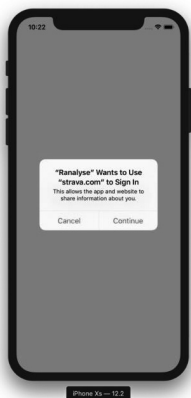


Рис. 2.4.8.1. Запит на авторизацію.

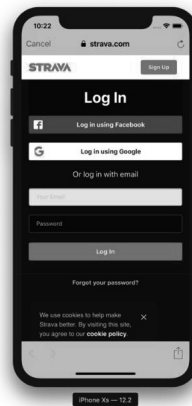


Рис. 2.4.8.2. Авторизація в Strava.

Після успішної авторизації Strava попросить користувача надати даному застосуванню доступ до перегляду публічного профайлу, а також до приватних активностей. Це не є обов'язково, проте буде добре надати цей доступ. Для цього потрібно натиснути на кнопку «Authorize» (рис. 2.4.8.3). Після чого користувач буде перенаправлений на початкову сцену застосування — «Список пробіжок» («Runs»). Внизу екрану відображається меню застосування у вигляді Tab Bar. В меню доступні 5 пунктів: 1. Список пробіжок (Runs); 2. Досягнення (Achievements); 3. Прогрес (Progress); 4. Результати (Results); 5. Налаштування (Settings).

Коли користувач вперше потрапить на сцену «Список пробіжок» — експорт пробіжок із Strava почнеться автоматично. Користувачу доведеться зачекати декілька хвилин, поки усі дані будуть експортовані. Час очікування напряму залежить від кількості пробіжок. Після завершення експорту користувач побачить список, у якому для кожної пробіжки буде відображено назву, дату і час (Рис. 2.4.8.4).

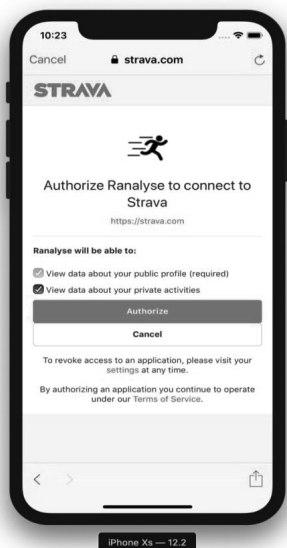


Рис. 2.4.8.3.

Надання доступу до даних в Strava.

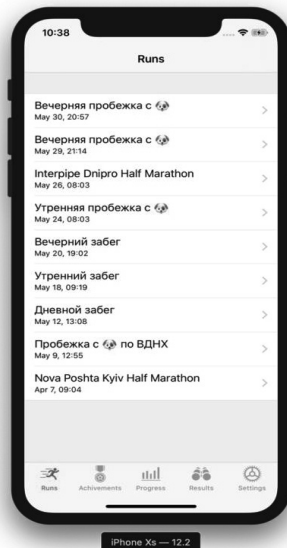


Рис. 2.4.8.4.

Сцена «Список пробіжок».

Для ознайомлення із деталями пробіжки та аналітикою потрібно натиснути на комірку пробіжки в таблиці. Про те, що на комірку можна натиснути говорить стрілочка в правому куті. Після чого відкриється сцена «Деталі пробіжки» (Рис. 2.4.8.5). На цій сцені користувач зможе переглянути основні характеристики пробіжки. Такі як:

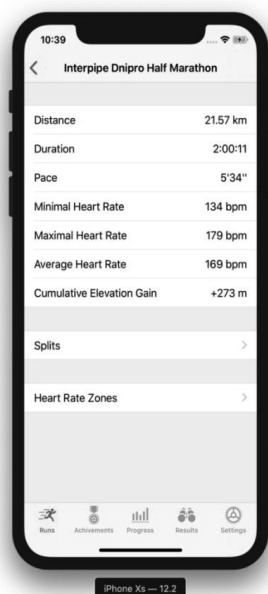
- Дистанція (Distance)
- Тривалість (Duration)
- Темп (Pace)

- Мінімальний пульс (Minimal Heart Rate)
- Максимальний пульс (Maximal Heart Rate)
- Середній пульс (Average Heart Rate)
- Набір висоти (Cumulative Elevation Gain)

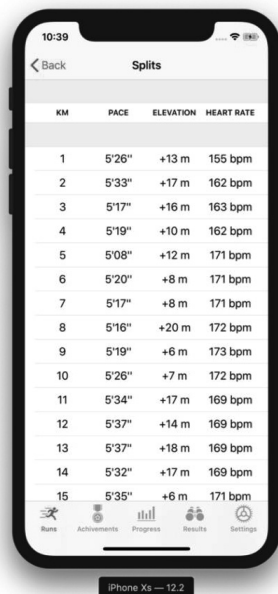
Також користувачу доступні можливості переглянути аналіз пробіжки по відрізках (Splits) і аналіз зон пульсу. Для того, щоб перейти до аналізу потрібно натиснути на відповідну комірчку у таблиці. Потрапивши на сцену «Відрізки» (Splits) користувач зможе переглянути детальну інформацію про кожний кілометр пробіжки (рис. 2.4.8.6), а саме:

- Темп (Pace)
- Набір висоти (Elevation)
- Середній пульс (Heart Rate)

Цю інформацію корисно використовувати для аналізу динаміки пробіжки.



iPhone Xs — 12.2



iPhone Xs — 12.2

Рис. 2.4.8.5. Сцена «Деталі пробіжки». Рис. 2.4.8.6. Сцена «Відрізки».

Потрапивши на сцену «Зони пульсу» (Heart Rate Zones) користувач зможе переглянути персональні рекомендації щодо діапазонів пульсу та часу який користувач провів в кожній зоні (рисунок 2.20):

- 1) Z1 — Спокійна (Endurance) — дуже низький рівень інтенсивності
- 2) Z2 — Середня (Moderate) — низький рівень інтенсивності
- 3) Z3 — Ритмічна (Tempo) — середній рівень інтенсивності
- 4) Z4 — Порогова (Threshold) — інтенсивне тренування
- 5) Z5 — Анаеробна (Anaerobic) — максимум

Для розрахунку зон пульсу потрібно використати максимальне значення пульсу. Це значення знаходиться автоматично шляхом аналізу усіх експортованих пробіжок користувача. Це спрощує процес, оскільки користувачу не потрібно вводити ці дані вручну, на відміну від інших бігових застосувань.

Наступним пунктом у меню є «Досягнення» (Achievements) — Рис. 2.4.8.7.

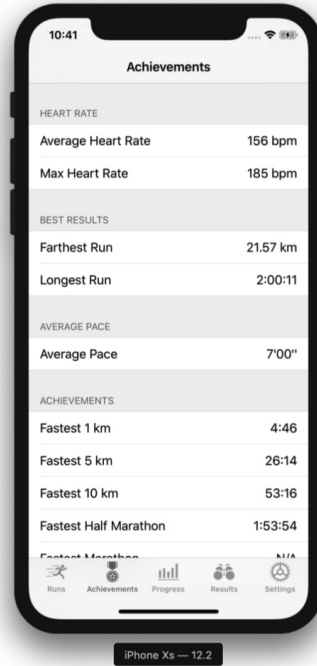


Рис. 2.4.8.7. Сцена «Зони пульсу». Рис. 2.4.8.8. Сцена «Досягнення».

На цій сцені користувач може ознайомитись із своїми найкращими результатами та деякою статистикою:

- Середній пульс за всі пробіжки (Average Heart Rate)
- Максимальний пульс за всі пробіжки (Max Heart Rate)
- Найдовша пробіжка (Farthest Run)
- Найтриваліша пробіжка (Longest Run)
- Середній темп за всі пробіжки (Average Pace)
- Найкращі результати на різних дистанціях:
 - 1 KM
 - 5 KM
 - 10 KM
 - Напівмарафон
 - Марафон

Наступним пунктом у меню є «Прогрес» (Progress) — Рис. 2.4.8.9, Рис. 2.4.8.10, Рис. 2.4.8.11. Щоб побачити графік прогресу тренувань потрібно завантажити усі необхідні дані, а також проаналізувати усі пробіжки. Цей процес є досить складним і може зайняти трохи часу. Тому спочатку користувачу буде відображено повідомлення, що потрібно зачекати, доки завантажаться дані (Рис. 2.4.8.12), а потім детальна інформація про те скільки пробіжок вже було проаналізовано і скільки залишилось (Рис. 2.4.8.13).



Рис. 2.4.8.9. Сцена «Завантаження даних». Рис. 2.4.8.10. Сцена «Аналіз даних».

Для кожного тижня буде відображено діапазон значень від мінімального до максимального VDOT. Цей показник використовується для визначення рівня прогресу бігуна. На сцені «Налаштування» (Settings) користувач може вийти із Strava аккаунту, натиснувши на кнопку «Log out» (Рис. 2.4.8.11, Рис. 2.4.8.12). В такому випадку усі експортовані пробіжки та проміжні результати аналізу будуть видалені.

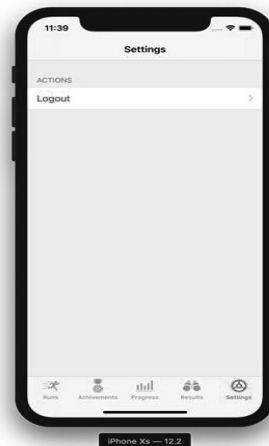
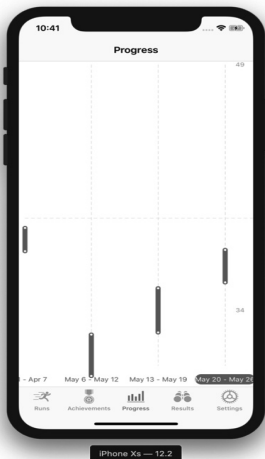


Рис. 2.4.8.11. Сцена «Прогрес».

Рис. 2.4.8.12. Сцена «Налаштування».

Висновки до розділу 2.4

Було досліджено існуючі методи аналізу ефективності тренувань та прогнозування результатів на конкретній дистанції. Запропоновано використовувати показники VDOT та VO2Max для визначення рівня підготовки спортсмену, ефективності та прогресу тренувань.

Було розроблене програмне забезпечення, що може використовуватись аматорами та професійними бігунами для оцінки ефективності процесу тренувань, для ознайомлення з прогресом рівня підготовки, для підбору відповідного темпу для участі на змаганнях.

2.5 Збільшення часу роботи датчиків небезпечних ситуацій шляхом регулювання енерговитрат

Запропоновано інформаційну технологію, що забезпечує підвищення терміну роботи датчиків шляхом регулювання величини енерговитрат, що досягається шляхом зменшення зони охоплення датчику [33]. Наведено приклад залежності радіусу охоплення датчику від величини енергії, що використовується. Побудована відповідна математична модель та описано алгоритм знаходження оптимальних зон покриття із заданим критерієм мінімізації області перетину. Наведено результати комп'ютерної симуляції з використанням запропонованого підходу та аналіз отриманих результатів, що свідчить про результативність запропонованої інформаційної технології.

Постановка проблеми. Із розвитком сучасних приладів та інформаційних технологій, зокрема безпроводних автономних датчиків, перевага надається мінімізації апаратної складової, наслідком чого є збільшення вартості продукції. В свою чергу, зменшення розміру приладу суттєво впливає на розмір елемента живлення, наслідком чого є зміна терміну використання сенсору, як правило зменшення. Також виготовлення датчиків з змінними елементами живлення або з можливістю відновлення запасу енергії є нерентабельним для виробника. Розвиток подібних систем спричиняє високе споживання енергії сучасними бездротовими пристроями. Завдання інженерів полягає в зменшенні споживаної енергії шляхом зміни компонент датчиків або у застосуванні інших підходів, наприклад переходу на спеціалізовані інформаційні системи, що дозволяють заощадити кількість споживаної енергії, що в свою чергу продовжить термін використання датчиків. Отже, створення нових та покращення вже існуючих інформаційних технологій керування датчиками дозволить підвищити термін використання приладів, і як наслідок знизити витрати.

Аналіз останніх досліджень і публікацій. У зв'язку з використанням сучасних технологій передачі даних та створенню нових протоколів та стандартів стрімкого розповсюдження набуло використання бездротових технологій [47]. Існує кілька стандартів бездротової передачі даних [47, 119], найпоширенішими з яких є технології Wi-Fi, Bluetooth і ZigBee. Пристрої Wi-Fi споживають багато енергії, щоб забезпечити необхідну швидкодію. З відстанню і при наявності перешкод швидкість передачі різко знижується. У свою чергу, Bluetooth розроблений для мобільних пристроїв і використовується для підключення бездротових пристроїв або стільникового телефону до навігатора GPS або ПК. Швидкість передачі по Bluetooth складає приблизно 1 Мбіт/с [119], але у порівнянні з Wi-Fi споживання енергії нижче. У мережі датчиків вимоги зовсім інші. Датчики часто працюють протягом декількох років від батарейки завбільшки з монету або від енергії, отриманої з навколишнього середовища. У більшості датчиків акумулятор неможливо підзарядити (наприклад, як телефон). Прикладом застосування бездротових мереж є система «розумний будинок» [89, 122], коли в приміщенні розташовано безліч датчиків

і виконавчих пристроїв, якими за відсутності користувача управляє сервер [96]. Також подібні мережі можливо застосувати до екологічного моніторингу [41, 71, 79]. Як приклад, у роботі [71] розглядається мережа датчиків звуку, що використовується визначення для рівня загрози для людини у зоні метрополітену шляхом порівняння отриманих результатів із іншими працями [41, 71, 79, 102] та із санітарними нормами [40]. Для збору необхідних вхідних даних використовуються як стаціонарні, так і датчики користувачів, що розташовані у мобільних пристроях [69, 81, 108]. Аналіз вказаних джерел дозволив виявити актуальність проблеми підвищення терміну використання датчиків різних типів та різного призначення.

Мета – забезпечення підвищення терміну використання датчиків шляхом застосування оптимізації енерговитрат, створення відповідної інформаційної технології та прикладного програмного забезпечення для вирішення поставленої проблеми, дослідження та аналіз отриманих результатів.

Нехай маємо множину датчиків $S = \{S_1, S_2, \dots, S_n\}$, де n – кількість датчиків. Очевидно, що кожен датчик володіє певними характеристиками, зокрема величиною, що вимірює, одиницями вимірювання величиною ємності акумулятора, позицією розташування, типом та методами зв'язку з іншими датчиками, тощо. Будемо брати до розгляду такі параметри: позиція розташування $P(x, y)$, радіус охоплення r (м), ємність акумулятора B_v (mAh), витрати енергії за одиницю часу роботи датчику B_c (mAh), тобто $S_i = S_i(P, r, B_v, B_c)$. Відомо, що величина радіусу охоплення датчиків регулюється в залежності від величини потужності, що впливає на величину B_{c_i} , та не може перевищувати максимальне значення для датчика r_i^{\max} . Цю залежність можна відобразити у наступному вигляді:

$$B_{c_i} = \frac{r_i}{r_i^{\max}} B_{c_i}^{\max}, \quad (2.5.1)$$

де r_i – поточний радіус охоплення датчику, r_i^{\max} – максимальний радіус охоплення датчику, $B_{c_i}^{\max}$ – витрати енергії при максимальному радіусу охоплення. Будемо вважати, що зона покриття датчику S_i є колом з центром в точці $(x_i; y_i)$ та радіусом r_i , $i = \overline{1, n}$, n – кількість датчиків. Перетин зон покриття датчиків схематично можна зобразити наступним чином (рис. 2.5.1).

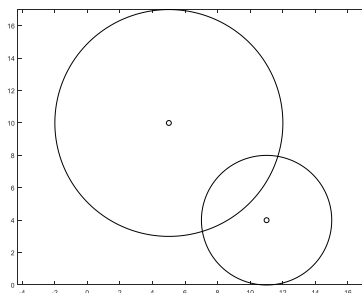


Рис. 2.5.1. Перетин зон покриття датчиків.

Очевидно, що для зменшення зон перетину покриття необхідно зменшити радіуси покриття. Слід урахувувати факт можливої відсутності перетину покриття. Наявність перетину покриття двох датчиків S_1 та S_2 можна визначити як перетин кіл з центрами у точках $(x_1; y_1)$ та $(x_2; y_2)$ з радіусами r_1 та r_2 у вигляді розв'язку наступної системи (2.5.2):

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 = r_1^2 \\ (x-x_2)^2 + (y-y_2)^2 = r_2^2 \end{cases} \quad (2.5.2)$$

Іншим способом визначення наявності перетину є спосіб з використання відстані між датчиками, що у нашому випадку є центрами кіл. Даний спосіб можна представити у наступному вигляді (2.5.1.3):

$$l_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \quad (2.5.3)$$

де $(x_1; y_1)$ та $(x_2; y_2)$ – координати датчиків.

Величина $l_{i,j}$ може набувати трьох значень в залежності від радіусів дії датчиків r_1 та r_2 . У наступній таблиці наведено опис даних випадків (Табл. 2.5.1):

Таблиця 2.5.1.

Залежність зони перетину від значення $l_{i,j}$.

Значення $l_{i,j}$	Опис
$l_{i,j} > r_1 + r_2$	Зони покриття датчиків не перетинаються.
$l_{i,j} < r_1 + r_2$	Зони покриття датчиків перетинаються та містять спільну зону.
$l_{i,j} = r_1 + r_2$	Зони покриття датчиків перетинаються у одній єдиній точці.

Введемо деяке обмеження перетину покриття датчиків $c > 0$, $c = \text{const}$. Задача полягає у зменшенні зон покриття, і, як наслідок відповідну величину енерговитрат таким чином, щоб для двох датчиків S_i та S_j виконувалася умова:

$$|r_1 - r_2| = c. \quad (2.5.4)$$

Схематично виконання умови (2.5.4) можна зобразити наступним чином (Рис. 2.5.2).

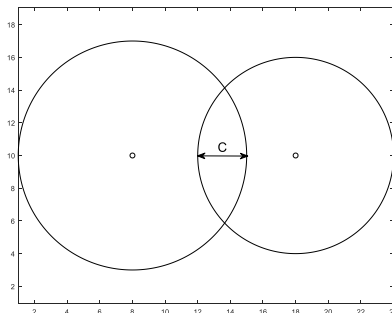


Рис. 2.5.2. Зміст значення величини c .

Також зміст величини c полягає у мінімально допустимому перетині зон покриття сенсорів у випадку передачі даних між ними.

Для подальшого розгляду проблеми додатково введемо матрицю перетинів A , елементи якої будуть рівними:

$$a_{i,j} = \begin{cases} 1, l_{i,j} < r_i + r_j \\ 0, l_{i,j} \geq r_i + r_j \end{cases} \quad (2.5.5)$$

Очевидно, що елементи $a_{i,j} |_{i=j} = 0$. Остаточна матриця перетинів A буде мати наступний вигляд:

$$A = \begin{pmatrix} 0 & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & 0 & \dots & a_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & 0 \end{pmatrix}. \quad (2.5.6)$$

Враховуючи сказане вище сформулюємо остаточно описану проблему у вигляді оптимізаційної задачі наступного вигляду:

$$F(x) = \sum_{i=1}^n r_i x_i \rightarrow \max, \quad (2.5.7)$$

$$\sum_{i=1}^n a_{i,j} (r_i x_i - r_j x_j) = a_{i,j} (c + l_{i,j}), \quad (2.5.8)$$

$$a_{i,j} = \begin{cases} 1, r_i > r_j \\ -1, r_i < r_j \end{cases}, \quad (2.5.9)$$

$$l_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \quad (2.5.10)$$

$$0 < x_i \leq 1, i = \overline{1, n}, j = \overline{i+1, n}, \quad (2.5.11)$$

де (2.5.7) – цільова функція, (2.5.8) та (2.5.11) лінійні обмеження, (2.5.9) та (2.5.10) – використовуються у обмеженнях (2.5.8), $a_{i,j}$ – елементи матриці перетинів (2.5.7), n – кількість датчиків.

Розв'язком задачі (2.5.7) – (2.5.11) є вектор $x = (x_1, x_2, \dots, x_n)$, в якому елементи x_i є коефіцієнтами на які необхідно поелементно помножити вхідний вектор радіусів $r = (r_1, r_2, \dots, r_n)$ для виконання умови (2.5.4). Вектор оптимальних радіусів для датчиків матиме наступний вигляд:

$$r^* = (r_1 x_1, r_2 x_2, \dots, r_n x_n). \quad (2.5.12)$$

Алгоритм роботи інформаційної технології та відповідного програмного забезпечення, що було створене для розв'язання описаної задачі складається з наступних етапів:

1. Аналіз координат датчиків та формування матриці перетинів A (2.5.6) шляхом перевірки умов (2.5.2) або (2.5.3) та (2.5.5). У даному алгоритмі використовується умова (2.5.3) та (2.5.5). У випадку якщо умови (2.5.5) виконуються заповнюється відповідний елемент матриці v , яка є матрицею

правих частин обмеження (2.5.8) з використанням (2.5.9) та (2.5.10), тобто $b_{i,j} = a_{i,j}(c + l_{i,j})$.

2. Враховуючи, що елементи матриці A є симетричними відносно головної діагоналі, в алгоритмі використовуються елементи, що знаходяться вище головної діагоналі у обмеженнях (2.5.8).

3. Формування цільової функції (2.5.7) та знаходження коефіцієнтів x_i , $i = \overline{1, n}$, n – кількість датчиків.

4. На основі отриманих результатів формування вектору оптимальних зон покриття шляхом виконання (2.5.12).

Результати комп'ютерного моделювання. Під час проведення обчислювальних експериментів та комп'ютерного моделюванні були використані вхідні дані, що є параметрами реальних сенсорів. Вхідні дані наведено у наступній таблиці (Табл. 2.5.2):

Таблиця 2.5.2.

Вхідні дані комп'ютерної симуляції.

Датчик	Координати (x; y)	Радіус охоплення r (м)	Ємність акумулятора V_v (mAh)	Витрата енергії V_e (mAh)
S_1	(4;5)	7	13700	12
S_2	(10;13)	8	11200	9
S_3	(13;19)	9	10000	23

Схематично розташування та зони охоплення датчиків можна зобразити у наступному вигляді (рис. 2.5.3).

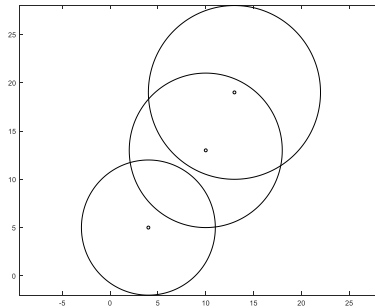


Рис. 2.5.3. Зображення початкових зон покриття сенсорів.

Елементи вектору-розв'язку, що отриманий шляхом проведення обчислювального експерименту при обмеженні $c=0.5$ мають наступні значення: $x = (1, 0.4375, 0.4120)$. Відповідно до цього, оптимальні радіуси зон покриття набувають наступних значень: $r^* = (7, 3.5, 3.7082)$. Схематично отриманий результат можна зобразити наступним чином (Рис. 2.5.4).

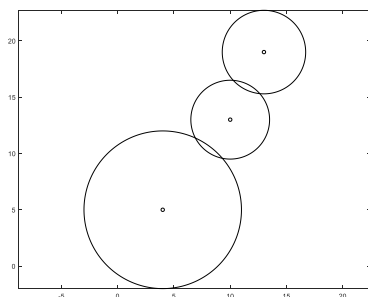


Рис. 2.5.4. Зображення оптимальних зон покриття.

Використавши формулу (2.5.1) отримаємо нові енерговитрати датчиків та нові значення терміну використання. Отримані значення можна відобразити у порівнянні із вхідними значеннями у вигляді наступної гістограми (Рис. 2.5.5):

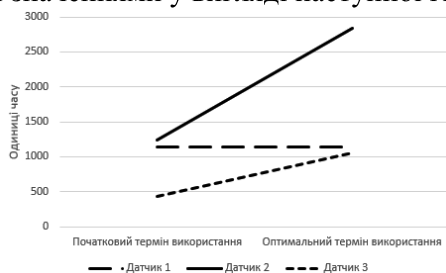


Рис. 2.5.5. Порівняння термінів використання.

Висновки до розділу 2.5

Запропоновано підхід, що дозволяє збільшити час роботи датчиків шляхом зменшення потужності споживання. В свою чергу, потужність датчику залежить від енергії, що він використовує. Описано вплив потужності датчику на площу зони покриття датчику. Зменшення використання енергії досягається шляхом зменшення зони покриття при перетині у мінімальній площі з іншим датчиком. Наведена відповідна залежність між площею покриття та величиною використання енергії за одиницю часу. Для розв'язання задачі було запропоновано математичну модель у вигляді оптимізаційної задачі лінійного програмування. Враховуючи специфіку запропонованої моделі описано покроковий алгоритм розв'язання поставленої задачі. Використовуючи запропонований алгоритм було проведено ряд комп'ютерних експериментів, що підтверджують ефективність запропонованої інформаційної технології. Подальший напрямок дослідження - покращення запропонованого алгоритму та удосконалення математичної моделі для отримання більш чітких результатів. Результати дослідження можуть бути використані як у екологічній сфері так і у технічній сфері.

3. ЗАХИЩЕНА СИСТЕМА АВТОМАТИЧНОГО КЕРУВАННЯ БЕЗПЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ ЯК ТЕХНОЛОГІЯ ІОЕ

3.1 Математична модель БПЛА

При побудові захищеної системи управління динамічним об'єктом, таким, як БПЛА, потрібно враховувати динаміку його руху. Тому аналіз і синтез захищеної системи автоматичного керування БПЛА розпочнемо зі створення моделі руху БПЛА в продольному каналі.

Прийоми, що будуть використані при створенні моделі динаміки польоту літального апарату (ЛА) є універсальними і можуть використовуватись при створенні моделей інших складних систем.

Зобразимо графічно взаємодію всіх факторів, що впливають на динаміку ЛА. В розглянутому випадку у якості таких факторів виступають сили, системи координат та кути між системами координат та силами (Рис.3.1.1).

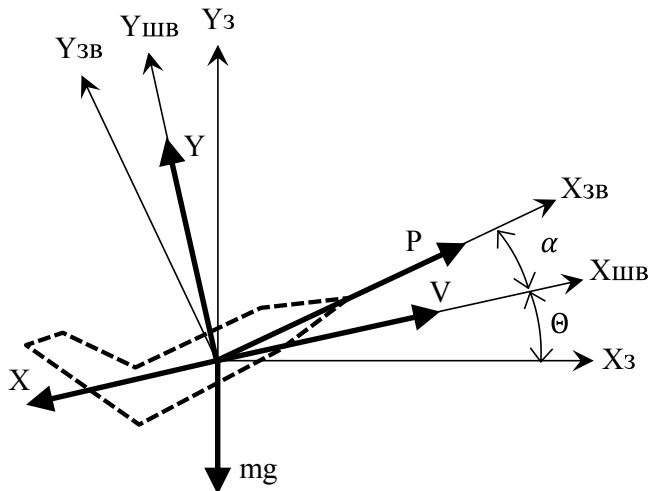


Рис. 3.1.1. Системи координат та сили, які використані для опису продольного руху літального апарату.

Визначимо **системи координат (СК)**, які будуть використані для опису динаміки продольного руху ЛА та **кути** між ними.

X_3, Y_3 – земна СК (вісь абсцис паралельна землі і співпадає з проекцією продольної вісі ЛА на землю, вісь ординат перпендикулярна поверхні землі).

$X_{шв}, Y_{шв}$ – швидкісна СК (вісь абсцис співпадає з вектором швидкості V , вісь ординат лежить у вертикальній площині).

Хзв, Узв – зв'язана СК (вісь абсцис співпадає продольною віссю конструкції ЛА, вісь ординат лежить у площині симетрії ЛА).

α – кут атаки – кут між вектором швидкості V і продольною віссю ЛА.

θ – кут нахилу траєкторії – кут між вектором швидкості V і землею поверхнею.

Визначимо **сили**, що діють на ЛА

P – сила тяги двигуна ЛА, яка співпадає з продольною віссю ЛА.

mg – сила тяжіння, яка перпендикулярна земній поверхні.

Y – підйомна сила, яка співпадає в віссю ординат швидкісної системи координат.

X – сила опору повітря, яка протилежна вектору швидкості.

Запишемо рівняння, що встановлюють баланс сил вздовж горизонтальної (3.1.1) та вертикальної (3.1.2) віссей швидкісної СК.

$$m\dot{V} = P \cos \alpha - X - m g \sin \theta \quad (3.1.1)$$

$$mV\dot{\theta} = P \sin \alpha + Y - m g \cos \theta \quad (3.1.2)$$

Запишемо рівняння для динаміки зміни висоти (3.1.1.3) та дальності (3.1.4) у земній СК.

$$\dot{h} = V \sin \theta \quad (3.1.3)$$

$$\dot{L} = V \cos \theta \quad (3.1.4)$$

Якщо ЛА літає довго, то то паливо витрачається і маса ЛА змінюється, що може суттєво впливати на динаміку. Отже додамо рівняння динаміки зміни маси ЛА через щосекундні витрати палива β

$$\dot{m} = -\beta \quad (3.1.5)$$

Рівняння (3.1.3-3.1.5) мають вигляд класичних звичайних диференціальних рівнянь (ЗДР). Логічно було б і рівняння (3.1.1, 3.1.2) привести до такого виду. Для цього поділимо рівняння (3.1.1) на m , а рівняння (3.1.2) на mV та винесемо множники за дужки так, щоб всі сили були представлені в явному вигляді.

$$\dot{V} = g \left(\frac{P \cos \alpha - X}{mg} - \sin \theta \right) \quad (3.1.1')$$

$$\dot{\theta} = \frac{g}{V} \left(\frac{P \sin \alpha + Y}{mg} - \cos \theta \right) \quad (3.1.2')$$

Представлення всіх сил у явному вигляді дозволяє знайти подальші моливості для вдалого агрегування змінних. Так, загальновідомими є вирази для перенавантажень продольного (3.1.6) і вертикального (3.1.7)

$$n_x = \frac{P \cos \alpha - X}{mg} \quad (3.1.6)$$

$$n_y = \frac{P \sin \alpha + Y}{mg} \quad (3.1.7)$$

Це дозволяє більш компактно записати рівняння (3.1.1, 3.1.1.2)

$$\dot{V} = g(n_x - \sin \theta) \quad (3.1.1'')$$

$$\dot{\theta} = \frac{g}{V}(n_y - \cos \theta) \quad (3.1.2'')$$

Тепер у нас є система із 5 досить наочних і зрозумілих звичайних диференціальних рівнянь, які описують динаміку 5 змінних. Потрібно переконатись, що для всіх параметрів і змінних значення вже визначені або будуть знайдені в ході ітераційних процедур інтегрування ЗДР.

Розкриємо вирази для аеродинамічних сил

$$\begin{aligned} Y &= C_y q S \\ X &= C_x q S \\ q &= \rho \frac{V^2}{2} \end{aligned}$$

Тут

S – площа крила.

C_y – коефіцієнт підйомної сили.

C_x – коефіцієнт опору повітря.

q - швидкісний напір.

ρ – щільність повітря.

Аеродинамічні коефіцієнти залежать від кута атаки

$$\begin{aligned} C_y &= C_y^\alpha \alpha \\ C_x &= C_{x0} + A C_y^2 \end{aligned}$$

Тут C_y^α , C_{x0} , A – постійні коефіцієнти.

Щільність повітря є постійною у землі. Але при зростанні висоти вона зменшується приблизно за експоненційним законом.

$$\rho = \rho_0 e^{-\frac{h}{\lambda}}$$

Є більш точні таблиці щільності повітря за висотами. Їх називають «стандартна атмосфера». Чи використовувати стандартну атмосферу, чи експоненційну залежність або постійну величину – залежить від постановки задачі. І точність визначення щільності повітря можна нарощувати поетапно на послідовних ітераційних кроках вдосконалення моделі.

На першому етапі можна використати суттєві спрощення постановки задачі, щоб переконатись, що модель працює в принципі. Такими спрощеннями (припущеннями) можуть бути.

- Постійна сила тяги.
- Постійна щільність повітря.
- Постійні аеродинамічні коефіцієнти.
- Постійний кут атаки.
- Постійна маса ЛА (неврахування 5-го ЗДР).
- Суттєве зменшення часу польоту (в декілька десятків, сотен або тисяч разів).
- Використання найбільш простих методів інтегрування рівнянь – метод Ейлера, який досить швидкий та досить наочний у всіх своїх проміжних результатах.
- Великий крок інтегрування.

Після перших спроб запуску такої дуже грубої моделі робляться попередні висновки щодо доцільного порядку нарощування якості моделі. На основі досвіду розробки подібних моделей можна запропонувати таку послідовність нарощування якості моделі.

- Зменшити крок інтегрування до появи хоч трохи правдоподібних результатів.
- Ввести залежність аеродинамічних коефіцієнтів від кута атаки.
- Ввести рівняння для динаміки зміни маси ЛА (5-те ЗДР).
- Врахувати зміну щільності повітря залежно від висоти (спочатку експоненційною залежністю, потім через стандартну атмосферу).
- Ввести залежність сили тяги двигуна від висоти і швидкості.
- Поступово перейти до реальної тривалості польоту.
- Після формування чіткої уяви про основні закономірності розв'язання системи ЗДР динаміки польоту можна переходити до більш точних методів інтегрування, наприклад, Рунге-Кутта, Адамса тощо (платаю за точність тут буде втрата наочності проміжних результатів).
- Ввести адаптивне управління ЛА або різні програми управління:
 - Кутом атаки.
 - Тягою двигуна в межах дозволених тяг.

В ітераційному процесі вдосконалення моделі можуть бути і інші складові. Правильне рішення приймається у інтерактивному режимі, як відповідь на певну поведінку моделі в умовах прийнятої постановки задачі.

3.2 Послідовність нарощування рівнів захищеності за етапами розвитку системи управління БПЛА

В роботі розроблена бажана послідовність нарощування рівнів захищеності інформаційної системи медичного призначення, як послідовність зміни варіантів стандартних функціональних профілів захищеності. Переваги розробленої послідовності полягають в тому, що вони мінімізують ризики переходів між різними рівнями захисту інформаційної системи з одного боку, та у чіткої відповідності діючим нормативним документам з іншого боку.

Варіант 1 (мінімально можливий). Забезпечує функціональність без врахування втрат часу. Відповідно до проведеного аналізу було визначено, що у штатному режимі ми маємо справу із автоматизованою системою класу 1. Найбільш важливим видом загроз, якому має відповідати профіль захищеності такої системи є цілісність. Тому що, часові обмеження щодо використання системи дозволяють витратити певний додатковий час на усунення проблем із функціональністю системи. Зрозуміло, що будь-які затримки в роботі погіршують якість роботи системи, але не унеможливають її. Краще втратити певний час на відновлення, а потім все ж таки виконати основні задачі щодо медичної діагностики та терапії.

Отже у якості стандартного функціонального профілю захищеності можна обрати 1.Ц.1 або 1.Ц.2 [17, 26].

1.Ц.1 = { НР-1, НИ-1, НК-1, НО-1, НЦ-1, НТ-1 }

1.Ц.2 = { ЦА-2, ЦО-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1 }

Варіант 2 (основний робочий). Забезпечує функціональність за всіма вимогами. На другому місці стоять загрози щодо доступності. Якщо система захисту буде забезпечувати захист від загроз як цілісності, так й доступності, то це буде забезпечувати практично повну функціональність системи. Отже для переходу на більш високий рівень захищеності бажано використовувати стандартний функціональний профіль захищеності 1.ЦД.1 або 1.ЦД.2-4 [17, 26].

1.ЦД.1 = { ЦА-1, ЦО-1,
ДР-1, ДВ-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1 }

1.ЦД.2 = { ЦА-1, ЦО-1,
ДР-2, ДС-1, ДЗ-1, ДВ-2, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-2 }

1.ЦД.3 = { ЦА-1, ЦО-1,
ДР-2, ДС-2, ДЗ-2, ДВ-2, НР-3, НИ-2, НК-1, НО-1, НЦ-2, НТ-2 }

1.ЦД.4 = { ЦА-1, ЦО-1,
ДР-2, ДС-3, ДЗ-3, ДВ-3, НР-4, НИ-2, НК-1, НО-1, НЦ-2, НТ-2 }

Варіант 3 (основний конфіденційний). Забезпечує повну функціональність та захист персональних даних пацієнтів. Взагалі то захист персональних даних в АС класу 1 забезпечується відносно легко, оскільки для цього можна обмежити доступ до системи на фізичному рівні. В АС класу 1 таке обмеження ніяк не вплине на функціональність. Але все ж таки слід

враховувати, що при роботі з системою поряд з лікарем знаходиться пацієнт, який потенційно може бути порушником інформаційної безпеки. Отже в цьому випадку доцільно використовувати один з таких стандартних функціональних профілів захищеності 1.КЦД.1-4 [17, 26].

- $$1.КЦД.1 = \{ \text{КА-1, КО-1, ЦА-1, ЦО-1, ДР-1, ДВ-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1} \}$$
- $$1.КЦД.2 = \{ \text{КА-1, КО-1, ЦА-1, ЦО-1, ДР-2, ДС-1, ДЗ-1, ДВ-2, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-2} \}$$
- $$1.КЦД.3 = \{ \text{КА-1, КО-1, ЦА-1, ЦО-1, ДР-2, ДС-2, ДЗ-2, ДВ-2, НР-3, НИ-2, НК-1, НО-1, НЦ-2, НТ-2} \}$$
- $$1.КЦД.4 = \{ \text{КА-1, КО-1, ЦА-1, ЦО-1, ДР-2, ДС-3, ДЗ-3, ДВ-3, НР-4, НИ-2, НК-1, НО-1, НЦ-2, НТ-2} \}$$

Варіант 4 (локально мережевий). Забезпечує повну функціональність та захист персональних даних пацієнтів при наявності підключення до локальної мережі. Оскільки сучасні медичні інформаційні системи об'єднують свої дані, то в цьому випадку доцільно використовувати один з таких стандартних функціональних профілів захищеності 2.КЦД.1-5 [17, 26].

- $$2.КЦД.1 = \{ \text{КД-2, КО-1, ЦД-1, ЦО-1, ДР-1, ДВ-1, НР-2, НИ-2, НК-1, НО-2, НЦ-2, НТ-2} \}$$
- $$2.КЦД.2 = \{ \text{КД-2, КА-2, КО-1, ЦД-1, ЦА-2, ЦО-1, ДР-1, ДВ-1, НР-2, НИ-2, НК-1, НО-2, НЦ-2, НТ-2} \}$$
- $$2.КЦД.3 = \{ \text{КД-2, КА-2, КО-1, КК-1, ЦД-1, ЦА-3, ЦО-2, ДР-2, ДС-1, ДЗ-1, ДВ-2, НР-3, НИ-2, НК-1, НО-2, НЦ-3, НТ-2} \}$$
- $$2.КЦД.4 = \{ \text{КД-3, КА-3, КО-1, КК-1, ЦД-1, ЦА-3, ЦО-2, ДР-3, ДС-2, ДЗ-2, ДВ-2, НР-4, НИ-2, НК-1, НО-3, НЦ-3, НТ-2} \}$$
- $$2.КЦД.5 = \{ \text{КД-4, КА-4, КО-1, КК-2, ЦД-4, ЦА-4, ЦО-2, ДР-3, ДС-3, ДЗ-3, ДВ-3, НР-5, НИ-2, НК-2, НО-3, НЦ-3, НТ-2} \}$$

Варіант 5 (глобально мережевий). Забезпечує повну функціональність та захист персональних даних пацієнтів при наявності підключення до глобальної мережі.

Часто сучасні медичні інформаційні системи мають розподілену структуру. В цьому випадку доцільно використовувати один з таких стандартних функціональних профілів захищеності 3.КЦД.1-5 [17, 26].

- 3.КЦД.1 = { КД-2, КО-1, КВ-1,
ЦД-1, ЦО-1, ЦВ-1,
ДР-1, ДВ-1,
НР-2, НИ-2, НК-1, НО-2, НЦ-2, НТ-2, НВ-1 }
- 3.КЦД.2 = { КД-2, КА-2, КО-1, КВ-2,
ЦД-1, ЦА-2, ЦО-1, ЦВ-2,
ДР-1, ДВ-1,
НР-2, НИ-2, НК-1, НО-2, НЦ-2, НТ-2, НВ-1 }
- 3.КЦД.3 = { КД-2, КА-2, КО-1, КК-1, КВ-3,
ЦД-1, ЦА-3, ЦО-2, ЦВ-2,
ДР-2, ДС-1, ДЗ-1, ДВ-2,
НР-3, НИ-2, НК-1, НО-2, НЦ-3, НТ-2, НВ-2 }
- 3.КЦД.4 = { КД-3, КА-3, КО-1, КК-1, КВ-3,
ЦД-1, ЦА-3, ЦО-2, ЦВ-2,
ДР-3, ДС-2, ДЗ-2, ДВ-2,
НР-4, НИ-2, НК-1, НО-3, НЦ-3, НТ-2, НВ-2, НА-1, НП-1 }
- 3.КЦД.5 = { КД-4, КА-4, КО-1, КК-2, КВ-4,
ЦД-4, ЦА-4, ЦО-2, ЦВ-3,
ДР-3, ДС-3, ДЗ-3, ДВ-3,
НР-5, НИ-2, НК-2, НО-3, НЦ-3, НТ-2, НВ-2, НА-1, НП-1 }

Для порівняльного аналізу обраних стандартних функціональних профілів захищеності узагальнимо їх характеристики в таблиці (Табл. 3.2.1).

Найбільший рівень функціональності забезпечує АС класу 3 (**варіант 5**), оскільки дозволяє залучати до роботи зовнішні інформаційні ресурси. Але вона є найбільш уразливою для інформаційних атак.

Використання АС класу 2 (**варіант 4**) мало що додає до функціональності лікувально-діагностичного комплексу у порівнянні з варіантами 1-3, але може гарно забезпечити роботу АСУ лікувального заходу відповідною інформацією щодо пацієнтів. При цьому економія на засобах захисту є незначною.

АС класу 1 (**варіанти 1-3**) дозволяє забезпечити повну функціональність лікувально-діагностичного комплексу, але в автономному режимі без взаємодії із зовнішніми інформаційними ресурсами. На відміну від варіанту 1, варіанти 2 та 3 забезпечують високий ступінь захисту при збереженні високого рівня функціональності системи. Отже можна казати, що за ознакою функціональної стійкості найбільшу функціональну стійкість для автономної системи забезпечує варіант 3. Але з урахуванням економічних показників оптимальним є варіант 2.

Таблиця 3.2.1.

Порівняльний аналіз обраних стандартних функціональних профілів захищеності

	В-т 1		Варіант 2				Варіант 3				Варіант 4					Варіант 5				
	1.Ц.		1.ЦД.				1.КЦД.				2.КЦД.					3.КЦД.				
	1	2	1	2	3	4	1	2	3	4	1	2	3	4	5	1	2	3	4	5
Послуги конфіденційності:																				
КД — довірча конфіденційність;											2	2	2	3	4	2	2	2	3	4
КА — адміністративна конфіденційність;							1	1	1	1		2	2	3	4		2	2	3	4
КО — повторне використання об'єктів;							1	1	1	1	1	1	1	1	1	1	1	1	1	1
КК — аналіз прихованих каналів;												1	1	2			1	1	2	
КВ — конфіденційність при обміні.																2	3	3	4	
Послуги цілісності:																				
ЦД — довірча цілісність;											1	1	1	1	4	1	1	1	1	4
ЦА — адміністративна цілісність;		2	1	1	1	1	1	1	1	1		2	3	3	4		2	3	3	4
ЦО — відкат;		1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	1	2	2	2
ЦВ — цілісність при обміні.																				
Послуги доступності:																				
ДР — використання ресурсів;			1	2	2	2	1	2	2	2	1	1	2	3	3	1	1	2	3	3
ДС — стійкість до відмов;				1	2	3		1	2	3			1	2	3			1	2	3
ДЗ — гаряча заміна;				1	2	3		1	2	3			1	2	3			1	2	3
ДВ — відновлення після збоїв.			1	2	2	3	1	2	2	3	1	1	2	2	3	1	2	2	2	3
Послуги спостережності:																				
НР — реєстрація;	1	2	2	2	3	4	2	2	3	4	2	2	3	4	5	2	2	3	4	5
НИ — ідентифікація і автентифікація;	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
НК — достовірний канал;	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	2
НО — розподіл обов'язків;	1	1	1	1	1	1	1	1	1	1	2	2	2	3	3	2	2	2	3	3
НЦ — цілісність КЗЗ;	1	1	1	1	2	2	1	1	2	2	2	2	3	3	3	2	2	3	3	3
НТ — самотестування;	1	1	1	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2
НВ — автентифікація при обміні;																1	1	2	2	2
НА — автентифікація відправника;																			1	1
НП — автентифікація одержувача.																			1	1

3.3 Створення КСЗІ в автоматизованих системах управління БПЛА

Вимоги щодо захисту інформації мають бути враховані вже на етапі проектування та створення програмного забезпечення, яке входить до складу автоматизованої системи (АС). Але на цьому робота щодо захисту не закінчується. Наступною складовою має бути створення комплексної системи захисту інформації (КСЗІ). Якщо технічні вимоги щодо захисту інформації не були враховані на етапі проектування та створення АС, то створити КСЗІ скоріше всього буде неможливо або вкрай важко. Якщо технічні вимоги були враховані, то далі починається етап створення КСЗІ із залученням інших нетехнічних галузей діяльності.

Комплексна система захисту інформації розглядається як єдина сукупність правових, організаційних, фізичних заходів, програмно-технічних засобів, спрямованих на нейтралізацію загроз інформаційній безпеці. Постановка задачі розробки КСЗІ формується як «різними засобами і заходами запобігти впливу загроз і відповідного витоку, руйнування та викривлення інформації до граничного рівня при мінімальних витратах». При цьому необхідно враховувати, що незахищеність будь-якої ланки в структурі АС може призвести до суттєвого знецінювання витрат, які були покладені на здійснення відповідних заходів захисту інформації в усіх інших ланках.

Оскільки КСЗІ є складовою частиною АС, терміни початку, завершення робіт, етапи створення КСЗІ повинні співпадати з розробкою самої АС, замовник АС повинен виступати і замовником КСЗІ. Розробник АС при необхідності та за погодженням з замовником для виконання робіт зі створення КСЗІ може залучати інші організації (співвиконавців), які мають необхідні ліцензії (дозволи) на проведення таких робіт. Такий підхід до створення КСЗІ забезпечує захист інформації на всіх етапах життєвого циклу АС.

Етапи створення КСЗІ включають:

1. Обґрунтування необхідності створення КСЗІ. Підставою для визначення необхідності створення КСЗІ є вимоги чинного законодавства, що встановлюють обов'язковість обмеженого доступу до інформації, що обробляється в АС. Необхідність створення КСЗІ одержується за результатами встановлення належності інформації.

2. Обстеження середовищ функціонування об'єктів інформаційної діяльності (ОІД).

Метою обстеження ОІД є вивчення інформаційної діяльності, визначення об'єктів захисту, аналіз та побудова моделі загроз. Проведення робіт з обстеження ОІД у загальному випадку включає:

аналіз умов функціонування ОІД (складання ситуаційного плану, план-схеми ОІД та їх опис);

характеристику приміщень (генеральні плани поверхів будівлі, на яких розташовано ОІД);

визначення склад, схеми розташування основних технічних засобів на ОІД та їх опис;

визначення складу, схеми розташування допоміжних технічних засобів (електроживлення побутової мережі та освітлення, автоматизації, заземлення, пожежної та охоронної сигналізації, телефонної мережі, опалення, водопостачання та каналізації, вентиляції і т. ін.) та їх опис. Визначення допоміжних технічних засобів, що потребують переобладнання або підлягають демонуванню;

виявлення незадіяних та транзитних електропровідних кабелів, дротів, ланцюгів та інженерних комунікацій, що проходять через виділені приміщення;

визначення наявності систем захисту інформації ОІД. Надання пропозицій щодо застосування додаткових заходів захисту;

перевірка наявності нормативних документів, які забезпечують функціонування системи захисту інформації, організацію проектування будівельних робіт з урахуванням вимог ТЗІ, а також нормативної та експлуатаційної документації, яка забезпечує інформаційну діяльність.

Для проведення обстеження ОІД наказом керівника установи/закладу призначається комісія. У наказі відбивається мета створення комісії, її склад, об'єкти, що підлягають обстеженню, строки подання результатів.

За рішенням керівника установи/закладу для участі в роботах, надання методичної допомоги, оцінювання повноти та якості реалізації заходів захисту можуть залучатися спеціалісти з ТЗІ інших організацій, які мають ліцензію на проведення відповідних видів робіт. Обстеження ОІД також може проводитися підприємствами, установами та організаціями, які отримали встановленим порядком відповідні ліцензії на проведення відповідних видів робіт.

Докладно порядок проведення обстеження викладено в [28]. Результати проведення робіт з обстеження оформлюються актом, який підписується членами комісії та затверджується керівником установи/закладу. До акту додаються ситуаційний план розташування будівлі ОІД, план-схема контрольованої території ОІД, генеральні плани поверхів будівлі ОІД, схеми транзитних та незадіяних електропровідних кабелів та інженерних комунікацій, протоколи вимірювань опору заземлення, вимірювань побічних електромагнітних випромінювань і наводок.

За результатами обстеження розробляється модель загроз та порушників відповідно до положень [24, 25, 28].

3. Розробка політики безпеки інформації в АС.

Політика безпеки являє собою набір вимог, правил, обмежень, рекомендацій, які регламентують порядок обробки інформації і спрямовані на захист її від потенційних загроз.

Розроблення політики безпеки передбачає проведення наступних робіт:

- визначення інформаційних та технічних ресурсів в АС, що підлягають захисту,
- уточнення моделі загроз і порушників,

- аналіз ризиків,
- вибір основних рішень з протидії можливим загрозам,
- формування загальних вимог, правил, обмежень, рекомендацій, які регламентують використання окремих засобів і заходів захисту інформації,
- правила діяльності користувачів усіх категорій.

Вимоги до формування політики безпеки та її змісту визначені відповідно з положеннями [24, 25].

Політика безпеки є основним документом, який визначає вимоги до пунктів технічного завдання на створення КСЗІ, оформляється у вигляді окремого документа Плану захисту.

4. Розробка технічного завдання (ТЗ) на створення КСЗІ.

Технічне завдання на створення КСЗІ оформляється у вигляді окремого (часткового) ТЗ, доповнення до загального ТЗ на АС або розділу до загального ТЗ на АС. В ТЗ на створення КСЗІ викладаються вимоги до функціонального складу і порядку розробки і впровадження засобів і заходів захисту інформації.

ТЗ повинно містити наступні розділи:

1) Загальні відомості. В цьому розділі вказуються структурна схема і склад АС, характеристика інформації (ступінь її обмеженості), характеристика фізичного середовища (територіальне розміщення компонентів АС), наявність категоризованих приміщень, характеристики каналів зв'язку (пропускна спроможність, типи кабельних ліній, види зв'язку з віддаленими сегментами АС і користувачами, характеристика персоналу (кількість і категорії за формою допуску користувачів), особливості функціонування АС (режими роботи), реалізовані організаційні та фізичні заходи захисту, потенційні загрози інформації (можливі способи НСД, технічні канали витоку інформації);

2) Мета і призначення КСЗІ. Визначається функціональне призначення КСЗІ, нормативно-правові документи, що регламентують організацію захисту інформації в АС.

3) вимоги до КСЗІ.

У частині захисту від несанкціонованого допуску (НСД). Захист від НСД повинен гарантувати виконання вимог конфіденційності, цілісності, доступності і спостереженості інформації з обмеженим доступом і включати:

- правила керування доступом користувачів до інформаційних ресурсів. В основу доступу повинен бути покладений адміністративний принцип розмежування, який повинен реалізовуватися за мінімумом повноважень. Згідно цього принципу користувачеві доступ до інформаційних ресурсів може бути подано лише за фактом службової необхідності;

- правила розмежування інформаційних потоків за їх ступенем секретності та інших обмежень у розповсюдженні;

- правила маркірування носіїв інформації, правила контролю та реєстрації за діями користувачів щодо доступу до інформаційних ресурсів, проведення контролю за процедурами модифікації, забезпечення цілісності інформаційних ресурсів.

Вимоги до захисту інформації реалізовується послугами, які включають технічні, апаратно-програмні та організаційні заходи. Перелік необхідних послуг, які повинна реалізувати КСЗІ, складає відповідний **функціональний профіль захищеності**. Зміст функціональних профілів захищеності визначається вимогами захисту певної інформації від певних загроз і відомих функціональних послуг, що дають змогу протистояти можливим загрозам. Порядок вибору функціональних профілів захищеності від ступеня обмеженості інформації викладено у [26].

Як засоби і заходи інформації від НСД повинні розглядатися:

а) програмно-технічного характеру:

- використання спеціальних програм для фіксації спроб несанкціонованого доступу до системи з метою здобуття інформації, руйнування або викривлення програмного забезпечення;
- періодичне оновлення захисних алгоритмів;
- використання засобів автентифікації користувачів (на підставі застосування різного виду паролів, фізико-антропометричних властивостей) та засобів контролю переданої, зберігаємої інформації;
- використання спеціальних обчислювачів для вирішення конкретних завдань, які працюють за програмами з постійно запам'ятовуючих пристроїв, до яких не може бути доступу і які не можуть бути модифікованими;
- реєстрація дій користувачів та процесів у відношенні до об'єктів, що підлягають захисту;
- використання апаратних засобів ідентифікації і автентифікації;
- впровадження заходів контролю цілісності інформаційних ресурсів (файлів, програмного забезпечення та ін.)
- удосконалення протоколів управління та взаємодії елементів системи;
- розробка та впровадження спеціальних програм, які дають змогу вибирати найкращі за надійністю маршрути передавання інформації технічного стану системи;
- використання захищених каналів зв'язку, застосування оптоволоконної технології в кабельних мережах;

б) організаційного характеру:

- розробка розпорядних документів, які регламентують порядок розмежування доступу до інформаційних ресурсів;

- періодичний контроль безпеки інформації та ресурсів системи, їх ідентифікація і автентифікація;
- контроль терміну зберігання інформації;
- своєчасна сертифікація нових елементів системи;
- упорядкування документів та копіювання інформації;
- використання ліцензійного програмного забезпечення;
- заборона використання носіїв інформації без антивірусної перевірки;
- організація системи зберігання та реєстрації ключової інформації;
- організація системи відновлення після збоїв;
- періодичний контроль ефективності заходів захисту інформації.

В ТЗ на створення КСЗІ як **основні засоби і заходи** інформації повинні розглядатися:

- системо і схемотехнічні методи – обмеження використання інтерфейсів з передачею сигналів у вигляді послідовного коду і в режимі багаторазових повторень, використання засобів обчислювальної техніки і технічних засобів, до складу яких входять стійкі до самозбудження схеми з низькими рівнями електромагнітних випромінювань, використання раціональних способів, монтажу з мінімальною довжиною електричних зв'язків і комутацій тощо;
- засоби просторового і лінійного зашумлення;
- засоби локального або загального екранування;
- розміщення засобів обчислювальної техніки і технічних засобів, що мінімізує зону, в межах якої граничне відношення сигнал/шум не перевищує встановлених норм.

4) вимоги до складу проектної та експлуатаційної документації.

Перелік необхідної проектної та експлуатаційної документації визначається розробником КСЗІ і погоджується з замовником.

В цілому при розробці ТЗ на створення КСЗІ необхідно керуватись нормативними документами [27].

5. На підставі і у відповідності з ТЗ на створення КСЗІ розробником КСЗІ обґрунтовуються і приймаються проектні рішення щодо реалізації вимог ТЗ.

Проектна документація складається з «Ескізного», «Технічного» та «Робочого проектів КСЗІ».

В зазначених проектах або в «Техноробочому проекті КСЗІ» розробляються проектні рішення, необхідні для реалізації вимог ТЗ на КСЗІ, структури КСЗІ (організаційної, технічної і програмних засобів), рішень щодо архітектури комплексу засобів захисту, механізмів реалізації визначених функціональним профілем послуг захисту інформації. Здійснюється розробка,

оформлення та затвердження робочої та експлуатаційної документації, передбаченої ТЗ на КСЗІ. Робоча документація повинна містити детальні рішення щодо реалізації технічних вимог, тестування, проведення пусконаладжувальних робіт, випробувань КСЗІ.

6. Випробування КСЗІ.

Випробування реалізованого проекту КСЗІ розподіляється на

- попередні випробування,
- дослідну експлуатацію та
- проведення державної експертизи.

Метою **попередніх випробувань** є перевірка працездатності, КСЗІ та перевірка можливості прийняття її у дослідну експлуатацію. Попередні випробування проводяться за програмою і методиками, які готуються розробником КСЗІ і узгоджуються із замовником. Для проведення випробувань замовником КСЗІ створюється комісія, головою комісії призначається представник замовника. Результати попередніх випробувань оформлюються «протоколом випробувань», де вказуються недоліки, виявлені у ході випробувань, заходи та терміни їх удосконалення. Після усунення недоліків та відповідного коригування проектної та експлуатаційної документації оформлюється акт про прийняття КСЗІ у дослідну експлуатацію.

Під час **дослідної експлуатації** доопрацьовується (за необхідністю) програмне забезпечення, експлуатаційна документація. За результатами дослідної експлуатації складається акт щодо можливості (неможливості) проведення державної експертизи КСЗІ.

Державна експертиза КСЗІ проводиться з метою визначення відповідності її вимогам ТЗ на її створення, можливості передання її в експлуатацію. Порядок проведення державної експертизи викладений у [23]. Організація виконавець експертизи за результатами аналізу наданих матеріалів формує програму та окремі методики проведення експертизи. Програма проведення експертизи погоджується із замовником та уповноваженим органом, окремі методики – з уповноваженим органом. Результати роботи оформлюються у вигляді протоколів за підписом експертів – виконавців роботи. Організація-виконавець експертизи складає експертний висновок, у разі затвердження якого уповноваженим органом організації-виконавцю надається зареєстрований Атестат відповідності.

3.4 Ідентифікація загроз «нульового дня» для цифрової системи управління за допомогою таксонометричного підходу

Загрози, що зустрічались раніше, зазвичай ідентифікують за їх сигнатурою (набором характерних ознак). **Загрози «нульового дня»** до їх першої появи ніхто не зустрічав, тому їх сигнатури невідомі, і такі загрози є найбільш небезпечними. Невідомо як їх ідентифікувати та невідомо, як їм протидіяти. В той же час, навіть, якщо не знайдені способи захисту від загрози «нульового дня», але ідентифікований хоча б сам факт наявності загрози, це дозволяє прийняти запобіжні заходи. Виявлення факту наявності загрози є репрезентативною інформацією для прийняття управлінського рішення щодо застосування інструментів протидії атаці (наприклад, спочатку карантин, а потім більш специфічна протидія). Недоліками існуючих підходів є потреба у формуванні еталонних значень для ситуацій, які швидко змінюються, для об'єктів, набір яких також швидко змінюється.

Більшість таких об'єктів співпрацює з ІС через глобальну мережу. Виникає задача оперативної **кластеризації** об'єктів по групах загрозливості для нашої системи. Виходячи з великого обсягу інформації, високої мінливості структур та їх параметрів, потрібно, щоб кластеризація виконувалась з мінімальним залученням людини. Крім того, якщо об'єкти постійно змінюють моделі своєї поведінки, то складним стає питання обрання еталону «нормальної поведінки», відносно якої ми будемо визначати рівні дисфункціональності та, відповідно, безпеки.

В такій ситуації доречним є **таксонометричний** метод [36, 45, 48, 49, 51], який дозволяє формувати еталони оперативно і без участі людини. Адаптуємо таксонометричний метод під задачу визначення еталонів поведінки об'єктів за ознаками безпеки.

Нехай в нас є n інформаційних об'єктів, що взаємодіють з нашою інформаційною системою. Система інформаційної безпеки веде постійний моніторинг цих об'єктів за формальними показниками, наприклад:

- Частота звернень;
- Обсяги обміну вхідної та вихідної інформації (за тематичними категоріями);
- Кількість дій, що є легальними, але наближені до таких, що можуть впливати на безпеку ІС;
- Кількість невдалих сеансів зв'язку з небезпечною або невизначеною причиною збою;
- Кількість невдалих спроб доступу (помилки в паролі, в логіні);
- Кількість порушень ролей та повноважень користувачів;
- Кількість роботизованих звернень (та окремо всі інші дії веб-ботів);
- Кількість спроб доступу до системного рівня ІС тощо.

Перелік контролюємих показників (факторів впливу) може змінюватись відповідно до:

- Цілей функціонування системи захисту;
- Загального стану та сценаріїв розвитку стану інформаційної безпеки організації або в цілому галузі, держави, регіону тощо.

Загальну кількість показників (факторів) вважаємо рівними m . Якщо при кластеризації нових інформаційних об'єктів інформація за певними показниками відсутня, то для збереження працездатності наведених нижче процедур, такі показники, до моменту їх реального визначення, вважаються рівними показникам еталону. Після отримання першого набору показників для всіх об'єктів, що спостерігаються, створюється відповідна матриця (Рис. 3.4.1).

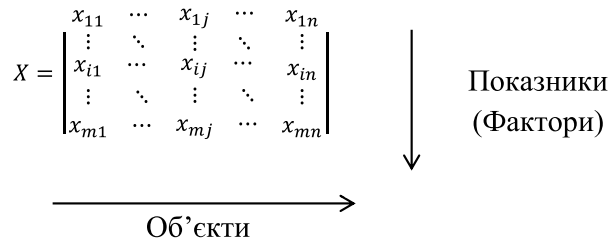


Рис. 3.4.1. Матриця показників об'єктів ІС.

Знайдемо математичне очікування та середнє квадратичне відхилення для кожного з показників.

$$M_i = M[x_{ij}] = \frac{1}{n} \sum_{j=1}^n x_{ij}$$

$$D_i = M[x_{ij} - M_i]^2 = \frac{1}{n} \sum_{j=1}^n (x_{ij} - M_i)^2$$

$$\sigma_i = \sqrt{D_i}$$

Для полегшення порівняння показників між собою, приведемо їх до єдиної міри шляхом нормування кожного елемента первинної матриці показників

$$z_{ij} = \frac{x_{ij} - M_i}{\sigma_i}$$

$$Z = \begin{bmatrix} z_{11} & \cdots & z_{1j} & \cdots & z_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{i1} & \cdots & z_{ij} & \cdots & z_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{m1} & \cdots & z_{mj} & \cdots & z_{mn} \end{bmatrix}$$

Далі потрібно побудувати еталонний безпечний інформацій об'єкт, тобто створити набір еталонних показників, які і будуть прийняті в якості характеристик еталонного об'єкту.

$$Z_{etalon} = \begin{bmatrix} Z_{1_etalon} \\ \vdots \\ Z_{i_etalon} \\ \vdots \\ Z_{m_etalon} \end{bmatrix}.$$

В класичному таксонометричному підході еталонним зазвичай вважають найменші або найбільші показники, серед наявних (Рис.3.4.2). По горизонталі наведені номери параметрів. По вертикалі їх значення. Кожна лінія відповідає окремому об'єкту. Товста лінія відповідає еталонному об'єкту (в цьому випадку за критерієм максимуму).

$$Z_{i_etalon} = \min_{j=1}^n (z_{ij}) \quad \text{або} \quad Z_{i_etalon} = \max_{j=1}^n (z_{ij}).$$

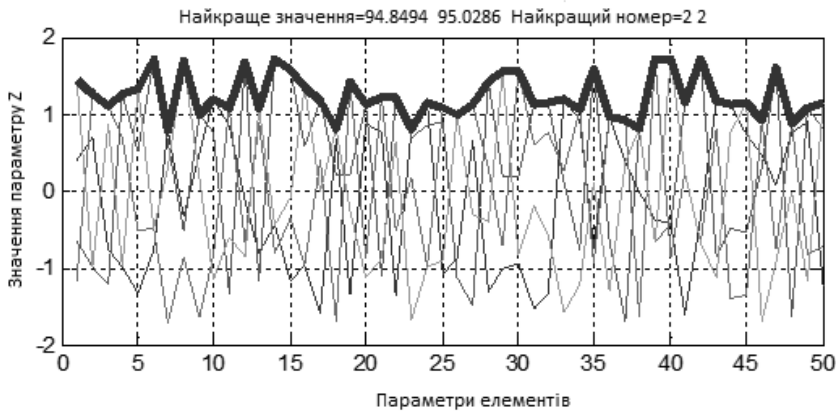


Рис. 3.4.2. Поточні значення параметрів для набору об'єктів та параметри еталонного об'єкту (мажоранта).

В нашому випадку так можна поступати, якщо величина функціональної стійкості монотонно залежить від величини показника. Нажаль динаміка зміни функціональної стійкості часто настільки складна та нелінійна, що така міра еталону буде неадекватною. В нашому випадку більш адекватною буде **міра типовості**, за якою найбільш безпечними можна вважати об'єкти з найбільш типовою поведінкою (у нормальному стані).

Розглянемо діапазон показників, що відповідають нормальному стану. Якщо такий діапазон для конкретного показника невідомий, то виходимо з того, що більша частка об'єктів, що спостерігаються до появи загроз не мають дисфункцій. Вивчення статистики показало, що до початку та на початкових стадіях епідемії в нормальному стані буде знаходитись від 95 до 99 % об'єктів системи. Це означає, що для виділення статистичних даних щодо об'єктів з нормальною поведінкою достатньо відкинути від 1 до 5% значень з найбільшими відхиленнями від середніх значень. Чим більше загроза

порушення нормального стану, тим більшу частку вибірки потрібно відкидати. Після цього для нової (обрізаної) виборки знаходимо нове математичне очікування, яке і обираємо у якості еталону (середини діапазону нормальних значень). Середні значення у якості міри найбільш безпечного стану можуть бути адекватними як для технічних (інформаційних), так і для біологічних систем

$$Z_{i_etalon} = \underset{j=1}{M}^n (z_{ij}) = \frac{1}{n} \sum_{j=1}^n z_{ij}.$$

Більш адекватним буде зважене середнє з ваговими коефіцієнтами β_j , які в оцінці будуть збільшувати вагу більш перевірених, більш надійних інформаційних об'єктів

$$Z_{i_etalon} = \underset{j=1}{M}^n (z_{ij}) = \frac{1}{n} \sum_{j=1}^n \beta_j z_{ij}.$$

Якщо такої впевненості немає, то всім ваговим коефіцієнтам привласнюється однакове значення. Обов'язковою умовою використання вагових коефіцієнтів є їх нормування $\sum_{j=1}^n \beta_j = 1$.

У якості іншої модифікації класичного таксонометричного підходу пропонується використання різних способів пошуку еталону для різних показників (мінімум, максимум, зважене середнє тощо). Загальний вираз для еталону в такому випадку запишемо у вигляді

$$Z_{i_etalon} = \underset{j=1}{etalon}^n (z_{ij}).$$

В якості міри віддаленості кожного з об'єктів Z_j від еталону Z_{etalon} використаємо квадрат евклідової відстані у m -вимірному просторі показників

$$R_j = \sum_{i=1}^m (z_{ij} - Z_{i_etalon})^2.$$

Найближчі об'єкти є найбільш безпечними. Найвіддаленіші - найбільш небезпечні або найбільш функціонально нестійкі. Додаткова аналітична інформація може бути отримана з аналізу зміни у часі віддаленості об'єктів від еталону, з прогнозування динаміки функціональної нестійкості та з визначення основних тенденцій. Серед нових тенденцій можуть бути як збільшення або поява нових множин небезпечних об'єктів, так й повернення небезпечних об'єктів в безпечний стан.

Чисельна реалізація методу виконана в обчислювальному середовищі Матлаб, яке забезпечує високу швидкодію матричних операцій. Результати моделювання наведені у вигляді дендрограми (Рис.3.4.3), в якій по горизонталі

розташовані номери об'єктів спостереження, а по вертикалі їх евклідова відстань від еталону в просторі показників.

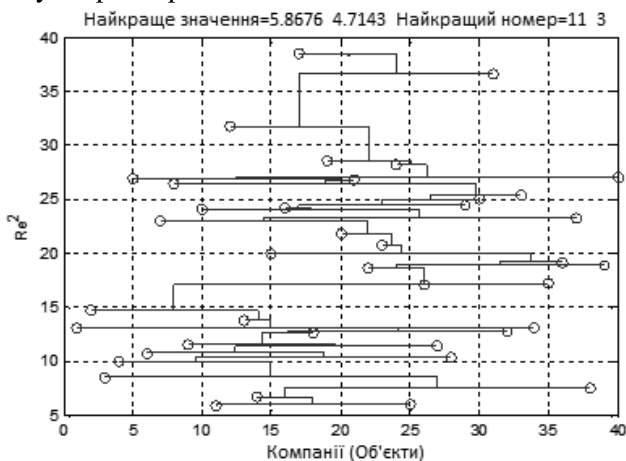


Рис. 3.4.3. Дендрограма об'єктів спостереження.

Об'єкти з найменшими відстанями від еталону є найбільш безпечними. Є певні групи об'єктів віддалених від еталону, але їх багато і за тривалим спостереженням їх можна об'єднати в кластер нетипових, але безпечних об'єктів. Хоча також можливо, що це буде кластер підготовки масованої атаки. Конкретний тип кластеру визначається за всією сукупністю ознак і аналізу конкретної ситуації. Але одне незаперечно, що виявлені найвіддаленіші об'єкти потребуватимуть додаткової уваги. Найбільш підозрілими завжди виглядають одиночні об'єкти в зоні великих відстаней. В нашому випадку – це об'єкти за номерами 12, 17, 31.

Основною проблемою чисельної реалізації методу є велика кількість n інформаційних об'єктів, які потрібно досліджувати. Моделювання показало, що реалізація запропонованого алгоритму можлива навіть на звичайній бюджетній обчислювальній техніці. Час розрахунків для 4000 об'єктів спостереження за 50 параметрами склав 2.136149 секунди, без врахування часу на реалізацію графіки. Перекладання алгоритму, наприклад, на алгоритмічну мову C++, надасть збільшення швидкодії розрахунків в 7-8 разів.

Адаптований в роботі таксонометричний метод дозволяє в автоматичному режимі виділяти потенційно дисфункціональні об'єкти для їх відслідковування з метою організації вчасної протидії.

Універсальність підходу полягає в тому, що математична постановка задачі ідентифікації загрози «нульового дня» для інформаційних та біологічних об'єктів має аналогічний вигляд.

Скорегуємо постановку задачі для прикладу **епідемій в суспільстві**.

Нехай в нас є виборка щодо n станів людей, що є об'єктами спостереження. Це можуть бути різні люди, або ті самі люди в різні моменти

часу, або те й інше. Наша інформаційна система веде постійний моніторинг цих об'єктів за формальними показниками, об'єктивних досліджень та суб'єктивних даних, наприклад:

- Температура, пульс, тиск, показники крові, дані досліджень з використанням медичної діагностичної апаратури;
- Суб'єктивні скарги пацієнтів;
- Суб'єктивні дані експертів-лікарів (консилиуму) тощо.

Набір показників може бути довільним. Єдина вимога - вони мають бути вимірюваними, тобто мати чисельне визначення в довільних шкалах. Перелік контролюємих показників (факторів) може змінюватись відповідно до:

- Захворювань, які були в людини раніше [84, 120];
- Зовнішніх умов, в яких знаходиться об'єкт спостереження;
- Загального тону людини тощо.

Висновки до розділу 3

1. При побудові захищеної системи управління динамічним об'єктом, таким, як БПЛА, потрібно враховувати динаміку його руху.

2. Прийоми, що були використані при створенні моделі динаміки польоту літального апарату (ЛА) є універсальними і можуть використовуватись при створенні моделей інших складних систем.

3. В роботі розроблена бажана послідовність нарощування рівнів захищеності нашої системи як послідовність зміни варіантів стандартних функціональних профілів захищеності. Переваги розробленої послідовності полягають в тому, що вони мінімізують ризики переходів між різними рівнями захисту інформаційної системи з одного боку, та у чіткої відповідності діючим нормативним документам з іншого боку.

4. Вимоги щодо захисту інформації мають бути враховані вже на етапі проектування та створення програмного забезпечення, яке входить до складу автоматизованої системи управління. Але на цьому робота щодо захисту не закінчується. Наступною складовою має бути створення комплексної системи захисту інформації.

5. Найбільшу небезпеку цифровим системам управління (зокрема й БПЛА) складають загрози «нульового дня» вивчення загроз. Тобто загрози абсолютно нові, про які нічого не відомо. Недоліками існуючих підходів є потреба у формуванні еталонних значень для ситуацій, які швидко змінюються, для об'єктів, набір яких також швидко змінюється.

6. Адаптований в роботі таксонометричний метод дозволяє в автоматичному режимі виділяти потенційно небезпечні об'єкти в мережі управління за ознаками нетиповості їх поведінки для організації вчасної протидії.

7. Універсальність підходу полягає в тому, що математична постановка задачі ідентифікації загрози «нульового дня» для інформаційних та біологічних об'єктів має аналогічний вигляд. Тобто підхід може бути застосований як для технічних (цифрових) систем, так й для біологічних.

4. ОБРОБКА 3-D СИГНАЛІВ МЕДИЧНОГО ПРИЗНАЧЕННЯ В ІНТЕРЕСАХ ІОЕ

4.1 Побудова сучасного сервіс-орієнтованого додатку для аналізу динамічних даних медичного призначення в інтересах Інтернету всього

Актуальність роботи обумовлена зростаючою необхідністю цифрового перетворення подій та станів об'єктів спостереження, їх класифікації та автоматизованого прийняття рішення. На даний час існує достатньо методів та технологій для оцифровки будь-яких сигналів, але більшість з них припускає наявність стабільного високошвидкісного з'єднання між компонентами системи. В цілому ряді випадків використання такого типу з'єднань не є можливим і як альтернатива застосовуються технології IoT, що мають суттєві обмеження в пропускну здатності. Оптимізація з'єднань, виділення сценаріїв використання та моделювання передачі даних в таких мережах є **актуальною** задачею через вимогу практики впровадження технології та недостатність теоретичного моделювання такого роду систем та процесів.

Метою роботи є покращення ефективності передачі даних та інформації про події та стани об'єкту спостереження, з використанням мереж з низькою пропускну здатністю.

Об'єкт роботи - інформаційна технологія, що здатна зменшити навантаження на мережу з низькою пропускну здатністю, за рахунок мінімізації об'єму даних, що передається між пристроями системи.

Предмет роботи створення програмного забезпечення та алгоритму що має на меті передачу цифрових даних від кінцевого пристрою до користувача у мережах типу LPWAN найбільш оптимальним шляхом та обробку отриманої інформації.

При виконанні досліджень та прийнятті рішень використовувались такі **підходи**: емпіричні дослідження (прототипування), теоретичні дослідження, трасування вимог (перевірка рішення відносно висунутих вимог), порівняння альтернатив та пошук найкращого рішення, математичне моделювання автоматичних систем.

4.1.1 Вступ

Прототипи мереж LPWAN з'явились у 1980-1990 рр. Але ці мережі поступилися місцем більш ефективним технологіям. Сучасні LPWAN [106] мережі можуть скласти конкуренцію навіть таким технологіям як 5G. Уся справа в покритті, низькій собівартості обладнання та обслуговування. LPWAN – сервіси незамінні в керуванні активами, будівлями, реалізації концепції «розумне місто», розумному обліку ресурсів та сільському господарстві [21]. Іншим аспектом використання є медицина та системи цивільного захисту.

Як можлива реалізація широкополосної мережі малої потужності (LPWAN), технологія дальнього діапазону LoRa [90] вважається майбутнім стандартом бездротового зв'язку в Інтернеті речей, оскільки пропонує конкурентоспроможні функції, такі як широкий діапазон зв'язку, низька вартість та зменшене енергоспоживання, що робить її оптимальною альтернативою нинішнім бездротовим сенсорним мережам та звичайним стільниковим технологіям. Однак обмежена пропускна здатність, доступна для модуляції фізичного рівня в LoRa, робить її непридатною для передачі даних з високою швидкістю, таких як відеозображення [116]. Слід також зауважити, що в сучасному світі пристрої працюють далеко не в ідеальних умовах і дуже часто «заважають» один одному, що призводить до помилок під час передачі даних [95]. Звичайно це призводить до обмежень у конфігуруванні та специфічному нетривіальному налаштуванні системи [20]. Саме тому аналіз стану об'єктів спостереження і своєчасне реагування на зміни середовища, це те що робить досвід використання сучасних технологій якісно «іншим».

4.1.2 Аналіз транспортного шару повідомлення

Транспортний шар, має суттєвий вплив на поведінку протоколу обміну даними. Останні дослідження [101] демонструють, що TCP в незначній мірі кращій за UDP навіть у середовищі з обмеженнями. Іншим важливим дослідом [62] продемонстрував, що оптимальним транспортним шаром є DCCP. Важливо зазначити, що останній припускає що втрачений пакет немає сенсу слати ще раз, тому що він не додасть цінності кінцевому споживачу. Виходячи з цього є можливість припустити, що для даних IoT можливо спостерігати таку саму властивість. Процес передачі даних в мережах IoT можливо представити за допомогою математичної моделі для неперервних систем (Рис 4.1.2.1).

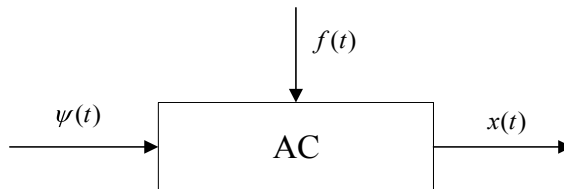


Рис 4.1.2.1. Автоматична система.

де $\psi(t)$ - функція зовнішнього сигналу,
 $f(t)$ - функція сбурення,
 $x(t)$ - функція вихідного сигналу.

Данні що можливо отримати з сенсорів, сильно відрізняються за способом вимірювання, точністю та частотою отримання інформації. Але не зважаючи на це їх все ж таки можливо класифікувати з точки зору способу передачі даних а отже побудувати математичну модель.

З метою узагальнення та лінерізації пропонується прийняти таку класифікацію для даних:

- **Пакетні** – фотографії мабуть кращий приклад.
- **Потокові данні** – температурні сенсори, GSM. Саме для цієї умовної групи отримання останніх даних може мати набагато більше сенсу ніж пересилка втраченого пакету.
- **Динамічні поточкові данні** – як дані акселерометру. Отримання останніх даних все ще важливо, але втрата пакетів може призвести до хибно-позитивних результатів на стадії аналізу.
- **Стиснуті (зашифровані) данні** – будь-які данні рівня додатку, що було попередньо стиснуто або зашифровано.

4.1.2.1 Ситуативне моделювання втрати пакетів

Визначемо функцію збурення як функцію Гевісайда [94]:

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (4.1.2.1.1)$$

де $x \geq 0$ прийемо за успішну доставку блоку даних, тоді пропонуємо визначити передавальну функція $\phi(t)$ таким чином:

$$\phi(t) = \frac{x(t)}{\psi(t)} = \frac{\psi(t)f(t)}{\psi(t)} + a_i f(t)^{-1} = f(t) + a_i f(t)^{-1} \quad (4.1.2.1.2)$$

$$D(\phi) = (-\infty, 0) \cup (0, +\infty) \quad (4.1.2.1.3)$$

де $\psi(t)$ - вхідний сигнал,

a_i - деяке значення, що застосовується коли відсутній пакет з повідомленням/сигналом,

$\phi(t)$ - вихідний сигнал,

$f(t)$ - функція збурення (4.1.2.1.1),

$D(\phi)$ - область визначення функції $\phi(t)$.

4.1.2.2 Експериментальні сценарії

З метою отримання більш достовірної інформації, щодо природи процесів було проведено експеримент, що мав на меті практичне моделювання застосування різних способів обробки втрачених даних під час передачі.

Були обрані такі сценарії взаємодії:

А. Втрата 5% даних (черговість не відслідковується)

$$\phi(t) = f(t) + a_i f(t)^{-1}, a_i = 0 \quad (4.1.2.2.1)$$

В. Втрата 5% даних та зміна порядку отримання для 5% (черговість не відслідковується)

$$\phi(t) = g(t) \quad (4.1.2.2.2)$$

де $g(t)$ функція трансформації сигналу.

С. Втрата 5% і заміна втрачених даних деяким значенням – наприклад 0 для датчика температури (черговість відслідковується).

$$\phi(t) = f(t) + a_i f(t)^{-1}, a_i \in \square, a_i \neq 0 \quad (4.1.2.2.3)$$

4.1.2.3 Пакетні і стиснуті дані

Очевидно що методологія оцінки для різних типів даних теж відрізнятиметься. Було вирішено почати зі стиснутих даних на прикладі фотозображення у форматі JPG, та пакетних даних – фотографія у форматі TIFF. (Рис. 4.1.2.3.1)



Рис. 4.1.2.3.1. Зображення у форматі TIFF зліва, JPG справа.



Рис. 4.1.3.2.2. Сценарії А, В, С для JPG зліва-направо.



Рис. 4.1.3.2.3. Сценарій В для TIFF.

Щодо стиснутих зображень у форматі JPEG експеримент зі стисненими даними продемонстрував очевидну річ, що навіть якщо між рисунками є очевидна схожість, візуально є можливість спостерігати значні відмінності (Рис. 4.1.2.3.2). Сценарії А, В для TIFF зображення не вдалось відкрити, це сталося через те що стандартне ПЗ не мало змоги вирахувати розмір зображення. Незважаючи на це заміна втрачених символів нулями дала несподівано кращі результати (Рис. 4.1.2.3.3). Є можливість візуально констатувати очевидну схожість зображень, а також припускати, що системи розпізнавання зможуть однаково класифікувати вхідне та вихідне зображення.

Таким чином можливо зробити наступні висновки: оскільки зображення не є потоком даних, то безпосередньо вихідний сигнал, що є об'єктом пошуку, є суммою обробки кожного окремого сигналу в межах однієї сесії. А тому функцію для зображення пропонуємо визначити як

$$im = \sum_{t=0}^n x(t). \quad (4.1.2.3.1)$$

Аналогічним чином пропонуємо визначити вхідний сигнал і таким чином визначимо фактично втрачені під час передачі дані

$$\xi = \Omega(\sum_{t=0}^n \psi(t), \sum_{t=0}^n x(t)). \quad (4.1.2.3.2)$$

Для знаходження ξ застосуємо специфічні для обробки зображень методи а саме індекси PSNR та SSIM [85], що можуть бути розраховані за формулами (4.1.2.3-4.1.2.9)

$$PSNR = 10 \log \frac{s^2}{MSE} \quad (4.1.2.3.3)$$

$$MSE = \frac{1}{nm} \sum_{m=0}^{m-1} \sum_{n=0}^{n-1} e(m,n)^2 \quad (4.1.2.3.4)$$

$$SSIM(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (4.1.2.3.5)$$

$$\mu_x = \frac{1}{T} \sum_{i=1}^T x_i \quad \mu_y = \frac{1}{T} \sum_{i=1}^T y_i \quad (4.1.2.3.6)$$

$$\sigma_x^2 = \frac{1}{T-1} \sum_{i=1}^T (x_i - \bar{x})^2 \quad \sigma_y^2 = \frac{1}{T-1} \sum_{i=1}^T (y_i - \bar{y})^2 \quad (4.1.2.3.7)$$

$$\sigma_{xy}^2 = \frac{1}{T-1} \sum_{i=1}^T (x_i - \bar{x})(y_i - \bar{y}) \quad (4.1.2.3.8)$$

$$MSSIM = \frac{1}{p} \sum_{j=1}^p SSIM_j \quad (4.1.2.3.9)$$

де MSSIM – (*Mean Structural Similarity Index Method*) середній індекс структурної схожості.

Оскільки блок даних з графічним зображенням складається з декількох пакетів в мережах LoRaWAN, то з урахуванням (4.1.2.3.9) пропонуємо застосовувати алгоритми PSNR (*Peak-signal-to-noise-ratio* – співвідношення сигналу до шуму) та SSIM [10] для порівняння зображень наступним чином (Табл. 4.1.2.3.1 та Табл. 4.1.2.3.2) :

$$\xi_{PSNR} = PSNR(\sum_{t=0}^n \psi(t), \sum_{t=0}^n x(t)) \quad (4.1.2.3.10)$$

$$\xi_{SSIM} = MSSIM(\sum_{t=0}^n \psi(t), \sum_{t=0}^n x(t)) \quad (4.1.2.3.11)$$

де ξ_{PSNR} вихідний PSNR між стиснутим та оригінальним зображенням, ξ_{SSIM} вихідний MSSIM між стиснутим та оригінальним зображенням.

Таблиця 4.1.2.3.1.

Результати аналізу стиснутих даних (JPG)

Експеримент	ξ_{SSIM}	ξ_{PSNR}
5% Втрат	0.81	11.5
5% Втрат та 5% перестановок	0.59	7.65
5% Втрат та заміна нулями	0.84	13.15

Таблиця 4.1.2.3.2.

Результати аналізу не стиснутих даних (TIFF)

Експеримент	ξ_{SSIM}	ξ_{PSNR}
5% Втрат	N/A	N/A
5% Втрат та 5% перестановок	N/A	N/A
5% Втрат та заміна нулями	0.96	20.7

Таким чином можливо зробити висновок, що для блочної передачі даних коли інтерпретація даних не залежить від наявності усіх повідомлень (приклад TIFF), можливо досягти достатньо високих показників. В зворотньому випадку ми спостерігаємо значну деградацію якості. В експериментальному випадку ми бачили втричі більші втрати значущої інформації ξ_{SSIM} і вдвічі знижену якість сигналу ξ_{PSNR} . Слід зауважити також, що не здатність розпізнавати черговість повідомлення в межах ланцюга демонструє найгірший результат.

4.1.2.4 Потоківі та динамічні потоківі дані

Розглянемо, як буде себе поводити система у випадках з поточковими даними. Як вже було зазначено прикладом таких даних є дані сенсорів таких як – температурні, акселерометри, геолокації. На відміну від попереднього прикладу ми маємо змогу брати будь який блок для порівняння (4.1.2.4.1)

$$[a, b] = \{t \in N \mid a \leq t \leq b\} \quad (4.1.2.4.1)$$

Для поточкових даних, ми можемо використати коефіцієнт Пірсона (4.1.2.4.2), щоб класифікувати схожість вхідного та вихідного сигналів.

$$r_{XY} = \frac{n(\sum_{i=a}^b XY) - (\sum_{i=a}^b X)(\sum_{i=a}^b Y)}{\sqrt{[n\sum_{i=a}^b X^2 - (\sum_{i=a}^b X)^2][n\sum_{i=a}^b Y^2 - (\sum_{i=a}^b Y)^2]}} \quad (4.1.2.4.2)$$

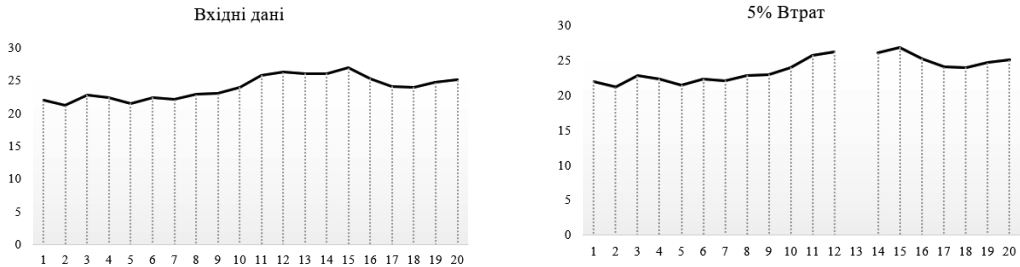


Рис. 4.1.2.4.1. Вхідні дані та Сценарій А.

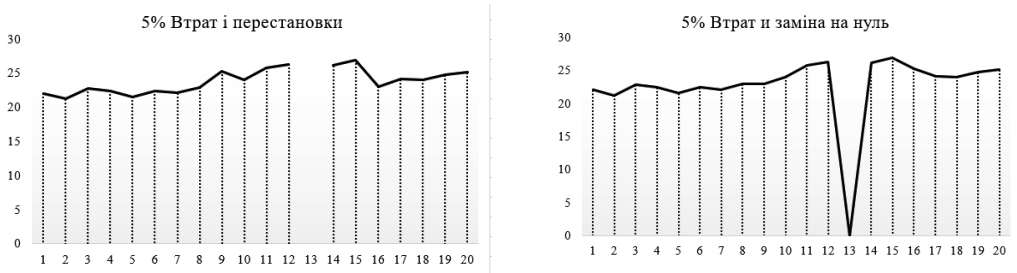


Рис. 4.1.2.4.2. Сценарій В та С.

Симуляція для поточкових даних була виконана для того самого набору сценаріїв, а в якості вхідних даних були взяті відомості про середньоденну температуру у літній період на протязі 20 діб. (Рис. 4.1.2.4.1-4.1.2.4.3)



Рис. 4.1.2.4.3. Сценарій В та екстраполяція для пропущеного елемента.

Для того, щоб кореляція Пірсона працювала коректно втрачені дані заповнюються із прогноз-моделі тому що не відомо коли їх очікувати. Тому в масштабі реального часу застосовується екстраполяція на підставі існуючих даних а по завершенню сеансу під час ретроспективного аналізу можливе

уточнення з використанням інтерполяції. Це дозволить виконати розрахунок і оцінити схожість. (Табл. 4.1.2.4.1)

$$x(t_i) = x(t_{i-1}) + \Delta x \quad (4.1.2.4.3)$$

$$\Delta x = x(t_{i-1}) - x(t_{i-2}) \quad (4.1.2.4.4)$$

Таблиця 4.1.2.4.1.

Результати аналізу поточкових даних

Експеримент	r_{XY}
5% Втрат	94%
5% Втрат та 5% перестановок	78%
5% Втрат та заміна нулями	1%
5% Втрат та екстраполяція	99%

Експеримент продемонстрував приблизно ті самі результати, що й попередній. Заміна втрачених пакетів заздалегіть відомими значеннями працює. На відміну від попереднього випадку кращим вибором є функція екстраполяція. Змішані (переставлені місцями пакети) виглядають найгірше, що підтверджує твердження про важливість послідовності отримання.

Для динамічних поточкових даних були використані дані акселерометра по осі X як вхід, та проведені ті самі симуляції (Табл 4.1.2.4.2), що й в попередньому випадку. (Рис 4.1.2.4.4-4.1.2.4.6)

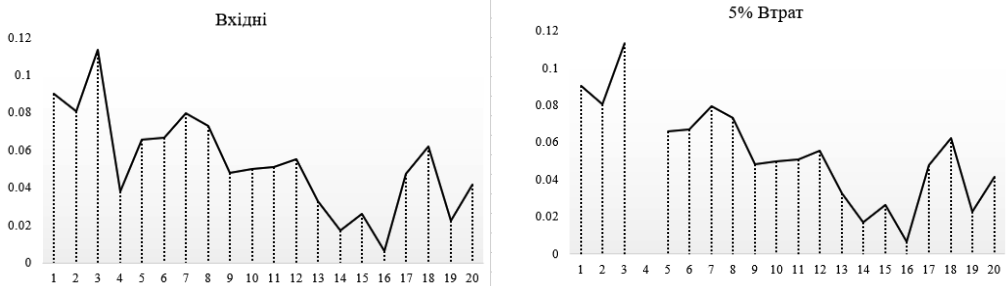


Рис. 4.1.2.4.4. Сценарій А та вхідні дані.

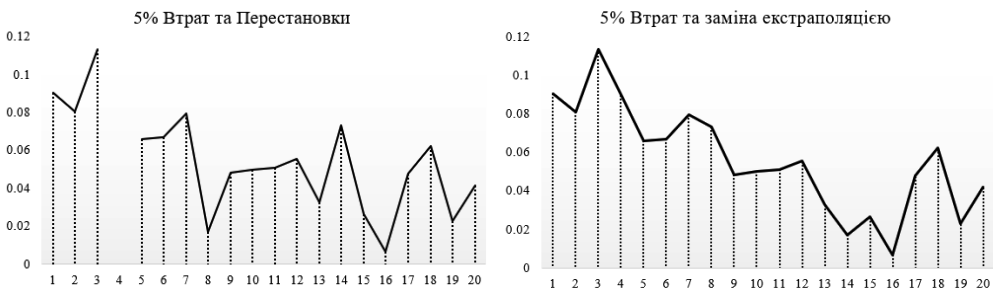


Рис. 4.1.2.4.5. Сценарій В та С – втрати з перестановками та екстраполяція для пропущеного елемента.



Рис. 4.1.2.4.6. Сценарій С – втрати з заміною нулями.

Таблиця 4.1.2.4.2

Результати аналізу потокових даних

Експеримент	r_{xy}
5% Втрат	73%
5% Втрат та 5% перестановок	53%
5% Втрат та заміна нулями	95%
5% Втрат та інтерполяція	90%

Симуляція втрат пакетів продемонструвала, що послідовність пакетів має суттєвий вплив на якість сервісу передачі даних IoT, у той же час втрата частини даних може бути компенсована гнучкими алгоритмами та методами. Також слід зазначити відсутність суттєвих відмінностей між різними типами даних. DCCP протокол згаданий раніше було розроблено для підтримки VoIP комунікації і вже має імплементацію найбільш значущих функцій, але це не може бути застосоване у протоколах прикладного рівня таких як, CoAP, MQTT-SN. LoRAWAN [57] взагалі використовує свою специфікацію.

4.1.3 LoRaWAN як транспорт для медіа-контенту

4.1.3.1 Використання LoRaWAN мережі для передачі графічного контенту

Очевидним є той факт, що постійна передача великих за розміром зображень у мережах з обмеженою пропускну здатність не є можливою, через цілу низку причин. Серед них:

- Передатчики будуть заважати один одному.
- Пропускна здатність зеднання [98].
- Інші технологічні обмеження

Передача даних з високим біт рейтом між пристроями, такими як сенсори зображень з використанням мережі типу LoRa очевидно буде дуже повільним. Така інформація перш за все не дуже підходить для передачі з такою низькою пропускну здатністю, в тому числі через обмеження LoRa в 1% робочого часу

(36 сек/час), що означає те що типовою є передача на протязі 36 сек щогодинно. LoRa MAC рівень зазвичай відповідальний за передачу даних з кінцевих станцій до шлюзів. Але він має обмеження по роботі з зображеннями через розмір та шифрування яке використовує MAC рівень. Проблему було вирішено і експериментальним шляхом доведено можливість передачі такого роду даних, за рахунок модифікація рівня шифрування [87].

Експериментальні дані можливості передачі зображення з використаннями LoRaWAN продемонстрували, що навіть в умовах очікування доставки усіх пакетів така можливість існує [117], прототипування передачі поодиноких зображень в реальних умовах [87] та збір статистичних даних о потрібних налаштуваннях пристроїв для пристроїв LoRa також був виконаний для стиснутих зображень з використанням алгоритму JPEG.

На підставі цього можливо зробити висновок про те що можливість постійної (з деякою частотою: щогодинно, що 4 години, щоденно, за розкладом і т. п.) передачі зображень існує, і ключові обмеження лежать в площині:

- Дистанція.
- Об'єм даних.
- Час на передачу.

4.1.3.2 Розмір зображення – критична велечина

Очевидно, що найбільш впливовим параметром є об'єм даних, що у тій чи іншій мірі формує вихідне значення для часу на передачу та граничну дистанцію для передачі. Перш за все зазначимо, що пристрої LoRa здатні передавати сигнал у наступній ширині спектру сигналу BW – 125кГц, 250кГц та 500кГц. Врахуємо також коефіцієнт розширення спектру сигналу SF , що може приймати значення від 7 до 12. Таким чином швидкість передачі даних в мережах LoRa визначена [90]

$$R = SF \frac{1}{2^{SF} / BW}, \quad (4.1.3.2.1)$$

$$c = \frac{SI}{p_s}, \quad (4.1.3.2.2)$$

де c - кількість пакетів що надсилається, SI - розмір зображення, p_s - розмір пакету, що залежить від SF та BW і визначений специфікацією LoRa.

Залежність між часом на передачу та розміром є лінійною і може бути розрахована без урахування службової інформації (4.1.3.2.3) [11]

$$t \approx SI * R, \quad (4.1.3.2.3)$$

де t - час на передачу даних розміру SI .

Слід також зазначити, що зі збільшенням дистанції слід підвищувати SF , знижувати BW , а отже і зменшувати розмір пакету p_s [87, 90]. У той же час чим більша дистанція між вузлами тим більша втрата пакету

$P(A)$ та одночасно з цим кількість відправлень суттєво підвищує вирогідність отримати втрати при пересилці окремого зображення [57]

$$P(A_x) = 1 - \frac{D}{D_{\max}} + k, \quad (4.1.3.2.4)$$

де D - дистанція, D_{\max} гранична дистанція передачі пакету згідно з специфікацією LoRa, k - деяке значення що відображає фактори підсилення сигналу, такі як висота пристрою над землею, відсутність перешкод, наявність підсилювача тощо.

$$P(B_x) = 1 - P(A_x), \quad (4.1.3.2.5)$$

де $P(B)$ - вирогідність втрати пакету. Тоді використовуючи формулу Бернуллі визначимо вирогідність втрати пакету як

$$p_t = c_c^l P(B_x)^l P(A_x)^{c-1} = c P(B_x) P(A_x)^{c-1}, \quad (4.1.3.2.6)$$

де p_t вирогідність втрати хоча б одного пакету для зображення, c - кількість пакетів, виходячі з того що ми говоримо про послідовність із n незалежних експериментів, для яких вирогідність успіху допустими прийняти за однаковою то величина p_t матиме біноміальний розподіл тоді математичне очікування

$$M_p = c P(B). \quad (4.1.3.2.7)$$

Таким чином вплив вплив дистанції виражено через кількість пакетів що відправляється [57], а отже усі перелічені ознаки у той чи інший спосіб залежить від розміру.

4.1.3.3 Особливості застосування JPEG2000

Попередніми дослідженнями визначено, що найбільш ефективним засобом стиснення даних, є JPEG2000 [118]. В той же час використання цього алгоритму вимагає набагато більше процесорного часу, що теж слід врахувати. Стиснення в JPEG2000 відбувається з втратами а отже, якщо втрат буде забагато користувач системи не зможе отримати необхідну інформацію із зображення, що надіслано кінцевим пристроєм. Кількісна характеристика втрат регулюється параметром якості, що є PSNR (Point-to-signal-noise-ratio) вираженим в db. Для прототипування був застосований інструмент ImageMagic [86].

Очевидно, що необхідно знайти відповідь у який саме спосіб вимірювати достатність якості зображення. Таке завдання слід вирішувати від часткового випадка до загального, а отже визначимо спочатку можливі варіації способу оцінювання:

- система отримує зображення і оцифровує деякі об'єкти,
- системи отримує повідомлення і людина переглядає зображення.

Розглянемо перший випадок з автоматизованим оцінюванням, оскільки варіант з людиною є частковим випадком з точки зору процесу що відбувається. З точки зору системи що проводить оцифровку зображення, критерієм якості зображення, є очевидно можливість знайти необхідний об'єкт на зображенні і коректно повідомити системі про його наявність та властивості якщо він є. Не залежно від того, що є об'єктом пошуку на зображенні людина, дія, предмет чи будь-що ще, його знаходження пов'язано з наявністю ключових точок [92]. Таким чином пропонуємо визначити кількість значущих коренів (ключових точок) як

$$\lambda = \upsilon(x). \quad (4.1.3.3.1)$$

Якщо функція λ має дійсні корені і ці корені це ключові точки або інші характеристики зображення, що є достатніми для автоматизованої обробки зображення то пропонуємо визначити корені (ключові точки) вихідного зображення після стиснення як

$$\lambda_1 = \omega(\upsilon(x)), \quad (4.1.3.3.2)$$

де $\omega(x)$ - це функція стиснення.

Виходячи з цього ми можемо визначити ті точки, що співпали між оригінальним зображенням та стиснутим за допомогою таких алгоритмів як швидкий пошук найближчого сусіда [92]. А також гранично допустимий мінімум співпадінь, що може мати варіативність в залежності від області застосування і є користувацьким очікуванням якості: Визначимо корені (ключові точки) що співпадають як

$$\eta = Z(\lambda, \lambda_1), \quad (4.1.3.3.3)$$

де $Z(x, x_1)$ - функція пошуку спільних ключових точок. На підставі цього пропонуємо визначити втрати ξ та фактичний відсоток співпадань p як:

$$\xi = \lambda - \eta, \quad \xi \geq 0 \quad (4.1.3.3.4)$$

$$\xi \in [0, \lambda], \quad \xi \in \mathbb{N} \quad (4.1.3.3.5)$$

$$p = \frac{\xi}{\lambda}, \quad k_{\min} \leq p \leq 1, \quad p \in [0, 1], \quad \lambda > 0 \quad (4.1.3.3.6)$$

де k_{\min} - гранично допустимий мінімум співпадінь визначений користувачем

4.1.3.4 Приклад «Дитяча поліцейська машинка»

Для наочності роботи алгоритму JPEG2000 розглянемо це на прикладі дитячого автомобіля, що розміщено на поверхні з ефектом відзеркалення. Рис. 4.1.3.4.1 демонструє іграшку з яскравими кольорами. Отримане зображення Рис. 4.1.3.4.1 зменшимо до розмірів 800x600 та 1200x800 і стиснемо з різним показником якості PSNR (Якість наведена у Табл. 4.1.3.1 та Табл. 4.1.3.2). Табл. 3 та Табл. 4 підсумовують результати стиснення та демонструють вхідний показник якості PSNR та вихідні ξ_{PSNR} (4.1.2.3.10) - вихідний PSNR (Point-to-signal-noise-ratio) між стиснутим та оригінальним зображенням, ξ_{SSIM} (4.1.2.3.10)

- вихідний MSSIM (*Mean Structural Similarity Index Method*) між стиснутим та оригінальним зображенням, розмір у форматах JPEG2000 та JPEG. Цікавим спостереженням є те що в обох випадках при візуальному контролі було встановлено, що при вхідному параметрі якості PSNR=35 зображення має достатньо високу якість.



Рис. 4.1.3.4.1. Вхідні зображення А та Б.

Таблиця 4.1.3.4.1

Результати стискання для розміру 800x600

Якість (PSNR)	JP2(kb)	JPEG (kb)	ξ_{PSNR}	ξ_{SSIM}
10	1	2	12	53%
15	1	2	12	53%
25	2	63	24	74%
30	7	82	29	85%
35	21	100	34	92%
40	52	114	37	95%

Таблиця 4.1.3.4.2

Результати стискання для розміру 1200x800

Якість (PSNR)	JP2(kb)	JPEG (kb)	ξ_{PSNR}	ξ_{SSIM}
10	1	2	12	56%
15	1	2	12	56%
25	2	63	24	74%
30	8	82	29	84%
35	29	100	34	91%
40	82	114	38	95%

Графічний аналіз Рис 4.1.3.4.2, Рис. 4.1.3.4.3 та Рис. 4.1.3.4.4 дає змогу припустити близьку до потенційної залежності між показниками. Також очевидна не значна відмінність в розмірі на граничних показниках якості.

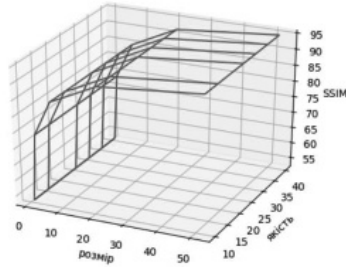


Рис. 4.1.3.4.2. 3D представлення даних Табл. 5 – вхідна якість (PSNR) [86], Розмір (kb) вихідного файлу 800x600 в форматі (JP2), ξ_{SSIM} і ξ_{PSNR} .

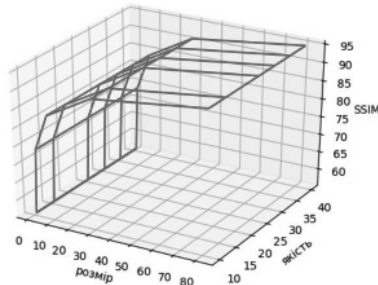


Рис. 4.1.3.4.3. 3D представлення даних Табл. 6 – вхідна якість (PSNR) [86], Розмір (kb) вихідного файлу 1200x800 в форматі (JP2), ξ_{SSIM} (4.1.2.3.10) і ξ_{PSNR} .

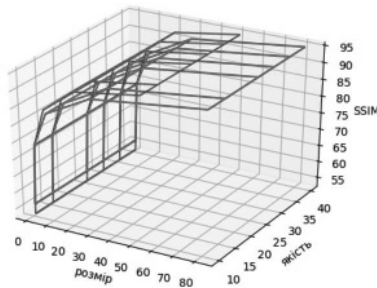


Рис. 4.1.3.4.4. Накладення графіку Рис 4.1.3.4.2 на Рис 4.1.3.4.3. Показники - вхідна якість (PSNR) [86], Розмір (kb) вихідного файлу в форматі (JP2), ξ_{SSIM} (4.1.2.3.10) і ξ_{PSNR} .

Із Рис 4.1.3.4.5. не важко побачити, що існує такий вхідний $PSNR \in [1,100]$, $PSNR \in \mathbb{N}$, що для всіх менших за нього характерний значний ріст вихідного ξ_{SSIM} , при малих змінах розміру, і навпаки для значно меншого приросту якості вимагає значно більшого об'єму даних (kb). Як вже було зазначено вище граничні показники втрат, є критерієм підбору оптимального значення вхідного параметра якості (PSNR) при стисненні алгоритмом JPEG2000, а тому доцільно визначитись з алгоритмом пошуку оптимуму.

Застосуємо властивості оптимальності по Парето, що визначає оптимальним «такий стан системи, що жодний показник не можливо покращити без погіршення інших» і побудуємо залежність вихідного розміру зображення від вхідної якості (PSNR) Рис. 4.1.3.4.5. Також розрахуємо коефіцієнти для функції регресії

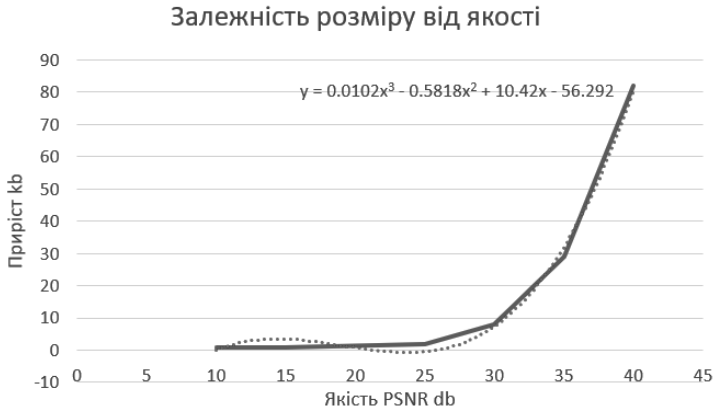


Рис. 4.1.3.4.5. Залежність вихідного розміру зображення від вхідної якості (PSNR) [86].

$$y = 0.0102x^3 - 0.5818x^2 + 10.42x - 56.292 \quad (4.1.3.4.1)$$

Для нашого випадку будь який приріст розміру файлу є погіршення стану системи через 4.1.2.3.10, а отже оптимум пролягає в межах мінімального необхідного об'єму для заданої користувачем якості зображення 4.1.2.3.11. Через те, що кожне зображення унікальне і не можливо розрахувати абсолютно точно всі показники заздалегіть, виникає потреба уточнювати. А отже алгоритм пошуку оптимуму ітеративний і, як наслідок, має початкові значення.

Для вирішення даної задачі пропонуємо виконувати процес уточнення ітеративним шляхом від деякого початкового значення якості зображення y_0 , до y_i , з заданим кроком h що задовольнить умові (4.1.3.4.2)

$$\begin{cases} y_{i+h} = y_i + h, h \in [0, 100], h \in \mathbb{N} \\ \bar{x}_{y+h} = \frac{\xi_{y+h}}{\lambda_{y_0}}, x_{y+h} \geq k_{\min} \end{cases} \quad (4.1.3.4.2)$$

Відкритим запитанням залишається оптимальний шлях вибору y_0 , такий щоб кількість ітерацій була мінімальною, перш за все через те, що процесорний час матиме негативні наслідки на енергоспоживання кінцевого пристрою. Для пошуку найкращого значення для y_0 проаналізуємо приріст для поліному функції регресії Рис 4.1.3.4.6.

Аналіз приросту проведемо за рахунок функції регресії, що у наближеній формі відтворює залежність між якістю і розміром з урахування кроку $h = 1$

$$\begin{aligned}
 y_1 - y_0 &= a_0 - a_0 + a_1(t_1 - t_0) + a_2(t_1^2 - t_0^2) + a_3(t_1^3 - t_0^3) \\
 &= (t_1 - t_0)(a_1 + a_2(t_1 + t_0) + a_3(t_1^2 + t_1t_0 + t_0^2)) \\
 &= a_1 + a_2(t_1 + t_0) + a_3(t_1^2 + t_1t_0 + t_0^2)
 \end{aligned}
 \tag{4.1.3.4.3}$$

Значення для $PSNR \leq 20$ Рис. 4.1.3.4.6 наочно ілюструють відсутність росту розміру файлу до деякої межі.

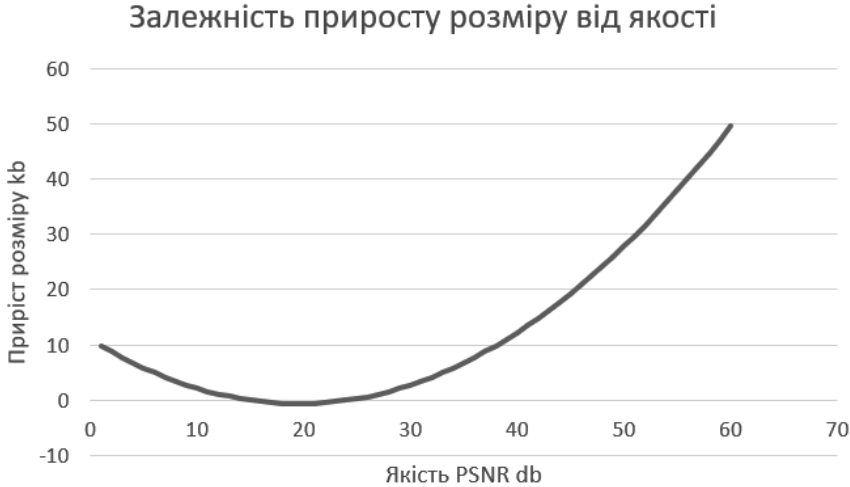


Рис. 4.1.3.4.6. Залежність приросту розміру вихідного файлу від вхідної якості PSNR (*Point-to-signal-noise-ratio*) [86].

Візуально, вищезазначений ефект ми можемо це спостерігати у вигляді відсутності будь-якої інформації на вихідному зображенні та появи розмитого малюнку Рис 4.1.3.4.7. Цікавим є те що ми спостерігаємо пропорцію характерну для правила Парето 80 - 20.



Рис. 4.1.3.4.7 Вихідне зображення на якому спостерігається зникнення усіх даних для $PSNR = 15, \xi_{SSIM} \approx 55$ та розміру 800x600 зліва 1200x800 зправа.

Таким чином можемо визначити нижню межу оптимальності $\xi_{SSIM} > 60$. Для того щоб встановити верхню границю вхідного PSNR дослідимо залежність вихідного розміру та вхідного PSNR від вихідного ξ_{SSIM} (*Mean Structural Similarity Index Method*)



Рис. 4.1.3.4.8 Вихідне зображення на якому спостерігається розмите зображення для $PSNR = 20$, $\xi_{SSIM} \approx 62$ та розміру 800×600 зліва 1200×800 зправа.

Рис 4.1.3.4.9 має перетин для $\xi_{SSIM} \approx 92$, $PSNR = 35$ таким чином можливо зазначити що верхня межа оптимальності може бути визначена як $\xi_{SSIM} < 95$

$$\begin{cases} 60 < \xi_{SSIM} < 95 \\ 20 \leq y_0 \leq 35 \end{cases} \quad (4.1.3.4.4)$$

Залежність розміру та вхідної якості від вихідного SSIM

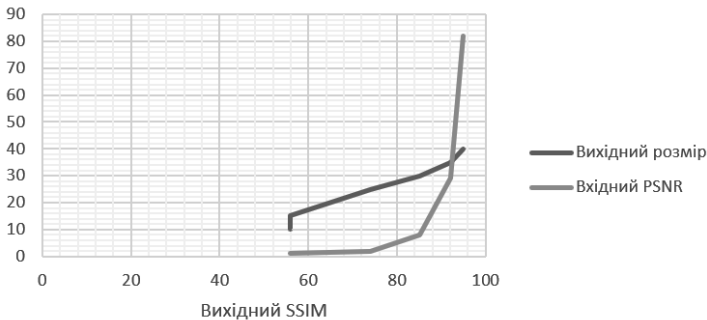


Рис. 4.1.3.4.9. Залежність вхідної якості PSNR [86] та вихідного розміру kb від ξ_{SSIM} (*Mean Structural Similarity Index Method*).

Нажаль підрахунок ξ_{SSIM} вимагає достатньо багато часу, а тому під час визначення y_0 слід обрати достатньо простий інструмент, що дасть найкращу початкову точку з найменшими зусиллями. На підставі цього пропоную прийняти $y_0 = (20 + 35) / 2 \approx 27$. При цьому для обох випадків вхідного зображення 800×600 та 1200×800 для вхідного $PSNR = 27$ вихідний $\xi_{SSIM} \approx 80$.

4.1.4 Розрахунок технологічних обмежень.

Перш за все проведемо необхідні розрахунки, щодо пропускної здатності системи та визначимо її можливості. Мережі типу LoRa мають наступні обмеження [57, 87, 98]:

1. Кожне повідомлення не може бути більшим за 222 байти та меншим за 51 байт.
2. Шлюз може отримувати повідомлення за 8 каналами а відправляти тільки за одним.
3. Рекомендований час в повітрі для кожного кінцевого пристрою 1%. Це забезпечую підєднання 1000 кінцевих пристроїв.

Очевидно що для передачі графічних зображень заданого часу не вистачить, але це можливо обійти додатковими шлюзами. Таким чином, кожний кінцевий пристрій надсилатиме наступне повідомлення до іншого шлюзу і все ще матиме показники наближені до оптимальних. Середній час на доставку одного повідомлення [87] згідно з експериментальними даними складає 0.215 мс.

Таблиця 4.1.4.1

Результати розрахунку для розміру 1200x800

Якість	Очікуваний розмір	Кількість пакетів	Час на доставку (сек)	Повідомлень в день (30 сек в мережі)
29	5490	26	6	5
30	8282	40	9	3
31	11761	57	12	2-3
32	15992	77	16	2-3
33	21038	101	21	1
34	26959	130	27	1
35	33820	163	34	1

Таким чином очевидно, що для випадку коли кількість кінцевих пристроїв на 1 шлюз нижча за 1000, існує можливість передачі даних за деяким інтервалом.

Результати розрахунків також продемонстрували, що використання граничних мінімумів вхідної якості (PSNR – Point-to-signal-noise-ratio) при трансформації зображення у формат JPEG2000 дозволяє виконувати послідовну доставку контенту від 2 до 5 разів на добу з одного пристрою без виходу за межі 30 секундної операційної межі, з вихідним $\xi_{SSIM} > 80$.

Висновки до розділу 4.1

1. В рамках роботи було досліджено можливості створення інформаційної технології (програмно-апаратного рішення), що має на меті передачу цифрових даних від кінцевого пристрою до користувача у мережах типу LPWAN.

2. Були з'ясовані мінімально необхідні компоненти системи та їх призначення. Визначені шляхи обміну даними, у тому числі протоколи та алгоритми згідно з постановкою завдання.

3. В даній роботі були досліджені алгоритми передачі даних у мережах IoT та висунуті гіпотези, щодо їх інтерпретації та обробки.

4. Були досліджені та класифіковані типові проблеми передачі даних та пониження QoS. Особливу увагу було приділено роботі з втратами під час передачі даних.

5. Було запропоновано модель обміну графічною інформацією у мережах LoRaWAN, що може надсилати повідомлення на регулярній основі.

6. Було досліджено шляхи оптимізації передачі даних і запропоновано алгоритм регуляції QoS, що дозволяє тримати об'єм даних, що передається на граничних мінімумах.

7. Доцільні напрямки практичного використання рішення:

Спостереження у аграрному секторі, спостереження за станами віддалених об'єктів у лісі, фермерство. Можливе застосування у медичному секторі для віддаленого нагляду за пацієнтами.

8. Інформаційна технологія дозволяє послідовно надсилати графічні повідомлення з кінцевого пристрою від 2 до 5 разів на добу, та гнучко корегувати якість отриманого зображення.

9. Напрямки подальших досліджень.

Ключовим напрямом розширення можливості даного рішення є подальша оптимізація об'єму даних, що передається за рахунок додавання обробки втрачених пакетів.

4.2 Архітектура додатку аналізу медичних динамічних даних інтернету всього на основі мікросервісної архітектури

4.2.1 Вступ

Актуальність роботи обумовлено практичною потребою розгортання нових сервісів у хмарному середовищі.

Надаючи спрощене середовище програмування, бессерверні обчислення значно спрощують використання хмарних ресурсів, тим самим залучаючи більше людей, які можуть і будуть його використовувати. Бессерверні обчислення включають в себе пропозиції FaaS і BaaS і знаменують собою важливий етап розвитку хмарного програмування. Це позбавляє від необхідності ручного управління ресурсами і їх оптимізації, до яких сьогоднішні серверні обчислення зобов'язують розробників додатків, що походить на перехід від мови асемблера до мов високого рівня понад чотири десятиліття тому.

Amazon Web Services, що є лідером в сфері надання веб послуг прогнозує, що використання безсерверного підходу буде стрімко рости. Також прогнозується, що локальні гібридні хмарні додатки з часом будуть скорочуватися, хоча деякі розгортання можуть зберігатися через нормативні обмеження і правила управління даними.

Бессерверні обчислення стануть стандартною обчислювальною парадигмою в епоху хмарних рішень, в значній мірі, витіснивши серверні обчислення і тим самим заклавши ґрунті клієнт-сервер.

Також використання капілярної архітектури доставки мікросервісів не є оптимальним рішенням з точки зору утилізації системних ресурсів. В більшості випадків конкретному користувачу немає необхідності мати доступ до усієї функціональності певного сервісу, проте ресурси вже є виділеними. Для того щоб покращити використання даних ресурсів слід змінити методологію розробки програмного забезпечення перейшовши на функціональний підхід, тим самим дозволяючи використовувати великий набір легких функцій замість громоздкого мікросерверу в порівнянні з ними.

Мета і задачі дослідження

Об'єкт дослідження: інформаційна технологія, що дозволяє будувати системи будь якої складності, та будь-якого типу, розмежовуючи набір функціональності який має у доступі визначений користувач та використовуючи методологію розділення апаратних ресурсів.

Предмет дослідження: створення системи, що дозволить реконфігуруватися в залежності від вимог користувача з можливістю одночасного обслуговування великої кількості користувачів.

Мета роботи: покращення гнучкості розгортання функціональності для користувача, та оптимальна утилізація апаратних ресурсів обчислювальних засобів використовуючи методи бессерверної та «Fog and Edge» архітектур.

Наукова новизна отриманих результатів

Вдосконалено методи вирішення типових проблем розгортання інфраструктури, що повинна забезпечувати потреби користувачів. Запропоновано рішення для розподілу функціональності на апаратні ресурси.

Була вдосконалена модель архітектури інформаційної системи гібридного типу розгортання, що сама себе конфігурує в залежності від потреб кінцевих користувачів.

Практичне значення одержаних результатів. Представлені методи та інформаційні технології дозволяють відмовитись від використання сервісного підходу замінивши його на використання функціонального підходу у побудові мережевих додатків. Запропоноване рішення покращує використання ресурсів, що знаходяться в контурі певної системи, дозволяючи обслуговувати більшу кількість користувачів за один і той же час, дозволяючи розгортати декілька наборів функціональності на одному і тому ж апаратному ресурсі.

4.2.2 Огляд базових архітектур

Мікросервісна архітектура - варіант сервіс-орієнтованої архітектури програмного забезпечення, орієнтований на взаємодію, наскільки це можливо, невеликих, слабо пов'язаних і легко змінюваних модулів - мікросервісів, що набув поширення в середині 2010-х років у зв'язку з розвитком практик гнучкої розробки і DevOps [78] (Рис. 4.2.2.1).

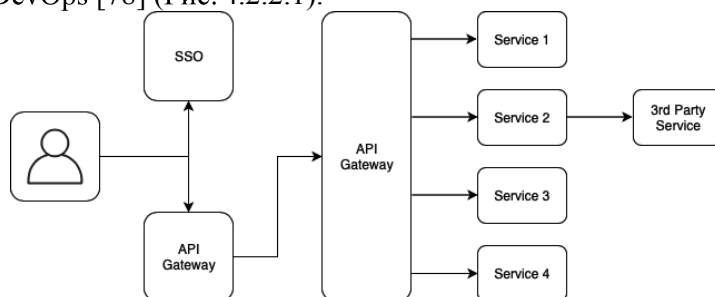


Рис. 4.2.2.1. Приклад мікросервісної архітектури.

Протягом останніх кількох років концепція DevOps з'явилася з обіцянкою радикальних поліпшень у життєвому циклі програми через розробку надійних, багаторазових програмних компонентів під назвою «Мікросервіси». Концепція мікросервісного підходу забезпечує архітектуру, яка спирається в основному на легкі контейнерні технології для віртуалізації. Така конструкція дає можливість розробляти розподілені обчислювальні програмні системи, які здатні досягати високих показників QoS [78].

Якщо в традиційних варіантах сервіс-орієнтованої архітектури модулі можуть бути самі по собі досить складними програмними системами, а взаємодія між ними часто покладається на стандартизовані великовагові протоколи (такі, як SOAP, XML-RPC), в мікросервісній архітектурі системи створюються з компонентів, що виконують дещо елементарні функції, і

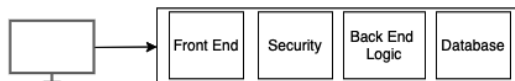
взаємодіючі з використанням економічних мережевих комунікаційних протоколів (в стилі REST з використанням, наприклад, JSON, Protocol Buffers, Thrift). За рахунок підвищення гранулярності модулів архітектура націлена на зменшення ступеня зачеплення і збільшення зв'язності, що дозволяє простіше додавати і змінювати функції в системі в будь-який час.

Найбільш популярне середовище для виконання мікросервісів - системи управління контейнеризованими додатками такі як Kubernetes і її надбудови OpenShift і CloudFoundry, Docker Swarm, Apache Mesos, в цьому випадку кожен з мікросервісів, як правило, ізолюється в окремий контейнер або невелику групу контейнерів, доступну через мережу іншим мікросервісам і зовнішнім споживачам, і управляється середовищем оркестрації, що забезпечує відмовостійкість і балансування навантаження. Типовою практикою є включення в контур середовища виконання системи безперервної інтеграції, що забезпечує автоматизацію поновлення і розгортання мікросервісів [78].

Бессерверна архітектура. Бессерверні обчислення [109] - природна для хмарних рішень архітектура, яка дозволяє передати більшу частину операційної відповідальності провайдеру бессерверних рішень і тим самим отримати більше гнучкості та полегшити процес розробки кінцевого додатку (Рис. 4.2.2.2). Бессерверні обчислення дозволяють створювати і запускати додатки і сервіси, не турбуючись про конфігурацію серверів. Вони усувають необхідність займатися питаннями управління інфраструктурою - такими, наприклад, як виділення серверів або кластерів, необхідних ресурсів, а також встановлення оновлень і обслуговування операційної системи. Їх можна використовувати практично для будь-якого типу додатків або сервісів серверної частини, при цьому все, що потрібно для запуску і масштабування додатка виконується без втручання клієнта.

У наш час існує безліч способів створення програми. Рішення, які приймаються на ранньому етапі, можуть і будуть впливати не тільки на життєвий цикл програми, але і на команди розробників і, в кінцевому рахунку, на компанію або організацію.

Традиційна архітектура



Бессерверна архітектура

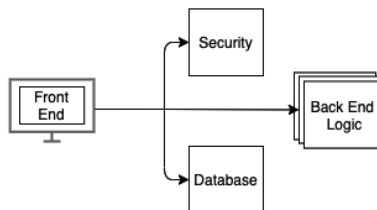


Рис. 4.2.2.2. Порівняння традиційної і бессерверної архітектури.

Fog And Edge архітектура. Для досягнення ефективної взаємодії з кінцевими пристроями, може не вистачати пропускної здатності мережі, або інформація вимагає якоїсь обробки. Один із гірших випадків відсутність точного місцезнаходження або постійна зміна локації кінцевих пристроїв що мають потребу у взаємодії [112]. Важливо зауважити, особливість Docker контейнерів, а саме що вони зберігаються як файл у реєстрі контейнерів (артефактів). Таким чином якщо ми побудуємо деякий проміжний рівень «Туман», що буде завантажувати контейнери з реєстру та зберігати їх ближче до кінцевих пристроїв. Тоді ми зможемо завантажувати на кінцеві пристрої на базі Raspberry PI необхідні сервіси і скеровувати трафік з Хмари на них. Таким чином деякі обчислення, фільтри та інший аналіз динамічних даних Інтернету Всього може бути скеровано з хмари ближче до споживача [112].

Очікувано кількість запитів від кінцевих пристроїв «Інтернету речей» продовжує невинно зростати і досягати мільярдів запитів в секунду. Централізовані сервери (включно з хмарними) не можуть впоратись з цим навантаженням. Суттєва частина користувачів не хоче передавати свої дані до хмарних накопичувачів. З появою Інтернету речей (IoT) розробляються різні алгоритми штучного інтелекту різних обчислювальних складностей, які працюють на безперервно породжених потоках даних, що прямують з датчиків. Багато програм IoT, наприклад, для автомобілів, що самостійно керуються, мають критичний до часу характер і потребують вирішення специфічних вимог щодо якості обслуговування (QoS), включаючи низьку затримку зв'язку, швидкі обчислення та високу пропускну здатність. Для того, щоб задовольнити такі вимоги, хмарна централізована обчислювальна парадигма розвивалася у напрямку нових перспективних розподілених Edge і Fog обчислювальних тенденцій [112] (Рис. 4.2.2.3).

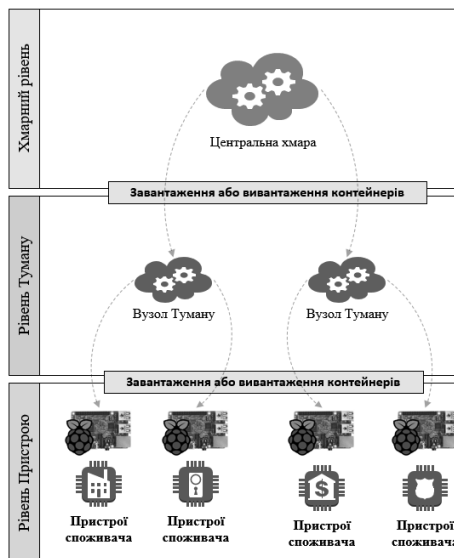


Рис. 4.2.2.3. Капілярна архітектура.

Edge node - обчислювальний пристрій, який має процесорну або багатопроцесорну бортову систему. Прикладами є акумуляторні машини, роботи, смартфони, Raspberry Pi або Arduino. У цьому контексті вузли туману можна розуміти як інфраструктура хмарних обчислень, яка існує в тісній географічній близькості від кінцевих вузлів. З цією метою різні легкі операційні системи, такі як RancherOS або CoreOS, дозволяють перетворити навіть невеликі пристрої, такі як маршрутизатори, на постачальників Cloud, в цьому випадку вони вважаються обчислювальною інфраструктурою Fog. Обчислювальна модель Edge / Fog - це нова парадигма обчислень, яка має на меті оптимізувати розумні додатки для розширення їх функціональних можливостей, близьких до географічного розташування пристроїв IoT, а не для далеких центрів обробки даних. У таблиці 4.2.2.1 представлено порівняння віртуалізації на базі контейнерів та віртуалізації на основі віртуальної машини.

Таблиця 4.2.2.1.

Порівняння технологій віртуалізації і контейнеризації

Характеристика	Контейнери	Віртуальні машини
Вимоги	Рушій контейнерів Наприклад, Docker	Гіпервізор, наприклад Xvisor
Вага	Легковісні	Важкі
Розмір	Невеликі	Великі
Час завантаження	Швидкі	Повільні

Технології віртуалізації на базі гіпервізора підтримують автономні віртуальні машини, які є незалежними та ізольованими від хост-машини. Кожен екземпляр віртуальної машини має свою власну операційну систему і набір бібліотек, і працює в середовищі емуляції гіпервізора. Це робить керовані віртуальні машини важкими. Для реалізації даної концепції використовуються контейнерні технології, що вважаються більш легкими. У порівнянні з віртуальними машинами, використання контейнерів не вимагає операційної системи для завантаження.

Віртуальні машини використовують багато ресурсів і як правило їх неможливо легко розгорнути на невеликих серверах або пристроях з обмежених ресурсами, таких як Raspberry Pi, або маршрутизатори. Контейнери ж можна використовувати гнучко і в таких рамках, оскільки їхня природа передбачає використання ресурсів основної ОС. Розгортання служб на базі контейнерів під час виконання може бути досягнуто швидше, ніж у віртуальних машинах. Завдяки цим перевагам різні контейнерні платформи віртуалізації, такі як Google Container Engine [111] і Amazon EC2 Container Service, стали альтернативами віртуалізації на основі гіпервізора.

4.2.3 Огляд та аналіз існуючих рішень в сфері безсерверних обчислень

AWS Lambda

Amazon Web Services (AWS) [103] - назва, що приходить на розум першою при думці про хмарні обчислення. Amazon є найбільшим провайдером в хмарному просторі і має більш широкий набір допоміжних інструментів і ресурсів, ніж будь-який з конкурентів. AWS Lambda дозволяє запускати програмний код без виділення серверів і керування ними. Оплаті підлягає тільки фактичний час виконання обчислень. За допомогою Lambda можна запускати практично будь-які види додатків і серверних сервісів, при цьому не потрібні будь-які операції адміністрування. Просто завантажте програмний код, і Lambda забезпечить всі ресурси, необхідні для його виконання, масштабування і забезпечення високої доступності. Можна налаштувати автоматичний запуск програмного коду з інших сервісів AWS або викликати його безпосередньо з будь-якого мобільного або інтернет-додатку [103]. Lambda підтримує функції, написані на JavaScript (і мови, які в його компілюють) за допомогою Node.js, Python, Java (підтримується Java 8) і сімейства Microsoft .NET (C #, Visual Basic і F #) за допомогою підтримки .NET Core [103]. Документація є всеосяжною для Lambda, чого і слід було очікувати від самого передового продукту в даний час. Документація оновлюється, відображаючи нові функції, і публікується онлайн. Має дуже велику і активну спільноту, це сприяє набагато більшими можливостями для росту протягом декількох років, з тих пір, як Lambda вийшла на сцену.

В даний час Lambda має безкоштовний тариф, який дозволяє отримувати мільйон запитів і 400 терабайт-секунд часу обчислень в місяць. Цього достатньо, щоб виконати досить багато запусків, перш ніж буде надіслано будь-які рахунки. Запити вище цього порогового значення виставляються за ціною \$ 0,00001667 / ГБ [103].

Microsoft Azure Functions

Функції Azure - це бессерверна служба обчислень, яка дозволяє виконувати активований подіями код без необхідності явно готувати або адмініструвати інфраструктуру. Функції Azure дозволяють запускати невеликі фрагменти коду (звані функціями), не турбуючись про порушення інфраструктури додатку. За допомогою функцій Azure хмарна інфраструктура реалізовує всі найновіші серверні можливості, необхідні для того, щоб додаток мав змогу масштабуватися. Функція активує подію певного типу. Підтримувані тригери включають в себе реагування на зміну даних, повідомлення, результат HTTP-запиту і виконання за розкладом.

Платформа Azure від Microsoft швидко розширює свою функціональність в останні роки, так як вона конкурує з AWS за частку на ринку. Список підтримуваних ресурсів в значній мірі аналогічний тому, що пропонує AWS, але Azure також надає досить багато додаткових функцій, характерних для аудиторії .NET і TypeScript. Картина тут трохи відрізняється від AWS. Мови -

JavaScript (і мови, які в нього компілюються) з Node.js, C#, F#, Python, PHP, Bash, Batch і PowerShell.

Microsoft бачить перспективи в розширенні бессерверного хостингу в центрах обробки даних і передбачає велику аудиторію серед розробників. Microsoft документує всі свої продукти - це найдосвідченіша компанія, коли справа доходить до взаємодії зі своїм співтовариством розробників. Microsoft поставив ціну на Azure Functions, рівну Amazon і відповідну безкоштовному тарифу Lambda. Коли трафік додається до вартості, оцінки витрат фактично є найнижчими серед великих провайдерів при одному і тому ж робочому навантаженні.

Google Cloud Functions

Хмарні функції [111] - безсерверна технологія Google, і хоча з точки зору паритету характеристик вона в основному схожа з Azure і AWS, але у неї є кілька помітних відмінностей. Google Cloud Functions - це середовище виконання без сервера для створення та підключення хмарних служб. За допомогою хмарних функцій можна писати прості, одноцільові функції, які додаються до подій, що викликаються із хмарної інфраструктури та сервісів. Хмарна функція спрацьовує при запуску події, яку викликають. Код виконується в повністю керованому середовищі. Не потрібно надавати жодної інфраструктури і турбуватися про управління будь-якими серверами.

В даний час Google Cloud Functions підтримує тільки JavaScript, але очікується, що Google Cloud Functions буде розширюватися в найближчому майбутньому [111]. Також варто відзначити, що інші мови, які можуть бути скомпільовані в JavaScript (наприклад, TypeScript, Coffeescript), також є цілком допустимими при розробці програмного забезпечення для Google Cloud Functions. Документація є всеосяжною і її легко використовувати. Спільнота відстає від Amazon і Microsoft хоча б тому, що вони останніми вийшли на ринок безсерверних функцій, також вони не мають великої аудиторії розробників як у Microsoft. Спільнота значно зросте, але Cloud Functions - все ще досить молодий продукт, тому потрібен час, щоб знайти аудиторію.

4.2.4 Огляд запропонованого рішення

4.2.4.1 Обране обладнання для реалізації та обґрунтування

Для реалізації проекту було обрано мікрокомп'ютер від компанії element14 - Raspberry Pi 4 [105].

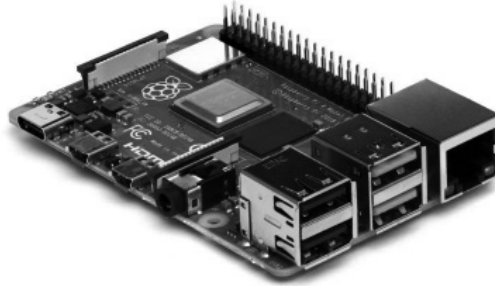


Рис.4.2.4.1.1. Raspberry Pi 4.

Таблиця 4.2.4.1.1

Характеристики Raspberry Pi 4.

Характеристики RPi3	
Чіп	Broadcom BCM2711
Процесор	4 ядерний Cortex-A72 (ARM v8) 64-bit
Тактова частота	1.5 GHz
Графічний процесор	VideoCore VI
ОЗП	4096 MB
Пам'ять	microSD карта

Таблиця 4.2.4.1.2

Інтерфейси вводу/виводу Raspberry Pi 4.

Інтерфейси вводу-виводу	
USB 2.0	2
USB 3.0	2
Micro HDMI	2
3,5mm Jack	1
GPIO	40
Ethernet	1
Wifi	802.11b/g/n
Bluetooth	4.1

Також на борту доступні порти для підключення дисплея DSI та камери CSI. Для підключення периферії буде використано GPIO входи-виходи. RPi3 підтримує велику кількість можливих ОС написаних на ядрі Unix, такі як: Raspbian, RISK OS, OSMC, Arch Linux, які можна завантажити і повністю безкоштовно встановити на пристрій.

Ціна RPi є оптимальною для використання в даному проекті і становить 50\$ та набір системних компонентів RPi3 повністю задовільняє вимоги для реалізації проекту, а саме:

- Компактний розмір;
- Низьке енергоспоживання завдяки ARM архітектурі;
- Операційна система з можливістю зібрати Node.JS.

Також до переваг даного пристрою необхідно віднести відсутність аналогів в даній ціновій категорії з подібним набором характеристик.

Для реалізації проекту було обрано такі інтерфейси передачі даних:

WiFi - загальноновживана назва для стандарту IEEE 802.11 передачі цифрових потоків даних по радіоканалах. Поточні реалізації Wi-Fi дозволяють отримати швидкість передачі даних понад 100 Мбіт/с, при цьому користувачі можуть переміщуватися між точками доступу на території покриття мережі Wi-Fi.

RJ-45 — фізичний інтерфейс, який загалом використовується для з'єднання комп'ютерних мереж за допомогою звітої пари через мережевий комутатор, або при створенні мережі з двох комп'ютерів до один одного через мережеву карту.

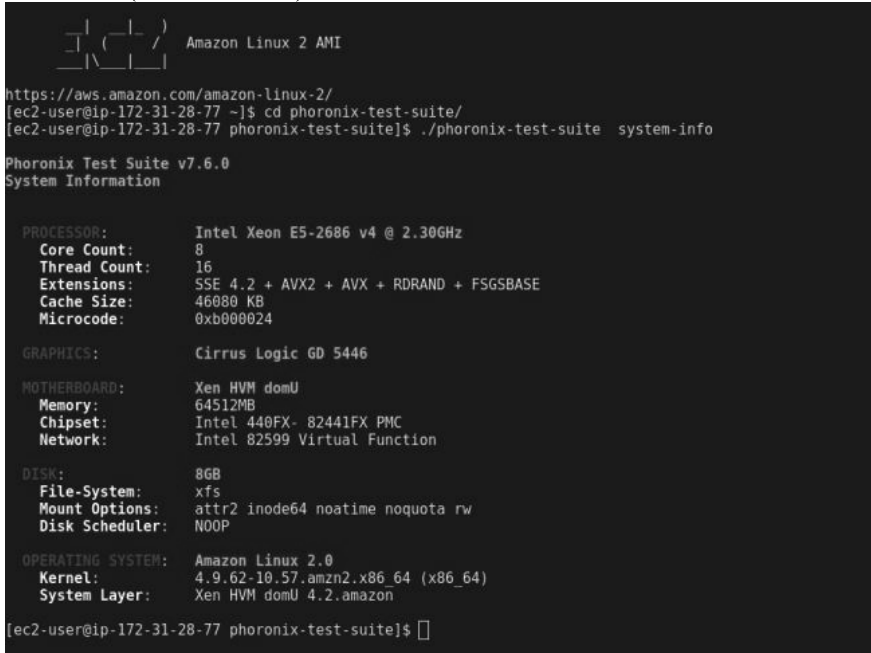
4.2.4.2 Програмний набір технологій та обґрунтування

Для реалізації проекту було обрано наступний набір технологій:

Raspberry Pi OS [105] - офіційна підтримувана операційна система Фонду. Встановлюється за допомогою NOOBS, або записом образу диска на карту пам'яті. Raspberry Pi OS поставляється заздалегідь встановленою з великою кількістю програмного забезпечення для освіти, програмування та загального використання. У ньому є Python, Scratch, Sonic Pi, Java та багато іншого. Raspberry Pi OS - це офіційна операційна система для Raspberry Pi, вона розроблена спеціально для цього пристрою і має все необхідне програмне забезпечення. Raspbian заснована на ARM версії Debian 8 Jessie і містить такі програми за замовчуванням - офісний пакет LibreOffice, веб-браузер, поштовий клієнт - Claws Mail, легке оточення робочого столу, а також інструменти для програмування.

ОС Amazon Linux AMI [103] - це дистрибутив Linux, що надається Amazon Web Services для використання в Amazon Elastic Compute Cloud (Amazon EC2). Він призначений для забезпечення стабільного, безпечного і високопродуктивного середовища виконання програм, які потребують в Amazon EC2. Образ підтримує можливості нових типів інстансів EC2 і містить пакети для простої інтеграції з AWS. AWS регулярно випускає оновлення безпеки і робочі виправлення для всіх інстансів, на яких використовується AMI Amazon Linux. AMI Amazon Linux надається користувачам Amazon EC2

безкоштовно. Amazon Linux AMI відрізняється широким спектром можливостей. У поточну стабільну версію включено понад п'ятдесят дев'ять тисяч пакетів програм для десяти архітектур на основі ядра Linux. Також до Amazon Linux AMI включена система управління пакетами Yellowdog Updater, Modified, жорстка політика по відношенню до пакетів, репозиторії з величезним їх кількістю, а також високу якість випущених версій. Це уможливило просте оновлення між версіями, а також автоматичну установку і видалення пакетів (Рис.4.2.4.2.1).



```

┌───┐ ┌───┐ ┌───┐
├───┘ └───┘ └───┘
└───┘ └───┘ └───┘
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-28-77 ~]$ cd phoronix-test-suite/
[ec2-user@ip-172-31-28-77 phoronix-test-suite]$ ./phoronix-test-suite system-info

Phoronix Test Suite v7.6.0
System Information

PROCESSOR: Intel Xeon E5-2686 v4 @ 2.30GHz
Core Count: 8
Thread Count: 16
Extensions: SSE 4.2 + AVX2 + AVX + RDRAND + FSGSBASE
Cache Size: 46080 KB
Microcode: 0xb000024

GRAPHICS: Cirrus Logic GD 5446

MOTHERBOARD: Xen HVM domU
Memory: 64512MB
Chipset: Intel 440FX- 82441FX PMC
Network: Intel 82599 Virtual Function

DISK: 8GB
File-System: xfs
Mount Options: attr2 inode64 noatime noquota rw
Disk Scheduler: NOOP

OPERATING SYSTEM: Amazon Linux 2.0
Kernel: 4.9.62-10.57.amzn2.x86_64 (x86_64)
System Layer: Xen HVM domU 4.2.amazon

[ec2-user@ip-172-31-28-77 phoronix-test-suite]$ █

```

Рис.4.2.4.2.1. Термінал Amazon Linux AMI

Середовище виконання Node.js [97] — платформа з відкритим сирцевим кодом для написання високопродуктивних мережевих застосунків, написаних мовою JavaScript або TypeScript. Засновником платформи є Райан Дал (Ryan Dahl). Якщо раніше JavaScript застосовувався для обробки даних в браузері на стороні користувача, то node.js надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їх виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників.

Для забезпечення обробки великої кількості паралельних запитів у Node.js використовується асинхронна модель запуску коду, заснована на обробці подій в неблокуючому режимі та визначенні обробників зворотніх викликів. Як способи мультиплексування з'єднань підтримується epoll, kqueue, /dev/poll і select. Для мультиплексування з'єднань використовується libuv, для створення пулу потоків (thread pool) задіяна бібліотека libeio, для виконання DNS-запитів у неблокуючому режимі інтегрований c-ares [97].

Оскільки Node.JS є монопоточним та використовує принципи асинхронного програмування, то необхідно розуміти як з ним працювати коректно. У синхронному коді кожна операція чекає закінчення попередньої. Тому вся програма може «зависнути», якщо одна з команд виконується дуже довго. Асинхронний код прибирає операцію, яка блокує основний потік програми, так що основний потік не блокується, і програма може виконувати інші операції.

Всі важкі операції переносяться в сторонні процеси-воркери. За допомогою бібліотеки `node-gyp` можливо писати власні модулі які будуть виконуватися у окремому потоці, тим самим не навантажуючи основний потік. Для цього необхідно дотримуватися зазначеної структури модуля у якому будуть прописані всі методи життєвого циклу додатку.

Підчас дослідження кінцевої плаформи розгортання сервісів було встановлено наступні технології/програмні каркаси як:

- Node.JS + Sequelize ORM + `express.js`;
- Java (Spring) + Hibernate ORM + Tomcat;
- C# + Entity Framework + IIS.

Оскільки для розробки застосунків реального часу краще всього підходить подійно-орієнтований підхід, що дозволяє тримати високі навантаження (від 10 тис. запитів), то мови програмування повинні мати в собі механізми асинхронного виконання коду.

У випадку стеку побудованому на основі Node.JS такий механізм забезпечує комбінація бібліотеки `libuv` та рушій V8 від Google [97].

У випадку стеку побудованому на основі C# такий механізм вбудований в можливості мови починаючи з версії 5.0.

У випадку стеку побудованому на основі Java такий механізм забезпечується фреймворком `DataKernel`.

Перевагу надано стеку з використанням Node.JS, оскільки:

1) Node.JS потребує набагато менше системних ресурсів, ніж представлені альтернативи, що є вагомим аргументом переваги при розгортанні на пристроях з обмеженими ресурсами, таких як Raspberry Pi 4;

2) Мова JS дозволяє найбільш ефективно реалізувати робочий прототип на відміну від інших;

3) Node.JS спроектовано для асинхронного підходу програмування;

4) NPM - менеджер застосунків для Node.JS має найбільшу спільноту серед інших менеджерів;

5) За умови успішної реалізації прототипу, використовуючи можливості сучасних інтегрованих середовищ розробки, проект можливо перевести на TypeScript - мову програмування рівня підприємства в короткий термін.

До того ж асинхронне програмування успішно вирішує безліч завдань. Одне з найважливіших — користувач може взаємодіяти з програмою поки виконується інше завдання. Ця модель дуже добре підходить для розробки додатків на подійно-орієнтованій архітектурі (Рис.4.2.4.2.2).

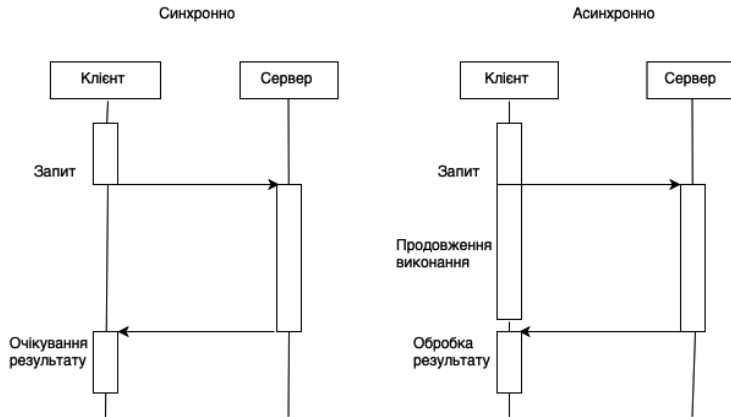


Рис.4.2.4.2.2. Порівняння синхронного та асинхронного виконання.

AWS EC2 Amazon Elastic Compute Cloud (Amazon EC2) [103] - це веб-сервіс, що надає безпечні масштабовані обчислювальні ресурси в хмарі. Він спрощує проведення обчислень в хмарі в масштабі всього Інтернету. Веб-інтерфейс сервісу Amazon EC2 дозволяє отримати доступ до обчислювальних ресурсів і налаштувати їх з мінімальними зусиллями. Він надає користувачам повний контроль над обчислювальними ресурсами, а також перевірене обчислювальне середовище Amazon для роботи. Amazon EC2 дозволяє скоротити час, необхідний для створення і завантаження нових інстансів сервера, до декількох хвилин, і забезпечує можливість швидко масштабуватися в будь-якому напрямку з урахуванням мінливих вимог до обчислювальних ресурсів. Amazon EC2 змінює економічну складову процесу обчислень, надаючи можливість платити тільки за використовувані ресурси. Amazon EC2 дозволяє розробникам уникати поширених хибних сценаріїв і створювати відмовостійкі додатки. Під час вибору провайдера послуг хмарної обчислювальної потужності ми надали перевагу саме AWS EC2 у зв'язку з наявністю великої експертизи саме з цим провайдером.

React [107] — відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків. Розробляється Facebook, Instagram і спільнотою індивідуальних розробників. React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React обробляє тільки користувацький інтерфейс у застосунках. Це відповідає видові у шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Серед SPA фреймворків ми розглядали такі варіанти: Angular, Ember.js, Vue.js, проте надали перевагу саме React оскільки він пропагує функціональний підхід, що по ідеології є близьким

до того як ми розробляли всю іншу частину системи. Також його можливості можна розширити використовуючи React Native — фреймворк для компілювання JavaScript коду в нативний код мобільних платформ [107].

4.2.4.3. Інфраструктура системи

Розроблена система складається з двох сервісів: «Ядро системи» та «Обслуговувач користувача». Ядро системи повинно бути розміщене на потужностях, що мають постійне підключення до мережі інтернет, та стабільну IP адресу. В той же час «обслуговувач» може бути розміщений на всіх платформах, що підтримують технологію Node.JS (виділений сервер, Raspberry Pi, віртуальна машина, контейнер).

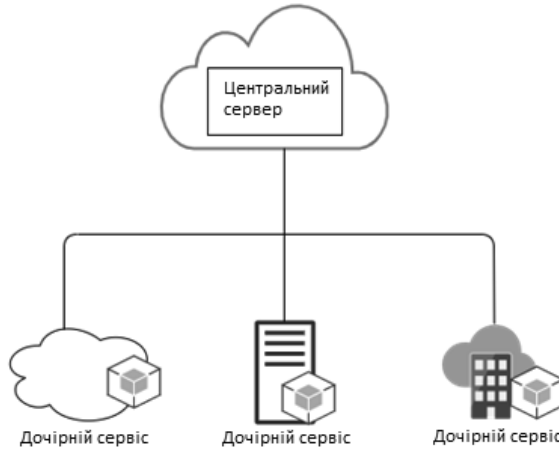


Рис.4.2.4.3.1. Високорівнева архітектура системи.

На відміну від архітектури «Fog and Edge» бізнес логіка, що необхідна користувачу не запускається в окремому контейнері. Замість цього вся бізнес логіка переноситься в «обслуговувач користувача» разом з конфігурацією яка необхідна користувачу. Для того, щоб зменшити вірогідність втручання в безпеку системи, слід обмежити технології, що можуть бути використані (драйвери бази даних, драйвери систем нотифікації, нативні розширення систем криптографії, тощо) до певного списку [100]. Для формування даного списку, можна використати статистику пакетного менеджера `rpm`.

Як ми можемо побачити з табл. 4.2.4.3.1, використання безсерверних функцій найкращий спосіб утилізувати апаратні ресурси за умови легковажності конкретної операції. Для того щоб опрацьовувати великий об'єм інформації в одному пакеті, слід використовувати рішення, які є найближчими до виділеного сервера. Притому безсерверні функції дозволяють відокремити набір функціональності використовуючи кожну функцію як окрему операцію, в той же час, використовуючи мікросервісний підхід, доведеться завантажувати повноцінний сервіс. Відмова від Docker контейнерів допомагає економити ресурси системи, в той же час дозволяючи обслуговувати набагато більшу

кількість користувачів. Оскільки використовується безсерверний підхід, ізоляція контексту виконання виконується на рівні кожної функції, це означає, що сусідні користувачі не можуть вплинути на хід виконання функції один одного.

Таблиця 4.2.4.3.1

Порівняння технологій віртуалізації і контейнеризації

Характеристика	Безсерверні функції	Контейнери	Віртуальні машини
Вимоги	Node.JS	Рушій контейнерів напр. Docker	Гіпервізор напр. Xvisor
Вага	Надлегкі	Легковісні	Важкі
Розмір	Найменший	Невеликі	Великі
Час завантаження	Моментальний	Швидкі	Повільні
Сервісні одиниці	Мільйони	Десятки	Десятки
Швидкодія на CPU intensive задачах	Повільна	Висока	Висока
Здатність до часткового завантаження функціональності	Так	Ні	Ні

4.2.4.3.1 Сервіс «Ядро»



Рис.4.2.4.3.1.1. Високорівнева архітектура сервісу «Ядро».

Це головний сервіс системи який спілкується з сервісами «обслуговувачами», та взаємодіє з кінцевими користувачами, що завантажують свою бізнес логіку до системи. Критично важливо щоб вся бізнес логіка була написана в функціональному стилі. Для цього слід користуватися засобами Node.JS, та правильно зконфігурувати набір залежностей для власної системи. Сервіс підключений до бази даних, в якій він зберігає все бізнес логіку користувача. Для того щоб система мала змогу розподілити бізнес логіку між серверами необхідно в JWT токен авторизації дописати ID функцій які він може викликати. Цей процес можна організувати на основі ролівої системи, де кожній із ролей дозволено викликати певні функції. Тому немає сенсу завантажувати функції, які користувач навіть не зможе викликати на апаратний засіб.

Для комунікації між «ядром» та «обслуговувачем» ми обрали REST-протокол. REST - скорочення від Representational State Transfer, що можна перекласти як «передача репрезентативного стану». Це стиль проектування розподілених систем за допомогою обмежень. Центральною абстракцією в REST є ресурс. А головні обмеження виглядають так:

- Клієнт-серверна модель.
- Взаємодія без збереження стану.
- Логічний інтерфейс.

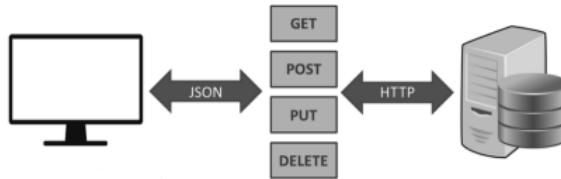


Рис.4.2.4.3.1.2. Відображення комунікації на основі REST протоколу.

Ресурс - це представлення віртуального об'єкту, реального об'єкту чи колекції об'єктів. Взагалі, ресурс може бути чим завгодно, розробник API вирішує що в нього буде ресурсом.

В REST кожен ресурс повинен бути унікальним. Тому ми присвоюємо їм ID. Для нашої системи /node/1 та /node/2 це два різних кінцевих вузла з ID 1 та 2. Також ресурси можуть бути вкладеними. Наприклад, URI клієнта 2 в першому кінцевому вузлі буде виглядати так /node/1/client/2.

4.2.4.3.2 Сервіс «Обслуговувач»

Цей сервіс використовується для того щоб надавати послуги користувачеві. Він тримає комунікацію з ядром системи та завантажує в свою оперативну пам'ять бізнес логіку користувача. За допомогою попередньої конфігурації він здатний виконувати функції з заданим контекстом.



Рис.4.2.4.3.2.1. Комунікація між користувачами та «обслуговувачами».

Для того щоб забезпечити максимальну продуктивність даного сервісу необхідно використовувати такі підходи як: багатопоточне програмування та асинхронна модель подій. За збереження функцій пам'яті відповідає сервіс контейнер функції. Він відповідальний за інвалідацію, завантаження, вивантаження, облік функцій в сервісі. Якщо на апаратному засобі більше ніж одне ядро центрального процесора, то відбувається розподіл навантаження на окремі потоки (Рис. 4.2.4.3.2.2). Для цього, коли піднімається процес даного сервісу він піднімає один потік в якості контрольного центру, а на всі інші вільні потоки процесора (вони можуть сприйматися як віртуальні ядра) піднімає по потоку контейнеру функцій. Кожен контейнер функцій працює на власному порті, щоб забезпечити можливість балансу навантаження зі сторони користувача.

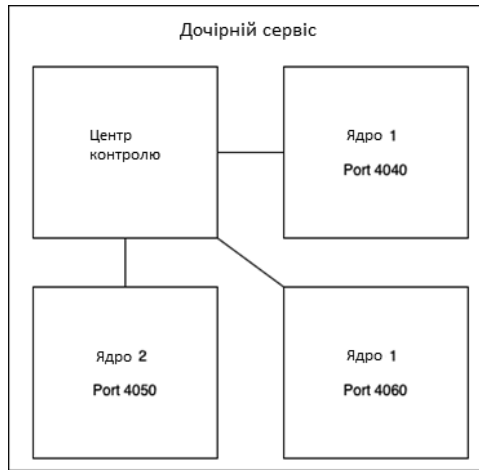


Рис. 4.2.4.3.2.2. Внутрішня будова сервісу «Обслуговувач».

У випадку якщо система володіє лише одним ядром, то в одному потоці обробляється відвантаження / завантаження функцій до контейнеру функцій та обслуговування клієнтів. Для зберігання інформації про підключення до різних сервісів рекомендовано використовувати змінні оточення. Для додавання змінних оточення користувач повинен вказати їх в спеціальному розділі користувацького інтерфейсу. Вони є доступними лише в контексті виконання функцій. Контекст виконання функцій необхідний для того, щоб обмежити використання функціями сторонніх бібліотек (для роботи з файловою системою, базами даних тощо). Для створення контексту необхідно перейти у відповідний розділ користувацького інтерфейсу. Для того щоб повернути дані з контексту їх необхідно завернути в об'єкт та присвоїти параметру `module.exports`. З контексту можна повертати як екземпляри третьосторонніх бібліотек так і певні константи. Для того, щоб створити функцію необхідно перейти у відповідний розділ користувацького інтерфейсу та задати такі параметри як: ім'я функції, шлях, тип запиту, контекст виконання та ввести тіло самої функції. Рекомендовано використовувати контексти виконання для того щоб обмежити

доступ до бібліотек для тих функцій де це не потрібно. Тим самим досягається принцип впровадження залежностей.

Для того щоб надати можливість користувачу використовувати сторонні бібліотеки рекомендується притримуватись такого ходу дій:

- Створити необхідні змінні оточення;
- Створити контексти виконання функцій у яких будуть використовуватися змінні оточення, створюватися екземпляри доступу до об'єктів та повертатися з контексту;
- Використати екземпляри з контексту в тілі функції.

Процес реєстрації «обслуговувача» в системі.

Для того щоб включитися в роботу системи, кожен сервіс обслуговувача повинен пройти наступну процедуру:

- Сервіс відправляє запит на реєстрацію у системі у якому він вказує інформацію про своє місце розташування (якщо є), IP- адресу;
- Центр керування отримує запит на реєстрацію від кінцевого вузла, генерує йому унікальний ідентифікатор та повідомляє його про те, що вузол пройшов реєстрацію відправляючи йому JWT-код у якому міститься тип вузла та його ідентифікаційний номер;
- Вузол записує цей ідентифікатор у пам'ять і з кожним наступним запитом відправляє його в ядро системи.

Процес обслуговування користувача в системі.

Для того щоб система могла обслуговувати користувача їй необхідно мати інформацію про даного користувача (список доступних функцій, тощо). Інформація береться з JWT токена, який користувач відправляє з кожним запитом до сервісу. Відносно даного токена узгоджується доступ до функцій для даного користувача та налаштовується роутінг для того щоб користувач міг користуватися даними функціями не вносячи зміни до власного API. В цілях безпеки рекомендується змінювати токен протягом певного періоду. Для цього можна використати техніку refresh token [100].

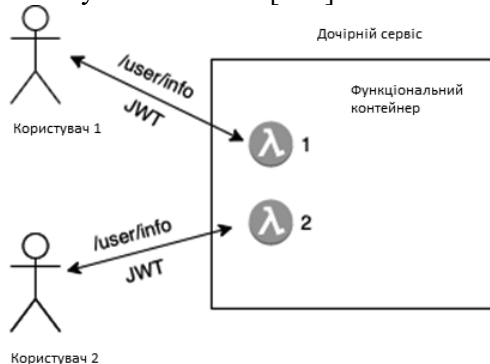


Рис. 4.2.4.3.2.3. Роутинг різних користувачів за одною адресою.

4.2.5 Користувачький інтерфейс

Список вузлів

ID	Node Name	Status	Type	IP	Detailed Info
1	kyiv_rpi	ONLINE	Raspberry Pi	89.128.138.48	Show
2	de_cloud	ONLINE	Cloud	189.12.183.4	Show
3	de_cloud_hz	ONLINE	Cloud	109.2.83.21	Show
4	gcp_ce	ONLINE	Cloud	19.122.83.41	Show
5	odesa_rpi	OFFLINE	Raspberry Pi	28.128.47.128	Show
6	it_aruba	ONLINE	Cloud	47.82.123.84	Show

Рис.4.2.5.1. Таблиця списку вузлів обслуговувача.

На цій сторінці можна переглянути загальні відомості з приводу підключених вузлів «обслуговувачів». Відомості доступні за такими полями даних як: ім'я, статус, тип, IP-адреса. Для того щоб провести сортування за значенням необхідно натиснути на назву колонки. При кліку на посилання в останній колонці відбувається перехід на сторінку з детальною інформацією щодо даного «обслуговувача».

Детальна інформація про вузел

Control Panel Nodes Lambdas Actions Log

kyiv_rpi node

Server Name: kyiv_rpi
 Type: Raspberry Pi
 IP: 80.124.32.199
 Lambdas list: auth_0, action_log,
 RPS: 1200
 Uptime: 2h

Рис.4.2.5.2. Детальна інформація про вузол.

На даній сторінці можна переглянути:

- Розташування обслуговувача на карті.
- Ім'я серверу.
- Тип серверу.
- IP адреса.
- Набір функцій які на нього завантаженні.
- Кількість запитів за секунду.
- Час активності.

Керування змінними оточення

ID	Variable Name	Value
1	PG_CONNECTION_STRING	postgresql://postgres@postgres:somehost:5432/test_database
2	RABBIT_MQ_HOST	localhost

Variable Name: VALUE:

Рис.4.2.5.3. Інтерфейс керування змінними оточення.

На цій сторінці можна додати, відредагувати та переглянути змінні оточення. Для того, щоб додати нову змінну оточення слід списати назву в поле для назви та значення даної змінної після чого натиснути на кнопку «зберегти». Редагування змінної відбувається після подвійного натискання на її значення. Для того, щоб відсортувати дані за значенням необхідно натиснути на назву колонки. Змінні потрапляють до контексту за принципом як є в вигляді стрічки. Тому якщо для конфігурації певної бібліотеки необхідно використовувати числове значення його потрібно перед тим задалегідь привести до типу який необхідний. Для кращої швидкодії також слід використовувати змінну `NODE_ENV` зі значенням «production», оскільки це загальноприйнятий підхід на який орієнтуються розробники різних бібліотек, драйверів тощо.

Створення контексту виконання

Control Panel | Lambdas | Nodes | Actions Log

Create Context

Context Name:

Context Body:

```
import pg from 'pg';

const connection = new pg.Client(process.env.PG_CONNECTION_STRING);

module.exports = { connection };
```

Рис.4.2.5.4. Інтерфейс створення контексту.

На даній сторінці можна створити контекст виконання функції. Для цього необхідно задати йому ім'я та саме тіло контекста. В тілі контекста можна проводити ініціалізацію модулів, виводити конфігурацію з вхідних даних, розраховувати значення констант на які будуть опиратися кінцеві функції. Для

виведення даних в функцію необхідно перевизначити атрибут `module.exports` новим об'єктом. В функції можна буде отримати доступ до контексту за допомогою вхідного параметру `ctx`. Для оптимізації швидкості виконання бізнес-логіки визначення контексту не повинно бути важким, оскільки даний контекст виконуються перед кожною функцію яку викликає користувач. Для оптимізації безпеки не слід повертати з контексту виконання змінні оточення та використовувати їх в тілі функцій [100]. Якщо виникає така необхідність - необхідно створити об'єкт, який буде виконувати певну бізнес-логіку та повернути його з контексту виконання.

ID	Lambda Name	Lambda Size	Is Loaded	Type	Route	Detailed Info
1	auth_8	2 KIB	TRUE	POST	/user/auth	Show
2	temperature_log	4 KIB	TRUE	GET	/weather/temperature	Show
3	send_stat	2 KIB	TRUE	PUT	/cc/stat	Show
4	cc_health	2 KIB	FALSE	GET	/cc/stat	Show
5	cookie_sync	3 KIB	FALSE	PUT	/user/cookie	Show
6	dsp_bidding	2 KIB	TRUE	POST	/bidder/bid	Show

Рис.4.2.5.5. Загальні відомості про функції.

Список функцій

На даній сторінці можна переглянути відомості щодо функцій зареєстрованих системи. Табличка на цій сторінці поводить себе аналогічно сторінці перегляду обслуговуючів. При кліку на посилання в останній колонці відбувається перехід на сторінку з детальною інформацією щодо даної функції.

Створення функції

Control Panel | Lambdas | Nodes | Actions Log

Create Lambda

Lambda Name

URL

Type

Context

Lambda Body

Submit

Рис.4.2.5.6. Інтерфейс створення функції.

На даній сторінці користувачу пропонується створити функцію. Для цього йому необхідно ввести ім'я функції, її шлях, тип запиту, обрати контекст виконання функції. В відповідне поле необхідно ввести тіло функції починаючи з перевизначення параметру `module.exports`.

На вхід функція приймає такі параметри:

`headers` - HTTP хедери користувача;

`params` - URL параметри, позначені двокрапкою (`/user/:id`);

`attributes` - URL - атрибути;

`body` - Тіло запиту для POST, PUT, DELETE, PATCH запитів;

`user` - дані про користувача, які взяті з його JWT токену;

`ctx` - контекст виконання функції;

Висновки до розділу 4.2

В рамках роботи було запропоновано та описано спосіб рознесення бізнес-логіки користувача на велику кількість кінцевих вузлів.

Було проведено дослідження, мета якого - встановлення оптимальної конфігурації для максимальної утилізації ресурсів обчислювального пристрою.

Було досліджено платформу Node.JS в порівнянні з альтернативними технологіями асинхронного програмування для побудови застосунків реального часу, що призначені для розгортання у гібридній сервіс-орієнтованій архітектурі безсерверних обчислень, та встановлено, що Node.JS - є оптимальним рішенням для побудови прототипу.

Було проведено дослідження, мета якого - встановлення оптимальної конфігурації для рознесення бізнес-логіки користувача в географічній близькості біля нього для покращення користувацького досвіду при використанні бізнес логіки. Запропоновано рішення для рознесення лише тої бізнес-логіки, яка необхідна користувачу в даний момент.

Спроектване рішення було порівняно з аналогами у сфері безсерверних обчислень та встановлено, що спроектване рішення дозволяє розташувати бізнес логіку в географічній близькості до кінцевого користувача.

За допомогою спроектваного рішення можливо обслуговувати багатьох користувачів різних бізнес-логік що посилаються на одні і ті ж адреси.

Запропоновано спосіб розділення користувачів використовуючи підхід багатопоточного програмування та асинхронної моделі подій. Запропоновано розподіляти навантаження посилаючи користувачів на різні порти.

4.3 Обробка даних щодо 3-D моделювання руху крові в серцево-судинній системі людини

В сучасному житті досить гостро стоїть проблема серцево-судинних захворювань – за статистикою, їм належить перше місце за причиною смерті. Саме з метою покращення цієї ситуації будуються моделі частин цієї системи. Ця робота присвячена моделюванню руху крові в судинах. Вирішення цієї задачі дозволить досліджувати широкий спектр проблемних ситуацій, пов'язаних з кров'яним тиском, наприклад таких, що призводять до розриву судин.

В якості математичної основи була обрана теорія, що є розвиненням теорії клітинних автоматів, і має назву Lattice-Boltzmann (решітчастий газ Больцмана). Вибір клітинних автоматів для моделювання руху крові по судинам пояснюється простотою їхньої формалізації та реалізації [1, 18, 65, 121]. Процеси динаміки в активних середовищах (реакційно-дифузійні процеси у фізиці, хімії, біології, екології) традиційно задаються складними диференціальними рівняннями (або системами рівнянь) із частинними похідними (наприклад, рівнянням Нав'є-Стокса). Рівняння цього типу вивчаються досить складно, тому що аналітичні розв'язки одержати неможливо через нелінійний член, а чисельні методи або погано розпаралелюються, або обмежені умовами стійкості й точності. Тим часом алгоритми для клітинних автоматів таких обмежень не мають. Вони досить добре розроблені і теоретично обґрунтовані. Програмна реалізація зазначених рівнянь дуже складна, потребує значних витрат часу й пам'яті, передбачає численні обмеження на параметри рівнянь. Як тільки ці обмеження задати не константами, а функціями, одразу перестає працювати весь математичний апарат. Клітинні ж автомати зручні тим, що адекватні правила переходів для них обираються тільки один раз, і форма судин уже не відіграє ролі, чого не можна зробити з диференціальними рівняннями.

Мета. Створити модель руху крові в судинах за допомогою решітчастого газу Больцмана задля вирішення проблемних ситуацій, пов'язаних з кров'яним тиском.

4.3.1. Використання клітинних автоматів та решітчастого газу Больцмана для моделювання потоків рідини

4.3.1.1. Клітинні автомати та решітчастий газ Больцмана

Класичний клітинний автомат

Клітинний автомат був запропонований фон-Нейманом у середині минулого століття. На основі цієї моделі підтверджувалася думка про те, що людина може створити пристрій, що володіє властивими йому самому властивостями, зокрема, здатністю відтворювати собі подібного. Тому клітинний автомат довгий час розглядався як модель самовідтворення і

тлумачився як спрощена модель деякого біологічного співтовариства, що складається з множини клітин. Кожній клітині ставиться у відповідність скінченний автомат, який називається елементарним автоматом, який може знаходитися в одному з двох станів: 0 і 1 (чорне або біле) і змінювати цей стан $0 \rightarrow 1$ або $1 \rightarrow 0$ в залежності від станів кліток деякого свого оточення, яке називається сусідством. Алгоритм обчислення наступного стану залежно від станів сусідів (функція переходів елементарного автомата) всіх кліток однаковий. Всі клітини виконують перехід в новий стан одночасно, тобто синхронно і паралельно. При цьому спостерігається зміна глобальної чорно-білої картини розподілу станів по простору клітинного автомата. Така картина називається конфігурацією клітинного автомата. У більш строгому вигляді основні властивості клітинного автомата формулюються в наступному вигляді.

- Дискретність простору, часу і станів.
- Однорідність (всі клітини організовані в регулярну просторову структуру).
- Синхронний режим зміни станів.
- Просторова локальність (аргументами функції переходів кожного елементарного автомата є стани елементарних автоматів з його обмеженої околиці).

Ітеративна зміна конфігурації при переходах всіх елементарних станів в нові стани називається еволюцією клітинного автомата. Еволюціонуючи клітинний автомат моделює просторову динаміку, яка може мати завершення, повторюватися періодично або змінюватися хаотично. Дослідження цієї "моделі світу" показали, що не дивлячись на простоту кожної клітини, їх кооперативна робота моделює дуже складні і різноманітні процеси, які іноді неможливо (або принаймні, невідомо як) описати іншим способом. У такому розумінні клітинний автомат тепер називається класичним клітинним автоматом.

Перший час клітинний автомат залишався об'єктом для гри розуму. Спочатку ця гра відбувалася на папері. Потім вона була перенесена на комп'ютер. Найвідомішою з таких "іграшок" дотепер є так звана "Гра Життя". Клітинний автомат, клітини якого розташовані в комірках двовимірної декартової решітки і мають тільки по вісім сусідів, може мати нескінченну кількість еволюцій, які відтворюють різні картини: рух кулі по якій-небудь траєкторії, розповсюдження хвилі, розкручування спіралі, тощо. Завдання в цій грі полягає в тому, щоб знайти таку початкову конфігурацію, при якій клітинний автомат, еволюціонував би бажаним чином. Всі спроби вирішити це завдання формально не увінчалися успіхом. Тільки досвід, навик і терпіння при випробуваннях безлічі спроб можуть допомогти в цій справі.

Коли почала бурхливо розвиватися мікроелектроніка (70-і роки минулого століття) і з'явилася потреба в швидкодіючих електронних схемах на багатьох елементарних автоматах, що діють паралельно, клітинний автомат здобув важливий практичний сенс. Закладені в його ідеї принципи паралельності і

близькодії міжклітинних взаємодій якнайкраще відповідають вимогам технології великих інтегральних схем. Апаратурна реалізація клітинних автоматів допомагала вирішити відразу дві проблеми: скорочення зв'язків на кристалі і збільшення продуктивності. Це викликало нову хвилю інтересу і сильно просунуло теорію клітинних автоматів: було запропоновано декілька модифікацій і формальних моделей клітинних автоматів, що дозволяють розробляти паралельні алгоритми і відображати їх на схемотехніку. В результаті була створена безліч паралельних високопродуктивних пристроїв таких, як паралельні суматори, кодери і декодери, розпізнавачі сигналів, тощо.

Клітинно-автоматна просторова динаміка

Новий сплеск інтересу до клітинних автоматів був пов'язаний вже не з побудовою паралельних пристроїв, а з побудовою нових моделей обчислень. Це відбулося у середині 80-х років минулого століття, коли були запропоновані клітинні автомати, еволюція яких моделює процеси дифузії, розділення фаз, реакційно-дифузійні процеси, рухи солітонів, тощо.

Справжнім потрясінням основ в моделюванні просторової динаміки стала поява клітинно-автоматної газової динаміки, названої "Gas-Lattice", що в перекладі на українську звучить як "решітчастий газ". Дуже важливий той факт, що була строго математично доведена відповідність Gas-Lattice моделей рівнянню Навьє-Стокса. Спочатку вважали, що решітчастий газ має ширші можливості, ніж рівняння Навьє-Стокса і зможе моделювати також турбулентні потоки. Ці надії, на жаль, не виправдалися, але надзвичайно простим і алгоритмічним виявилось застосування моделі до моделювання потоків в пористих середовищах.

Передбачається, що цей новий підхід до моделювання природних явищ стане доповненням до традиційних моделей математичної фізики, заснованих на диференціальних рівняннях в часткових похідних і добре розвиненим чисельним методам їх вирішення на комп'ютерах, обчислювальна потужність і архітектура яких змінюється швидше, ніж створюється необхідне математичне забезпечення. Разом із зростанням продуктивності, яка стимулює розвиток обчислювальних методів, відбуваються також якісні зміни в архітектурі комп'ютерів, орієнтовані на поглиблення паралелізму і спеціалізації процесорів. Таким чином, мотивацією до розвитку і розповсюдження клітинно-автоматних моделей послугувало два чинники: технічний і математичний.

Технічний чинник, в свою чергу має два аспекти. По-перше, було підраховано що клітинно-автоматні алгоритми, втілені в "залізо" можуть збільшити продуктивність на 3 порядки і вище по відношенню до вирішення аналогічних задач традиційними методами, наприклад, вирішенню диференціальних рівнянь в часткових похідних. Для підтвердження цієї тези була побудована клітино автоматна машина (САМ-8), на якій проводилися випробування клітинно-автоматних моделей, зокрема, досліджувалися алгоритми решітчастої гідро- і газодинаміки. Інший технічний чинник полягає в тому, що з'явилися високопродуктивні багатопроцесорні суперкомп'ютери і

мережеві обчислювальні технології, які дозволили емулювати еволюцію клітинних автоматів дуже великих розмірів достатньо швидко. І, хоча в цих випадках великих прискорень обчислень не спостерігається, клітинно-автоматне моделювання на багатопроцесорних системах витримує конкуренцію з традиційним моделюванням.

Другий чинник, що дав право на існування клітинно-автоматному моделюванню, складають труднощі чисельного моделювання на основі вирішення диференціальних рівнянь, що виникли у зв'язку із зростанням розмірів і складності задач, а також необхідністю їх розміщення на багатьох процесорах. Основна проблема полягає в тому, що рішення стаціонарних задач математичної фізики зазвичай виконується неявними методами, які не накладають обмежень на обчислювальну стійкість, що робить їх швидшими, ніж явні. Проте, паралельна реалізація цих методів вимагає додаткових зусиль як математиків, так і програмістів, і завжди пов'язана з деградацією ефективності розпаралелювання при збільшенні кількості процесорів. При рішенні нестационарних задач перевага віддається явним методам, які легко і ефективно розпаралелюються, але обмежені умовами обчислювальної стійкості (вимагають застосування малого кроку за часом). Явні методи володіють багатьма властивостями клітинних автоматів, іноді розглядаються як "безперервні клітинні автомати".

Порівняння клітинно-автоматного моделювання з традиційними методами чисельного моделювання може бути зроблено скільки-небудь коректно тільки для конкретних завдань. У тих випадках, для яких відомі диференціальні рівняння, але невідома їх клітинно-автоматна модель або навпаки, клітинно-автоматна модель відома, а диференціальне рівняння не існує, вибору немає і порівняння не має сенсу. Більш того, для процесів з сильно вираженою нелінійністю, чисельні методи можуть виявитись дуже складними. І навпаки, для відомих добре пропрацьованих чисельних методів, а іноді і наявного програмного забезпечення, перевага віддається чисельному методу. Крім того, порівняння не може вважатися коректним ще і тому, що традиційний чисельний аналіз — добре розроблена наука, а клітинно-автоматне моделювання знаходиться у стадії початкового розвитку. Найголовніша теза при порівнянні клітинно-автоматного моделювання з традиційним моделюванням полягає в тому, що клітинний автомат не апроксимує (спрощує) диференціальні рівняння, а є альтернативною моделлю.

Згодом з'явилися і фізичні обґрунтування правомірності клітинно-автоматних моделей. Ньютонова механіка описує оборотні процеси, тоді як для природних явищ в яких взаємодії нелінійні, характерна безповоротність. Диференціальні рівняння іноді виявляються безсилі коли необхідно моделювати явища необоротні і далекі від рівноважних станів. Крім того, диференціальні рівняння при сильних нелінійностях і великих розмірах просторів вирішувати складно. Звідси виникло бажання повернутися до витоків виведень диференціальних співвідношень у фізиці, тобто до опису явищ на мікроскопічному рівні. Але, оскільки моделювати поведінку реальних молекул

неможливо через їх величезні кількості навіть в найменших об'ємах, то в моделях використовується поняття абстрактних (модельних) частинок і швидкостей їх руху в дискретному просторі, які легко виражаються в термінах клітинних автоматів. Такі моделі називаються "стилізованими під молекулярну динаміку". Стани таких клітинних автоматів представлені булевими векторами, в яких наявність частинки в клітці відповідає "одиниці", відсутність "нулю". Наявність моменту руху частинки в певному напрямі руху у частинки моделюється "одиницею" у відповідному розряді булевого вектора стану клітини.

За своєю поведінкою клітинні автомати розбиваються на 4 класи, що розрізняються типами конфігурацій, до яких еволюціонує клітинний автомат.

- **Клас 1.** Клітинні автомати, що еволюціонують до просторово однорідного глобального стану. Наприклад, всі клітини клітинного автомата приходять в стан 1 або всі клітини приходять в стан 0. Невеликі локальні збурення початкової конфігурації не змінюють термінального стану клітинного автомата.

- **Клас 2.** Клітинні автомати, що еволюціонують до простих стійких або періодично розташованих структур. Невеликі локальні зміни початкових конфігурацій можуть змінити термінальний стан на обмеженій ділянці клітинного автомата.

- **Клас 3.** Клітинні автомати, поведінка яких хаотична, тобто нескінченно змінюється непередбачуваним чином. Будь-яка локальна зміна початкового стану розповсюджується на весь клітинний автомат.

- **Клас 4.** Клітинні автомати, що еволюціонують до досить складних локальних фігур, іноді рухомих (розповсюдження хвилі). Зміни початкових станів спричиняють за собою складні нерегулярні зміни поведінки.

Четвертий клас є найважливішим для моделювання просторової динаміки, його поведінка поки ніякому математичному аналізу не піддається. Саме цей клас клітинних автоматів має обчислювальну потужність машини Тюрінга, тобто володіє універсальною обчислювальною здатністю. Активно розвиваються також і чисто теоретичні дослідження клітинних автоматів.

Корисно порівняти властивості клітинних автоматів з традиційними методами, які засновані на розв'язанні диференціальних рівнянь в часткових похідних, маючи на увазі тільки ті випадки, коли існує альтернатива вибору, тобто процес який моделюється може бути представлений і тією і іншою моделлю.

Основні переваги клітинно-автоматних моделей:

1. Відсутність помилок округлення. Оскільки стани кліток представлені булевими векторами, і обчислення ведуться над ними, всі розряди в них рівноправні, округлення не проводиться. При ітеративних обчисленнях це важливо.

2. Віртуальна необмеженість паралелізму. Допускається розрізання

області моделювання будь-яким чином. Розпаралелювання на будь-яку кількість процесорів не приводить до деградації ефективності розпаралелювання.

3. Простота завдання граничних умов. Межі областей моделювання представлені особливими клітинами, які мають граничні правила переходів. Навіть при дуже складних формах меж (пористі середовища, зігнуті трубки) алгоритм моделювання практично не ускладнюється в порівнянні з тим, коли перешкод немає.

Головні **недоліки** клітинно-автоматних моделей:

1. Автоматний шум, пов'язаний з дискретним представленням даних. Уникнути його неможливо, але при достатньо великому радіусі усереднювання, він стає непомітним.

2. Відсутність формальних методів "прогнозування" поведінки клітинного автомата, по заданих функціях переходу. Звідси відсутність методів синтезу функції переходу клітинного автомата по заданих властивостях його еволюції.

Вирішуючи саме проблему автоматного шуму була побудована більш витончена модель **Lattice-Boltzmann (решітчастий газ Больцмана)**. Ця модель зберігає всі властивості і принципи функціонування клітинних автоматів, окрім одної – дискретності станів. Стани такого автомату є векторами дійсних чисел, а не булевими векторами, як це було для звичайних клітинних автоматів. Звісно мінусом цієї моделі є значно ускладнена задача побудови функцій переходів. Але такі функції потрібно побудувати лише один раз, а натомість значно зменшується ефект автоматного шуму, що збільшує точність моделювання. Оскільки решітчастий газ Больцмана так само складається з клітин і функціонує за такими ж принципами, що й звичайний клітинний автомат (відрізняється лише будова елементарних автоматів – клітин), то ми можемо також вважати його клітинним автоматом.

4.3.1.2. Формальний опис моделі

Клітинний автомат визначається множиною клітин, які щільно заповнюють дискретний D -мірний простір. Відстань між центрами клітин $d=1$. Кожній клітині поставлений у відповідність елементарний автомат, входами якого є виходи кліток, розташованих від нього не далі ніж на відстані r , тобто в деякому обмеженому сусідстві. Всі елементарні автомати однакові. Перехід в новий стан всіх елементарних автоматів, що входять в клітинний автомат, відбувається одночасно, синхронно і переводить клітинний автомат в новий глобальний стан. Ітеративний процес переходів зі стану в стан називається еволюцією клітинного автомата. Таким чином, для задання класичного клітинного автомата досить визначити структуру дискретного простору (декартова решітка, n -мірний куб, гексагональна структура, тощо) і елементарний автомат. Більшість формальних моделей клітинних автоматів і

заснованих на них мов їх опису обмежуються цими двома поняттями. Цього достатньо для теоретичних дослідження класичного клітинного автомата, як математичного об'єкту. Проте, коли з'явилася необхідність в конструюванні алгоритмів, що моделюють складні процеси, такі представлення клітинного автомата виявилися дуже слабкими. Знадобилося введення у формальну модель засобів, що управляють функціонуванням клітинного автомата як в часі (включення, виключення, зміна кроку за часом), так і просторі (зміна області застосування функцій переходів, зміна розмірів сусідства, забезпечення спільної роботи декількох клітинних автоматів). Таким чином, виникла потреба в нових формалізмах, серед яких найкращим чином відповідає перерахованим вимогам так званий Алгоритм паралельних підстановок (АПП). Багаті виразні можливості АПП роблять його цілком достатнім для представлення клітинних автоматів різних типів і всіляких маніпуляцій з ними.

Клітиною називається пара $(x, m) \in A \times M$, де $x \in A$ - символ алфавіту A , що позначає стан клітини, а $m \in M$ - ім'я клітини з множини імен M , що позначає місце розташування клітини в дискретному просторі. Алфавіт A може бути будь-яким. Зазвичай він задається як $\{0,1\}^n$ (класичні автомати) або $[0,1]^n$ (решітчастий газ Больцмана). Стани можуть бути задані змінними, які визначені на одному з цих алфавітів.

Множина імен може бути будь-якою (не більше ніж зліченною) множиною. При моделюванні просторової динаміки його елементами зазвичай є набори просторових координат точок дискретного простору скінченних розмірів.

Визначення 1. Множина клітин $\Omega = \{(x, m)\}$, у якій немає клітин з однаковими іменами, називається клітинним масивом. Клітинні масиви, в яких стани визначені на множині A , а імена на множині M , складають клас клітинних масивів, що позначається $\Omega(A, M)$.

На множині імен визначені іменуючі функції $\phi : M \rightarrow M$. Іменуюча функція $\phi(m)$ визначає для будь-якої клітини m одного з її сусідів. Для декартових клітинних масивів іменуючі функції мають вид зсуву, $\phi(i, j) = (i+a, j+b)$, а, b — константи, які визначають пари взаємодіючих клітин.

Визначення 2. Скінчена множина іменуючих функцій називається шаблоном сусідства.

$$T(m) = \{ \phi_0(m), \phi_1(m), \dots, \phi_q(m) \}$$

Для кожної клітини $m \in M$ шаблон $T(m)$ визначає множину її сусідів. Нумерація клітин в $T(m)$ може бути будь-якою. При цьому прийнято вважати, що $\phi_0(m) = m$.

Визначення 3. Вираз вигляду

$$S(m) = \{ (x_0, \phi_0(m)), (x_1, \phi_1(m)), \dots, (x_q, \phi_q(m)) \}$$

називається локальною конфігурацією, якщо існує таке r , що відстань від m до будь-якого $\phi_k(m) \in T(m)$, $d(m, \phi_k(m)) < r$, де $T(m)$ - множина іменуючих функцій, які входять в $S(m)$, яка називається визначальним шаблоном локальної конфігурації.

Далі базуючись на викладених означеннях буде описано алгоритм паралельних та пов'язані з ним поняття.

4.3.1.3. Алгоритм паралельних підстановок

Визначення 4. Паралельною підстановкою називається вираз вигляду

$$\theta(m) : S(m) * S''(m) \rightarrow S'(m),$$

де

$$S(m) = \{ (x_0, \phi(m)), (x_1, \phi_1(m)), \dots, (x_q, \phi_q(m)) \},$$

$$S'(m) = \{ (f_0, \phi(m)), (f_1, \phi_1(m)), \dots, (f_q, \phi_q(m)) \},$$

$$S''(m) = \{ (g_0, \psi(m)), (g_1, \psi_1(m)), \dots, (g_n, \psi_n(m)) \}.$$

$S(m)$ називається базовою конфігурацією (або базою) $S''(m)$ -контекстом, а $S'(m)$ - правою частиною паралельної підстановки. База і права частина підстановки мають один і той же визначальний шаблон. Шаблон контексту від них відрізняється. Символи f_k в клітинах правої частини позначають функції від значень станів в клітинах бази і контексту, тобто

$$f_k = F(x_0, x_1, \dots, x_q, g_0, \dots, g_n).$$

Вони називаються функціями переходів і можуть бути булевими, дійсними, а також простими замінами символів відповідно до алфавіту, який використовується. Локальна конфігурація $S(m)$ зі змінними станами вважається сумісною з клітинним масивом $\Omega \in \Omega(A \times M)$, що позначається $S(m) \sqsubseteq \Omega$, якщо

1) для всіх $(x_k, \phi_k(m)) \in S(m)$, $k = 0, 1, \dots, q$, змінні x_k визначені на тому ж алфавіті, що і стан однойменної клітини $(a_k, \phi_k(m)) \in \Omega$.

2) однойменні клітини в $S(m)$ і Ω з константними станами рівні, тобто якщо $x_k = a$, то $(a, \phi_k(m)) \in \Omega$.

Паралельна підстановка $\theta(m)$ застосовна до клітинного масиву $\Omega \in A \times M$, якщо в ньому знайдеться підмножина клітин $M' \subseteq M$ з іменами $t' \in M'$, для яких задовольняється умова

$$S(m') * S''(m') \sqsubseteq \Omega$$

Якщо $M' = \emptyset$, то підстановка не застосовна. Застосування підстановки $\theta(m)$ до Ω в клітині $t' \in M'$ полягає в заміні станів в клітинах $(a_k, \phi_k(m'))$, на значення функцій f_k .

Паралельна підстановка є локальним оператором клітинно-автоматної моделі.

Визначення 5. Результат застосування локального оператора до всіх клітин масиву Ω називається глобальним оператором $\theta(\Omega)$.

Існує два основні режими застосування глобального оператора обчислення наступного глобального стану $\Omega(t+1)$: синхронний і асинхронний.

Синхронний режим припускає, що аргументи функції переходів – це стани клітин-сусідів на поточній ітерації t . На кожній ітерації клітини обчислюють значення нового стану i , потім, всі клітини разом синхронно замінюють старі стани на нові. Результат застосування паралельної підстановки до клітинного масиву Ω може бути представлений як різниця Ω базових клітин та всіх застосовних підстановок і додавання їх правих частин.

$$\theta(\Omega) = (\Omega \setminus \bigcup_{M'} S(m')) \cup (\bigcup_{M'} S'(m'))$$

При **асинхронному режимі** кожна клітина обчислює функцію переходу від тих значень станів сусідів, які на даний момент мають місце і відразу змінює свій стан. Таким чином, кожна ітерація клітинного автомата складається з $|\Omega|$ кроків, на яких тільки одна клітина змінює свій стан.

$$\theta(\Omega) = (((\Omega \setminus S(m_1)) \cup S'(m_1)) \setminus S(m_2)) \cup S'(m_2)) \dots \setminus S(m_{|M|}) \cup S'(m_{|M|})$$

Порядок вибору клітин $m_1, m_2, \dots, m_{|M|}$ в Ω може бути випадковим, але може бути і впорядкованим. Це визначається вимогами процесу, який моделюється.

Визначення 6. Скінчена множина паралельних підстановок $\Phi = \{\theta_1, \dots, \theta_l\}$, які визначені на одному і тому ж класі клітинних масивів $\Omega (A \times M)$, називається системою паралельних підстановок. Система паралельних підстановок Φ є застосовною до клітинного масиву Ω , якщо хоч би одна з підстановок $\theta_k \in \Phi$ є застосовною до нього. Однією ітерацією еволюції системи Φ до Ω є застосування всіх його підстановок до всіх клітин масиву. Для синхронного режиму результат має вигляд $\Phi(\Omega) = \bigcup_{k=1}^l \theta_k(\Omega)$.

Розділення лівої частини паралельної підстановки на дві частини: базу і контекст, зроблено для того, щоб при застосуванні її до будь-якої конкретної клітини $m_k \in M$ відокремити множину клітин $S(m_k)$, стани яких вона змінює, від множини клітин $S''(m_k)$, стани яких вона не міняє. При цьому стани клітин з контексту $S''(m_k)$ можуть бути змінені на тій же ітерації застосуванням цієї ж підстановки до іншої клітини m_g , якщо $S''(m_k) \cap S(m_g) \neq \emptyset$ або іншою підстановкою θ' з системи $\Phi = \{\theta, \theta'\}$.

Контексти паралельних підстановок грають двояку роль, що розрізняються своїм впливом на еволюцію. Відповідно, вони підрозділяються на контексти першого роду і контексти другого роду.

- Контексти першого роду містять клітини вигляду $(x_k, \psi_k(m))$, в яких $\psi_k(m)$ визначені на базовій множині імен, тобто в області еволюції клітинного автомата. Такі контексти характерні для класичних клітинних автоматів.

- Контексти другого роду виконують керуючі функції. Локальна конфігурація контексту другого роду має вигляд

$$S''(m) = \{(\alpha_1, m_1), \dots, (\alpha_n, m_n)\}$$

де $\alpha_k = \xi_k(x_1, \dots, x_n)$ - функція, значення якої грають роль предиката, що робить підстановку застосовною. Для обчислення значень функції ξ_k необхідні додаткові контекстні підстановки θ'' , визначені на $\Omega''(A'' \times M'')$. Змінні x_k можуть бути зовнішніми змінними, наприклад, випадковими числами від генератора випадкових чисел, або значеннями тактуючих імпульсів, або показниками області значень імен клітин, тощо.

Наступні контексти другого роду вважаються типовими:

Імовірнісний контекст $S_p''(m) = (\alpha_p, m_p)$, який вживається в імовірнісних клітинних автоматах. Його контекстна підстановка має вигляд:

$$\theta_p'' : (x, m_p) \rightarrow (\alpha_p, m_p), \quad \alpha_p = \begin{cases} 1, & \text{rand} < p \\ 0, & \text{rand} \geq p \end{cases}$$

де $m_p \in M''$, rand - випадкове число в інтервалі $[0,1]$, p - ймовірність застосовності підстановки.

Тактуючий контекст $S_t''(m) = (\alpha_t, m_t)$ застосовується для організації n-тактного синхронного режиму застосування підстановок.

$$\theta_t'' : (x, m_t) \rightarrow (\alpha_t, m_t), \quad \alpha_t = \begin{cases} 1, & t_{\text{mod } n} < p \\ 0, & t_{\text{mod } n} \geq p \end{cases}$$

де $m_t \in M''$.

Просторовий контекст $S_s''(m) = (\alpha_s, m_s)$ застосовується для виділення тих областей клітинного масиву, де підстановка повинна бути застосовна.

$$\theta_s'' : (x, m_s) \rightarrow (\alpha_s, m_s), \quad \alpha_s = \begin{cases} 1, & a < m < b \\ 0, & 0 < m < a \end{cases}$$

Контексти другого роду застосовуються також для композиції двох або більш підстановок в систему підстановок.

Паралельна підстановка θ , визначена на $\Omega(A \times M)$, називається **несуперечливою**, якщо результат її $\theta(\Omega)$ над будь-яким $\Omega \in \Omega(A \times M)$ є клітинним масивом, тобто не містить двох різних клітин з одним і тим самим ім'ям. Несуперечлива підстановка при одночасному застосуванні в різних клітинах ніколи не намагається записати різні стани в клітину з одним і тим самим ім'ям, тобто

$$((x_1, m_1) \in S'(m) \ \& \ (x_2, m_1) \in S'(m)) \Rightarrow (x_1 = x_2).$$

Якщо права частина підстановки містить тільки одну клітину, як це має місце у класичному клітинному автоматі, то вона завжди несуперечлива. Якщо ж $|S'(m)| > 1$, то існує небезпека колізії при синхронному режимі. Встановити наперед суперечність дуже складно. Тому, зазвичай задовольняються **достатньою умовою несуперечності**, яка для однієї підстановки має наступний вигляд

$$T'(m_g) \cap T'(m_h) = \emptyset \quad \forall (m_g, m_h) \in M \times M$$

де $T'(m_g)$ та $T'(m_h)$ — шаблони локальної конфігурації $S'(m)$ при $m = m_g$ та $m = m_h$, відповідно.

Суперечлива підстановка не може виконуватися синхронно. Проте, її можна замінити на систему несуперечливих підстановок, які застосовуються на різних підтактах ітерації, використавши тактуючий контекст. При асинхронному режимі колізії неможливі, оскільки ніякі дві клітини одночасно не змінюють своїх станів. Представлені однією і тією ж системою паралельних підстановок клітинні автомати при синхронному, багатотактному синхронному і асинхронному режимах функціонування мають різні еволюції і, отже, не можуть бути моделями одного і того ж явища.

Визначення 7. Несуперечлива система підстановок $\Phi = \{\theta_1, \dots, \theta_l\}$, спільно з режимом їх застосування, який може задаватися за допомогою контекстів другого роду, називається Алгоритмом Паралельних Підстановок (АПП).

Важливий той факт, що будь-який клітинний автомат і всі його відомі модифікації можуть бути описані в термінах АПП, оскільки ця формальна система володіє обчислювальною універсальністю. Крім того, для конструювання складних клітинно-автоматних моделей дуже важлива наявність такого засобу як контекст другого роду.

4.3.1.4. Еволюція клітинно-автоматних моделей потоків рідини. Решітчастий газ Больцмана

Класичні клітинно-автоматні моделі потоків рідини. Клітинно-автоматне моделювання потоків рідини утворює окремий напрям у клітинно-автоматному моделюванні, який одержав назву Gas-Lattice, що українською мовою перекладається як "решітчастий газ". Цей напрям об'єднує цілу групу моделей, в основі якої лежить поняття класичного клітинного автомата.

Принципи побудови клітинно-автоматних моделей виходять із загальних законів фізики, що діють в деякому "дискретному світі", стилізованому під молекулярну динаміку. Газ представлений деякими гіпотетичними частинками, що рухаються у вільному просторі і стикаються один з одним. Сусідство клітини визначається шаблоном сусідства. Рухомі частинка забезпечена вектором швидкості, направленим у бік одного з сусідів. Частинки володіють наступними **властивостями**:

- Маса частинки рівна одиниці.
- Координати місцеположення частинок можуть приймати значення тільки рівні іменам клітин з клітинного масиву.
- Вектор швидкості частинки може бути направлений тільки у бік однієї з сусідніх кліток.

У один і той же момент часу в одній клітці не може знаходитися більше однієї частинки з однаковими векторами швидкості.

Таким чином, в кожен момент часу стан клітини представлений булевим вектором $s = (s_1, \dots, s_b)$ довжини $b = b_n + b_r$, де b_n - кількість сусідніх кліток, b_r - кількість частинок спокою. Кожен розряд вектора визначає наявність ($s_l = 1$) або відсутність ($s_l = 0$) частинки з певним вектором швидкості c_l , направленим до l -ого сусіда. Частинки спокою мають нульову швидкість.

Алфавітом станів клітинного автомата є множина булевих векторів, $A = \{(s_1, \dots, s_b) : s_l \in \{0, 1\}\}$, потужність алфавіту $|A| = 2^b$. Множина імен $M = \{m : m = (i, j, k)\}$ містить набори координат дискретного простору. Шаблон сусідства $T(m) = \{m, \phi_1(m), \phi_2(m), \dots, \phi_{b_n}(m)\}$. Режим роботи клітинного автомата - синхронний.

Функція переходів елементарного автомата будується згідно наступним правилам.

Кожна ітерація складається з двох тактів, званих фазою зсуву і фазою зіткнення. Чергування фаз регулюється тактуючою контекстною кліткою m_t , стани якої $x = 0$ і $x = 1$ дозволяють застосування підстановок фази зрушення і зіткнення, відповідно.

Фаза зсуву моделює складову перенесення, тобто відбувається переміщення всіх рухомих частинок на одну клітку в напрямі вказаному її вектором швидкості. Паралельна підстановка, що описує фазу зсуву має контекст першого роду. Базова частина підстановки містить тільки одну клітку. Функція переходу фази зсуву детермінована навіть в імовірнісних моделях. Формально фазу зсуву зручно представляти у вигляді системи підстановок, кожна з яких визначає зміни одного компоненту вектора станів, тобто переміщення однієї частинки.

$$\theta_l : \{(s_l, m)\} * \{(s_l', \phi_l(m)), (m_t, 0)\} \rightarrow \{(s_l', m)\}, \quad l=1, \dots, b_n$$

У фазі зіткнення відбувається зміна напрямку руху частинок, коли вони з'являються в одній і тій же клітці. Фаза зіткнення описується одноклітинною підстановкою, що не має контексту першого роду

$$\theta(m) : \{(s, m)\} \rightarrow \{(s', m)\}$$

у якій проводиться заміна одного стану $s = (s_1, \dots, s_b)$ клітки на інше $s' = (s_1', \dots, s_b')$. Функція переходів фази зіткнення може бути як детермінованою, так і імовірнісною. Зазвичай ці функції задаються у вигляді булевих таблиць з числом рядків, рівним 2^b . При цьому в кожному рядку міститься декілька рівноймовірних вихідних станів.

Головна умова, якій повинна задовольняти функція переходів: забезпечення виконання законів збереження маси і імпульсу. Щоб ці вимоги виконувалися, потрібне щоб: 1) булеві вектори станів s і s' мали рівне число одиниць, і 2) сума векторів швидкостей частинок, відповідних одиницям в s і s' були рівні, тобто

$$\sum_{l=1}^b s_l = \sum_{l=1}^b s_l', \quad \sum_{l=1}^b c_l = \sum_{l=1}^b c_l'$$

Першою клітинно-автоматною моделлю газової динаміки стала модель «НРР». Ця модель заповнює двовимірний простір квадратними клітинами зі стороною 1, кожна клітина може містити до 4-х частинок (по кількості найближчих сусідів), а правила переходу задаються лише двома співвідношеннями. Як видно ця модель є дуже простою, але навіть вона дозволила достатньо точно моделювати певні явища (наприклад розповсюдження хвилі, та інше). Крім цього вона дала старт розвитку теорії, і наступна модель, яка отримала назву «ФНР» вже була більш складною, а отже більш точною.

«ФНР» модель також працює в двовимірному просторі, але клітинами автомату є шестикутники. Таким чином кожна клітина має 6 сусідів. Існує три модифікації цієї моделі: ФНР-I, ФНР-II і ФНР-III. Вони відрізняються між собою кількістю частинок спокою, а також кількість правил переходів. Ця модель мала великий успіх, що стимулювало пошук аналогічних моделей для тривимірного простору. Але на жаль ці пошуки не були такими ж плідними.

Решітчастий газ Больцмана. Як вже було зазначено цей тип атоматів відрізняється від класичного лише тим, що стани клітин приймають значення з простору векторів дійсних чисел від 0 до 1, а не булевих векторів. Тобто стан клітини тепер відповідає не на запитання «є чи немає частинки з певним вектором швидкості?», а на запитання «з якою ймовірністю ця частинка є?». Хоча слід зазначити, що іноді стан клітини решітчастого газу Больцмана трактується як концентрація частинок з певним вектором швидкості в клітині, від деякої максимально можливої концентрації, але очевидно, що ця різниця трактувань є несуттєвою і не впливає на функціонування моделі. Таке трактування стану клітини пішло від функції розподілу молекулярних ймовірностей, відомої в кінетичній теорії газів, а в основу функцій переходів покладено рівняння Больцмана, побудоване на цій функції.

Визначення. Функція розподілу молекулярних ймовірностей:

$f(\vec{r}, \vec{v}, t)$ - ймовірність того, що в точці \vec{r} в час t знаходиться частинка, що рухається зі швидкістю \vec{v} .

Локальні макроскопічні величини визначаються з цієї функції інтегруючи за швидкістю:

- Сумарна маса частинок в точці \vec{x} в момент часу t :

$$p(\vec{x}, t) = m \int f(\vec{x}, \vec{v}, t) d\vec{v};$$

- Швидкість потоку в точці \vec{x} в момент часу t :

$$\vec{u}(\vec{x}, t) = \frac{m}{p} \int \vec{v} f(\vec{x}, \vec{v}, t) d\vec{v};$$

- Температура:
$$T = \frac{m}{3pR} \int |\vec{v} - \vec{u}|^2 f(\vec{r}, \vec{v}, t) d\vec{v},$$

де m – маса частинки, R – універсальна газова константа (за умов незмінної температури $\sqrt{RT} = c_s$ - швидкість розповсюдження звуку).

Користуючись функцією розподілу просторова динаміка газів і рідин задається рівнянням Больцмана:

$$\partial_t f + \vec{u} \vec{\nabla}_{\vec{x}} = \Omega(f).$$

Ліва частина цього рівняння описує переміщення частинки в просторі відповідно до її вектору швидкості, права частина – зіткнення частинок.

Функцію зіткнень отримують через детальний опис зіткнення двох молекул. Зазвичай для цього використовується припущення, що частинки є недеформуємими сферами, що стикаються абсолютно пружно.

Найчастіше використовується наступна форма функції зіткнень:

$$\Omega(f) = \Omega(\vec{r}, \vec{v}, t) = -\omega(f(\vec{r}, \vec{v}, t) - f^{eq}(p(\vec{r}, t), \vec{v})), \text{ де}$$

$$f^{eq}(p, \vec{v}) = \frac{P}{(2\pi RT)^{3/2}} \exp\left(-\frac{|\vec{v} - \vec{u}|^2}{2RT}\right) - \text{рівняння Максвелла-Больцмана, а } \omega -$$

константа релаксації, специфічна для конкретної рідини.

Хоча така модель зіткнень є дещо спрощеною – вона забезпечує коректну гідродинаміку і виконання основних теорем, пов'язаних з рухом газів і рідин. Далі буде виписано дискретні варіанти цих формул для використання їх в клітинному автоматі. Неперервний простір замінюється дискретним з кроком dr для позицій і кроком dt для часу. Тобто простір всіх позицій представлений дискретним набором точок, який в тривимірному варіанті позначається трьома індексами \vec{r}_{xyz} , ці точки знаходяться на сітці, з однаковою константною відстанню між вузлами за всіма трьома напрямками: $\vec{r}_{xyz} = \vec{r}_0 + (xdr, ydr, zdr)$, де \vec{r}_0 – точка початку координат. Простір всіх можливих швидкостей представлений набором з q векторів $\vec{v}_i, i=0..q-1$. Вони обираються таким чином, що кожен вектор v_i сполучає точку \vec{r}_{xyz} р деякою точкою $\vec{r}_{x_0y_0z_0}$, за один часовий крок, тобто: $\vec{r}_{x_0y_0z_0} = \vec{r}_{xyz} + \vec{v}_i dt$. Отриманий таким чином називається решіткою (сіткою) і позначається $DdQq$ – де d – розмірність простору, і q кількість векторів швидкостей.

Найчастіше в тривимірному просторі використовується решітка D3Q19 (Рис. 4.3.1.4.1):

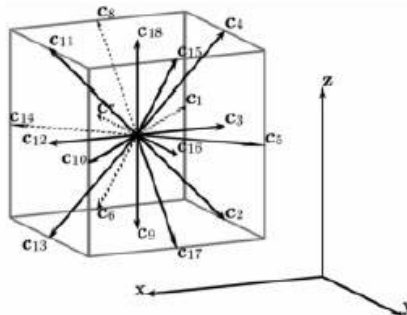


Рисунок 4.3.1.4.1. Решітка D3Q19.

Для цієї решітки вектори швидкостей \vec{v}_i задаються наступним чином:

$$\begin{aligned} v_0 &= (0, 0, 0); \\ v_1 &= (-1, -1, 0); & v_2 &= (-1, 0, -1); & v_3 &= (-1, 0, 0); & v_4 &= (-1, 0, 1); & v_5 &= (-1, 1, 0); \\ v_6 &= (0, -1, -1); & v_7 &= (0, -1, 0); & v_8 &= (0, -1, 1); & v_9 &= (0, 0, -1); \\ v_{10} &= (1, 1, 0); & v_{11} &= (1, 0, 1); & v_{12} &= (1, 0, 0); & v_{13} &= (1, 0, -1); & v_{14} &= (1, -1, 0); \\ v_{15} &= (0, 1, 1); & v_{16} &= (0, 1, 0); & v_{17} &= (0, 1, -1); & v_{18} &= (0, 0, 1) \end{aligned}$$

Враховуючи таку обмеженість простору швидкостей, доцільно виключити швидкість з параметрів функції f , і позначити її наступним чином: $f_i(\vec{x}, t) \equiv f(\vec{x}, \vec{v}_i, t)$. Враховуючи ці позначення запишемо дискретний варіант рівняння Больцмана. Ліва частина прийме вигляд: $f_i(\vec{x} + dt \cdot \vec{v}_i, t + dt) - f_i(\vec{x}, t)$. Оскільки права частина не містить похідних і інтегралів вона залишиться незмінною з точністю до позначень: $-\omega(f_i - f_i^{eq})$. Крім того спростимо й

$$f_i^{eq} : f_i^{eq} = p t_i \left(1 + \frac{\vec{v}_i \cdot \vec{u}}{c_s^2} + \frac{1}{c_s^4} \left((\vec{v}_i \cdot \vec{u})^2 - c_s^2 |\vec{u}|^2 \right) \right), \text{ де } p = m \sum_{i=0}^q f_i, \vec{u} = \frac{m}{p} \sum_{i=0}^q \vec{v}_i f_i,$$

а t_i – вагові константи, що компенсують різницю в модулях векторів швидкостей. Список значень цих констант:

$$t = \begin{bmatrix} 1/3 \\ 1/36 & 1/36 & 1/18 & 1/36 & 1/36 \\ 1/36 & 1/18 & 1/36 & 1/18 \\ 1/36 & 1/36 & 1/18 & 1/36 & 1/36 \\ 1/36 & 1/18 & 1/36 & 1/18 \end{bmatrix}$$

Дослідження показали, що записані вище дискретні рівняння апроксимують рівняння Нав'є-Стокса з другим порядком точності. Це означає, що решітчастий газ Больцмана є адекватним методом розв'язку поставленої задачі.

Повертаючись до моделі клітинного автомата – будемо вважати клітиною (елементарним автоматом), кожен вузол дискретної решітки. Стан клітини з координатами $\vec{x} = (x, y, z)$ в момент часу t – це вектор дійсних чисел: $(f_0(\vec{x}, t), \dots, f_q(\vec{x}, t))$. Такт роботи клітинного автомату (що відповідає одній одиниці дискретного часу dt) – паралельна зміна станів всіх клітин автомату, за наступним правилом: $f_i(\vec{x} + dt \cdot \vec{v}_i, t + dt) = f_i(\vec{x}, t) - \omega(f_i(\vec{x}, t) - f_i^{eq})$.

Представлення решітчастого газу Больцмана в термінах алгоритму паралельних підстановок майже не відрізняється від представлення класичних клітинно-автоматних моделей. Єдина відмінність полягає в тому, що тут фаза зсуву і фаза зіткнень об'єднана.

4.3.2. Застосування решітчастого газу Больцмана для моделювання руху крові в системі судин

4.3.2.1. Типи елементарних автоматів

Очевидно, що описана в загальному вигляді модель є неповною для моделювання конкретних явищ, адже вона не містить граничних умов. Для задання граничних умов введемо **два додаткових типи елементарних автоматів** (клітин), функціонування яких буде дещо відрізнятися від основних клітин.

1. **Клітини-генератори:** ці клітини здатні запам'ятати певний стан, і з певною періодичністю відтворювати його. Використовуючи ці клітини буде змодельовано вихід крові з серця.

2. **Клітини-стінки:** моделюють відбиття частинок від стінок.

Існуючі підходи до побудови клітин-стінок:

- **«віддзеркалення»** - цей підхід підходить тільки для «стінок», розміщених паралельно одній з осей координат і полягає в безпосередній заміні елементів вектори стану один на одного у відповідності з розміщенням «стінок»

- **«константа»** - цей метод полягає в тому що для клітини зафіксований певний стан, який ніколи не змінюється. Наприклад стан за якого сумарний вектор швидкості направлений вздовж стіни, або перпендикулярно до неї. Очевидно, що цей метод є надто неточним.

- **точний метод** полягає в знаходженні сумарного вектору швидкості, віддзеркаленні його відносно вектору нормалі до стіни у цій точці, а потім знаходження стану клітини, за якого сумарний вектор буде співпадати з знайденим. Хоча цей метод і є найточнішим – він пов'язаний з розв'язанням системи рівнянь у кожній клітині-стінці на кожному кроці роботи автомата.

У цій роботі запропоновано використання методу, що є деяким поєднанням методу віддзеркалення і розв'язку системи. Хоча він і має дещо більшу похибку, ніж точний метод, але він потребує значно менших обчислювальних витрат, що є критичним для моделювання руху крові в реальному часі. **Алгоритм роботи методу** полягає в наступному:

- Цикл по всім ненульовим векторам швидкостей $\vec{v}_i, i = \overline{1, q-1}$.
- Знаходження віддзеркалення вектору швидкості відносно вектору нормалі.
- Знаходження вектору швидкості, найближчого до отриманого віддзеркалення.
- Запис значення елемента вектору стану, що відповідає початковому вектору, у елемент, що відповідає знайденому вектору.

Очевидно, що перші три кроки можна виконати один раз при створенні клітини, а тому, як зазначено вище, цей метод є достатньо швидким.

Еволюція клітинного автомату (тобто зміна станів клітин) відбувається паралельно, за правилами, описаними вище, з урахуванням відмінностей у типах клітин.

4.3.2.2. Система судин

В цій роботі запропоновано моделювання руху крові, по складній, розгалуженій системі судин. Рух крові починається в аорті, яка далі розгалужується на судини, меншого діаметру, які в свою чергу роблять теж саме, і т. д. Тобто можна сказати, що система будується за фрактальним принципом (Рис. 4.3.2.2.1).

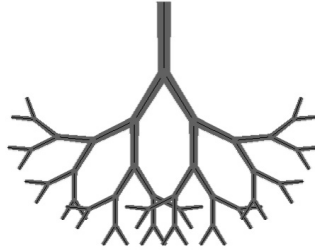


Рис. 4.3.2.2.1. Розгалужена система судин.

Побудована система заповнюється елементарними автоматами, у відповідності до позиції. Перший шар клітин, що відповідає початку аорти, заповнюється клітинами-генераторами. Всі клітини, що відповідають стінкам судин заповнюються клітинами-стінами, яким при створенні передається розрахований у відповідності до структури судини вектор нормалі. Всі інші клітини, що знаходяться в середині судин заповнюються клітинами звичайного типу.

4.3.2.3. Програмна реалізація

Для реалізації запропонованої моделі руху крові, був створений програмний комплекс з використанням технології Microsoft .NET, який має об'єктно-орієнтовану структуру (Рис. 4.3.2.3.1). Представлено дещо спрощену модель основних класів, що відповідають за реалізацію описаної в попередніх розділах моделі. Далі буде коротко описано призначення кожного з них:

- CellState – зберігає стан клітини.
- Cell – абстрактна клітина (елементарний автомат) клітинного автомату. Для кожної клітини зберігається її положення в просторі, посилення на всіх сусідів, а також свій поточний та новий стан. Для клітини описано (визначення покладається на конкретні реалізації клітин) методи ChangePhase() та MovePhase(), що відповідають за дві фази роботи автомату. Слід зазначити, що навіть для автоматів, ітерація яких передбачає лише одну фазу (напр. решітчастий газ Больцмана), нам все одно необхідно розділити її на дві віртуальні фази, для моделювання паралельного виконання, на послідовному, взагалі кажучи, комп'ютері.
- LBCell – абстрактна клітина решітчастого газу Больцмана: в ній реалізовано методи, що відповідають за ітерацію автомату. Таким чином,

наслідуючи цей клас, можна створити клітину будь-якого решітчастого газу, задавши лише вектори швидкостей та вагові коефіцієнти.

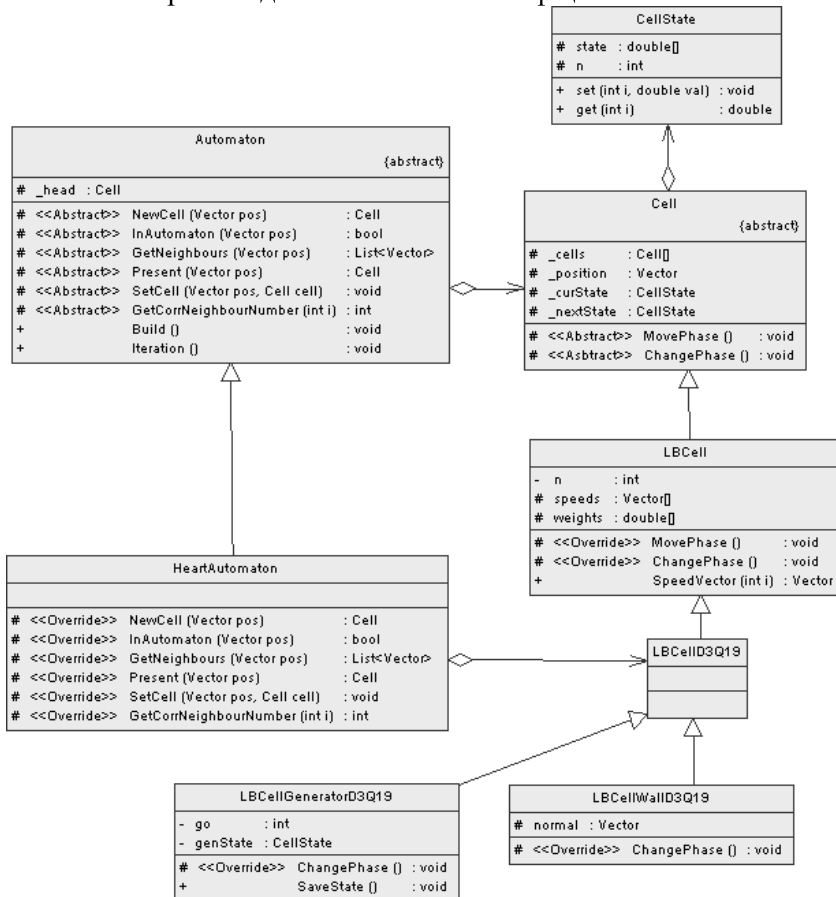


Рис. 4.3.2.3.1. Спрощена UML модель ієрархії класів.

- LBCellID3Q19 – конкретна реалізація класу LBCell, що моделює клітину звичайного типу решітки D3Q19. Як було вказано вище, саме цей тип решітки використовується в побудованій моделі.

- LBCellGeneratorD3Q19 – клітина-генератор, в решітці D3Q19. Має додаткові поле для зберігання стану, який відтворює (genState), та для зберігання частоти відтворення цього стану (go). Має додатково метод для зберігання поточного стану в genState, та перевизначає одну з фаз роботи, для відтворення своєї поведінки.

- LBCellWallD3Q19 – клітина-стінка, в решітці D3Q19. Має додатково поле normal для зберігання вектору нормалі до стінки в точці, що відповідає позиції цієї клітини. Перевизначає одну з фаз роботи для відтворення своєї поведінки.

- Automaton – абстрактний автомат, в якому реалізовано шаблонні методи Build() – побудова автомату, та Iteration() – ітерація автомату. Ці методи

використовують віртуальні методи, що мають бути реалізовані в конкретному автоматі:

- `NewCell(pos)` – повертає нову клітину з позицією `pos`. Має використовувати позицію, для визначення необхідного типу клітини.
- `InAutomaton(pos)` – перевіряє, чи знаходиться позиція `pos` в межах автомату.
- `GetNeighbours(pos)` – за позицією повертає позиції всіх сусідів клітини, з позицією `pos` (сусід – клітина, до якою частинка може перейти за один крок).
- та інші, менш важливі методи.
- `HeartAutomaton` – конкретна реалізація класу `Automaton`, що відповідає клітинному автомату, реалізує поведінку описану в попередніх розділах, а отже моделює рух крові по системі судин.

Далі буде описано деякі особливості такої програмної реалізації.

Особливості програмної реалізації.

Необхідно зазначити, що окрім посилань в клітинах один на одного, в класі `HeartAutomaton`, для пошвидшення доступу, зберігається тривимірний масив посилань на всі клітини простору. Оскільки ця реалізація орієнтована на моделювання руху крові в реальному часі, нехтуючи втратами пам'яті, оптимізована швидкодія.

Інше суттєве зауваження стосується алгоритму побудови автомату, заснованому на шаблонному методі `Build()`, та відносно простих віртуальних функціях. Така реалізація дозволяє майже без змін в програмі ввести в автомат додаткові елементи, наприклад патології судин.

В свою чергу ієрархія класів, що відповідають клітинам, дозволяє змінювати їхню поведінку, додавати нові типи клітин в будь-яких інших місцях програми.

Інтерфейс користувача поділений на дві частини: панель керування і вікно перегляду.

На панелі керування задаються початкові умови, для побудови системи судин, а також можна змінювати деякі параметри моделі і перегляду.

У вікні перегляду зображується рух деякої кількості абстрактних «частинок». Швидкість і маса частинки визначається значеннями \vec{u} і p клітини в якій вона знаходиться. Частинки можуть зображуватися в двох режимах: в режимі векторів, та в режимі точок. В режимі векторів (рис. 4.3.2.3.2), кожна частинка зображується вектором, напрям якого співпадає з напрямком вектору \vec{u} , довжина пропорційна довжині \vec{u} , а інтенсивність кольору пропорційна значенню p . В режимі точок (рис. А.3), кожна частинка зображена точкою, що рухається вздовж вектора \vec{u} .

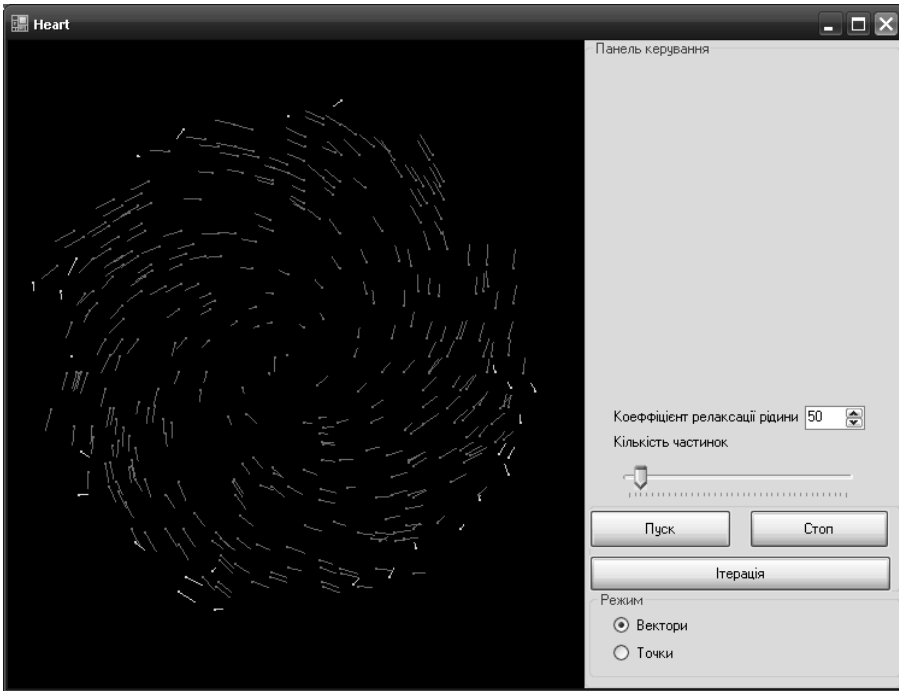


Рис. 4.3.2.3.2. Програмна реалізація: режим векторів.

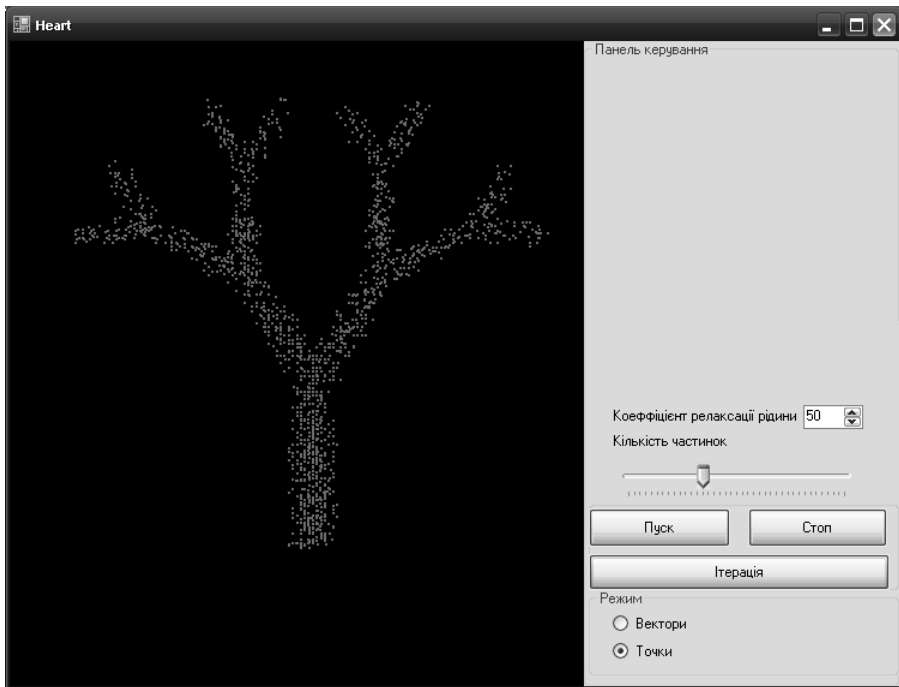


Рис. 4.3.2.3.3. Програмна реалізація: режим точок.

Висновки до розділу 4.3

В ході виконання роботи було розглянуто теорії, що дозволяють моделювати рух рідини. Серед цих теорій був обраний решітчастий газ Больцмана, який має всі позитивні риси класичних клітинно-автоматних моделей, але не має деяких суттєвих недоліків. Було розроблено метод використання решітчастого газу Больцмана, для моделювання руху крові в системі судин в *реальному часі*.

На основі розробленого методу було створене відповідне програмне забезпечення. Воно було спроектовано таким чином, щоб дозволити максимально легке розширення і вдосконалення:

- зміна способу побудови автомату, в тому числі моделювання деяких патологій судин через зміну структури клітинного автомату;
- вдосконалення функцій переходів автоматів, а отже і функціонування автомату;
- вимірювання тиску на стінки судини в певному місці, що є необхідним для моделювання деяких хвороб.

Подальшими напрямками розвитку цієї роботи будуть:

- вдосконалення програмної реалізації у наведених вище напрямках;
- розробка клітинно-автоматної моделі серця, для більш точного моделювання виходу крові з нього;
- вдосконалення функцій переходів, для того, щоб урахувати багатофракційність крові;
- розробка нових типів крайових клітин, для моделювання зміни діаметру судин;
- вдосконалення системи побудови судин, та моделювання повного кола кровообігу людини.

ВИСНОВКИ

Розділ 1.

1. В роботі надані визначення, встановлений зв'язок із суміжними поняттями, прослідкована історія розвитку, виявлені основні тенденції та сформовані прогнози подальшого розвитку **Інтернету Всього (ІоЕ)**.

2. Важливим аспектом ІоЕ, який відрізняє ІоЕ від ІоТ, є "мережевий ефект". ІоТ - лише складова частина ІоЕ. ІоЕ ґрунтується на зв'язках між людьми, процесами, даними і речами, причому мова не йде про кожен з цих вимірів окремо. Кожен підсилює можливості трьох інших, і справжня сила ІоЕ проявляється на їх перетині.

3. Перспективи розвитку Інтернету.

- ІоЕ вдихне життя в усе, що тільки можна собі уявити, підключаючи неживий світ до Інтернету..
- Всеохоплюючий Інтернет буде сприяти розвитку економіки.
- Всеохоплюючий Інтернет стане більш персональним і прогнозованим.
- Всеохоплюючий Інтернет сприятиме схрещуванню технологій з гуманітарними науками.

4. За оцінками компанії Cisco, 99% фізичних об'єктів, які можуть стати частиною ІоЕ, до Мережі поки що не підключені. У наступному десятилітті завдяки підключенню того, що ще не підключено, ІоЕ може забезпечити приватному сектору світової економіки прибуток, що обчислюється 14,4 трлн. доларів США.

5. Три ключові чинники, що сприяють прийняттю ІоЕ бізнесом.

- зростання темпів інновацій (36% опитаних)
- необхідність задовольнити зацікавленість споживачів в нових методах взаємодії (35%)
- необхідність автоматизувати бізнес-процеси (35%).

6. Головні технологічні фактори впровадження ІоЕ:

- нові типи пристроїв (33%),
- зростаючий обсяг даних (32%)
- хмарні технології (31%).

7. Щоб отримати максимальну користь з економіки ІоЕ, компанії повинні взяти на озброєння стратегію, яка передбачає:

- інвестування в високоякісні технологічні інфраструктури та інструменти,
- повсюдне впровадження методів бізнесу, що дозволяють всім співробітникам вносити особистий вклад у загальну справу,
- розробку ефективних методів управління інформацією.

8. Головним фактором, що визначає ефективність використання можливостей ІоЕ, є якість технологічної інфраструктури та відповідних інструментів.

9. Компаніям слід зосередитися на тих можливостях ІоЕ, які найбільш значимі для їх галузей.

Для виробничих підприємств це - багатовимірний аналіз даних в реальному часі, спільна робота з інтеграцією відеозв'язку, віддалений контроль фізичних активів.

Для енергетичних компаній - інтеграція даних лічильників, можливість знаходити потрібних фахівців, попереджуючий аналіз.

Для компаній роздрібною торгівлі - попереджуючий аналіз і візуалізація даних, BYOD (Bring Your Own Device), взаємодія з замовниками шляхом застосування мультимедійних технологій, мобільні платежі і віддалений моніторинг споживачів.

10. Фірми і компанії середнього розміру з країн з ринком, що формується, мають відмінні і при цьому зростаючі шанси скласти гідну конкуренцію нинішнім лідерам.

11. Для компаній важливо не упустити момент зміни технологій. В результаті цифрової революції в найближчі п'ять років в 12 галузях зникне 40% компаній. Що стосується ТОП-500 світових корпорацій, то тут в найближчі десять років очікується зникнення 35% компаній, які пропусають або не впишуться в цей цифровий перехід.

12. В першу чергу, необхідно забезпечити інформаційну безпеку.

13. Зміни в світі цифрових технологій відбуваються все швидше. Невиробничі компанії: раніше - 3-5 років, тепер - 1-1,5 років. Промисловість: раніше - 10-12 років, тепер - 2 роки.

14. За консервативними оцінками, в світовому масштабі потенційна економічна вигода ІоЕ в найближче десятиліття може скласти \$ 19 трлн. З них: \$ 14,4 трлн - приватний бізнес, \$ 4,6 трлн - держсектор.

15. Основні можливості для реалізації ІоТ вже є: новий інтернет-протокол - IPv6; мікрочіпи з наднизьким споживанням енергії; бездротові модулі.

16. Чотири технологічні опори ІоЕ: мобільні технології, хмарні технології, великі дані, речі.

17. Основні сфери застосування ІоЕ в промисловості: «розумне» місто, транспорт, важка і легка промисловість, будівництво, енергетика, безпека, медицина, ритейл; в повсякденному житті: носимі пристрої, "розумний будинок.

18. Сьогодні ми маємо близько 50 млрд. підключених пристроїв: транспорт - 500 млн пристроїв (аварійне реагування, самодіагностика автомобіля та ін.); розумне місто - 2 млрд пристроїв (розумне освітлення, розумні парковки, інтелектуальний транспорт, розумні будівлі і т.д.); безпека - 100 млн пристроїв (біометрія, сканери, контроль водних ресурсів, моніторинг навколишнього середовища та ін.); електронна охорона здоров'я - 100 млн пристроїв (мобільні лабораторії, хірургічне обладнання, віддалений моніторинг, телемедицина та багато інших); промисловість - 1 млрд пристроїв (управління активами, розумні насоси, компресори, клапани, безпека (las), smart-мітки (nfc) і ін.); енергетика і жкг - 500 млн - 1 млрд пристроїв (розумні лічильники,

польові мережі і т.д.); торгівля - 200 млн пристроїв (заправні станції, ігри, соціальні події, автомати з їжею (вендинг), smart-мітки і т.п.)

Розділ 2

1. Спроектовано та реалізовано систему, яка за допомогою даних вбудованих сенсорів розпізнає ознаки небезпеки та надає інформацію про можливу загрозу здоров'ю користувача.

2. В роботі розроблене програмне забезпечення для оцінки ефективності процесу тренувань, для ознайомлення з прогресом рівня підготовки та підбору оптимального навантаження.

3. Запропоновано підхід, що дозволяє збільшити час роботи датчиків шляхом керування потужністю.

Описано вплив потужності датчику на площу зони покриття датчику. Зменшення використання енергії досягається шляхом зменшення зони покриття при перетині у мінімальній площі з іншим датчиком в зоні покриття.

Наведена відповідна залежність між площею покриття та величиною використання енергії за одиницю часу.

Запропоновано математичну модель у вигляді оптимізаційної задачі лінійного програмування. Описано покроковий алгоритм розв'язання поставленої задачі.

Було проведено ряд комп'ютерних експериментів, що підтвердили ефективність запропонованої інформаційної технології.

Результати дослідження можуть бути використані як у екологічній сфері так і у технічній сфері.

Розділ 3

1. При побудові захищеної системи управління динамічним об'єктом, таким, як БПЛА, потрібно враховувати динаміку його руху.

2. Прийоми, що були використані при створенні моделі динаміки польоту літального апарату (ЛА) є універсальними і можуть використовуватись при створенні моделей інших складних систем.

3. В роботі розроблена бажана послідовність нарощування рівнів захищеності інформаційної системи як послідовність зміни варіантів стандартних функціональних профілів захищеності. Переваги розробленої послідовності полягають в тому, що вони мінімізують ризики переходів між різними рівнями захисту інформаційної системи з одного боку, та у чіткої відповідності діючим нормативним документам з іншого боку.

4. Вимоги щодо захисту інформації мають бути враховані вже на етапі проектування та створення програмного забезпечення, яке входить до складу автоматизованої системи управління. Наступною складовою має бути створення комплексної системи захисту інформації.

5. Найбільшу небезпеку цифровим системам управління складають загрози «нульового дня», тобто загрози абсолютно нові, про які нічого не відомо. Недоліками існуючих підходів є потреба у формуванні еталонних

значень для ситуацій, які швидко змінюються, для об'єктів, набір яких також швидко змінюється. Адаптований в роботі таксонометричний метод дозволяє автоматично виділяти потенційно небезпечні об'єкти в мережі управління за ознаками нетиповості їх поведінки для організації вчасної протидії.

7. Універсальність підходу полягає в тому, що математична постановка задачі ідентифікації загрози «нульового дня» для інформаційних та біологічних об'єктів має аналогічний вигляд. Тобто підхід може бути застосований як для технічних (цифрових) систем, так й для біологічних.

Розділ 4

4.1.

1. В рамках роботи було досліджено можливості створення інформаційної технології (програмно-апаратного рішення), що має на меті передачу цифрових даних від кінцевого пристрою до користувача у мережах типу LPWAN..

2. Були з'ясовані мінімально необхідні компоненти системи та їх призначення. Визначені шляхи обміну даними, у тому числі протоколи та алгоритми згідно з постановкою завдання.

3. Дослідженні алгоритми передачі даних у мережа IoT та висунуті гіпотези, щодо їх інтерпритації та обробки.

4. Дослідженні та класифіковані типові проблеми передачі даних та пониження QoS. Особливу увагу було приділено роботі з втратами під час передачі даних.

5. Запропоновано модель обміну графічною інформацією у мережах LoRaWAN, що може надсилати повідомлення на регулярній основі.

6. Дослідженно шляхи оптимізації передачі даних і запропоновано алгоритм регуляції QoS, що дозволяє тримати об'єм даних що передається на граничних мінімумах.

7. Напрямки використання рішення: спостереження у аграрному секторі, екології, лісництві, фермерстві, віддалений нагляд за пацієнтами тощо.

8. Інформаційна технологія дозволяє послідовно надсилати графічні повідомлення з кінцевого пристрою від 2 до 5 разів на добу, та гнучко корегувати якість отриманих зображень.

9. Напрямки подальших досліджень: подальша оптимізація об'єму даних, що передається за рахунок додавання обробки втрат пакетів.

4.2.

1. В рамках роботи запропоновано та описано спосіб рознесення бізнес-логіки користувача на велику кількість кінцевих вузлів.

2. Розвинутий підхід щодо встановлення оптимальної конфігурації для максимальної утилізації ресурсів обчислювального пристрою.

3. Досліджено платформу Node.JS у порівнянні з альтернативними технологіями асинхронного програмування для побудови застосунків реального часу, що призначені для розгортання у гібридній сервіс-орієнтованій

архітектурі безсерверних обчислень, та встановлено, що Node.JS - є оптимальним рішенням для створення прототипу.

4. Розвинутий підхід щодо встановлення оптимальної конфігурації для рознесення бізнес-логіки користувача в географічній близькості біля нього для покращення користувацького досвіду при використанні бізнес логіки. Запропоновано рішення для рознесення лише тої бізнес-логіки, яка необхідна користувачу в даний момент. Спроектване рішення дозволяє розташувати бізнес логіку в географічній близькості до кінцевого користувача.

5. За допомогою спроектваного рішення можливо обслуговувати багатьох користувачів різних бізнес-логіки що посилаються на одні і ті ж адреси.

6. Запропоновано спосіб розділення користувачів використовуючи підхід багатопоточного програмування та асинхронної моделі подій. Запропоновано розподіляти навантаження посилаючи користувачів на різні порти.

4.3

1. В роботі розглянуто теорії, що дозволяють моделювати рух рідини. Серед цих теорій був обраний решітчастий газ Больцмана, який має всі позитивні риси класичних клітинно-автоматних моделей, але позбавлений суттєвих недоліків. Було розроблено метод використання решітчастого газу Больцмана, для моделювання руху крові в системі судин в *реальному часі*.

2. Відповідне програмне забезпечення було спроектвано таким чином, щоб дозволити максимально легке розширення і вдосконалення:

- зміна способу побудови автомату, в тому числі моделювання деяких паталогій судин через зміну структури клітинного автомату;
- вдосконалення функцій переходів автоматів, а отже і функціонування автомату;
- вимірювання тиску на стінки судини в певному місці, що є необхідним для моделювання деяких хвороб.

3. Напрямки подальших досліджень:

- вдосконалення програмної реалізації у наведених вище напрямках;
- розробка клітинно-автоматної моделі серця, для більш точного моделювання виходу крові з нього;
- вдосконалення функцій переходів, для того, щоб урахувати багатofракційність крові;
- розробка нових типів крайових клітин, для моделювання зміни діаметру судин;
- вдосконалення системи побудови судин, та моделювання повного кола кровообігу людини.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бандман О.Л. Клеточно-автоматные модели пространственной динамики / О. Л. Бандман // Системная информатика: сб. научн. тр. – 2006. - №10
2. Бояров В. Всеобъемлющий интернет: технологии и маркетинг [Электронный ресурс] / В. Бояров. – 2015. – Режим доступа до ресурсу: <http://sptc.ru/raznoe/1511-vseob-emlyushchij-internet-tekhnologii-i-marketing>.
3. Бражиненко М.Г., Козачок П.А., Шевченко В.Л. . Проблемы публикации-подписки у бездротових мережах для вузлів Інтернету всього / MSTIoE 2018-3 : East European Conference on Mathematical Foundations and Software Technology of Internet of Everything (21-22.12.2017, Kyiv - Київ). Зб. Матер. С.32.
4. Бредли Д. Всеобъемлющий Интернет на весах ценности для частного бизнеса [Электронный ресурс] / Джозеф Бредли. – 2013. – Режим доступа до ресурсу: <http://www.iksmedia.ru/articles/4951514-Vseobemlyushhij-Internet-na-vesax.html>.
5. В столице Татарстана запущен пилотный проект в рамках создания комплексного решения "Умный" и безопасный город Казань [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: <http://www.cisco.com/web/RU/news/releases/txt/2014/09/090314c.html>.
6. Всеобъемлющий Интернет улучшит экологию в Копенгагене. Блог старшего вице-президента компании Cisco, главного директора по вопросам глобализации Вима Элфринка (WimElfrink) [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/06-062314b.html.
7. Градостроительная сказка стала былью в Барселоне [Электронный ресурс] – Режим доступа до ресурсу: https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/07-072814a.html.
8. Датчик движения (PIR Motion sensor) HC-SR501 [Электронный ресурс] – Режим доступа до ресурсу: <http://digitalchip.ru/datchik-dvizheniya-pir-motion-sensor-hc-sr501>.
9. Дин Д. Всеобъемлющий Интернет удешевляет «умные» города [Электронный ресурс] / Д. Дин. – 2014. – Режим доступа до ресурсу: https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/12-121214b.html.
10. Жданов А. MediaTech: технологии будущего в кинематографе [Электронный ресурс] / Алексей Жданов – Режим доступа до ресурсу: <http://www.telesputnik.ru/materials/tekhnika-i-tekhnologii/article/mediatech-tekhnologii-budushchego-v-kinematografe/>.
11. Завтра начинается здесь: город будущего. Блог старшего вице-президента компании Cisco, главного директора по вопросам глобализации Вима Элфринка (Wim Elfrink) [Электронный ресурс]. – 2014. – Режим доступа до

- ресурсы: https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/07-071414d.html.
12. Зотов В. Internet of What? В чем на самом деле разница между M2M, IoT и IoE. [Электронный ресурс] / Владимир Зотов. – 2016. – Режим доступа до ресурсу: <https://www.billing.ru/blog/internet-what>.
 13. Интернет вещей и Всеобщий Интернет [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: http://www.bizhit.ru/index/trend_www_traffic/0-171.
 14. Как Всеобъемлющий Интернет сформирует следующие четверть века Интернета [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: Как Всеобъемлющий Интернет сформирует следующие четверть века Интернета https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/03-033114d.html.
 15. Как обуздать Internet of Everything. Вторая статья главного футуролога компании Cisco Дэйва Эванса (Dave Evans) о Всеобъемлющем Интернете [Электронный ресурс] – Режим доступа до ресурсу: <http://press-relizy.ru/archive/hi-tech/102739.html>.
 16. Компания Cisco подписала дорожную карту по взаимодействию с Республикой Татарстан [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: <http://www.cisco.com/web/RU/news/releases/txt/2014/10/100914b.html>.
 17. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу. Затв.нак. Департаменту спеціальних телекомунікаційних систем та захисту інформації СБ України від 28.04.1999 р. №22 із змінами згідно нак. Адміністрації Держспецзв'язку від 28.12.2012 № 806 : НД ТЗІ 2.5-004-99. [електронний ресурс] - 2012. - Режим доступу: www.dsszzi.gov.ua/dsszzi/doccatalog/document?id=106342
 18. Медведев Ю.Г. Трехмерная клеточно-автоматная модель потока вязкой жидкости / Ю.Г. Медведев // Автометрия. – 2003. – №3
 19. Метрополис будущего – новые стандарты жизни. Блог старшего вице-президента компании Cisco, главного директора по вопросам глобализации Вима Элфринка (Wim Elfrink) [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/02-021114b.html.
 20. Мктарян Ю. Досвід використання LoRaWAN в системі АСУЕК в реальних міських умовах [Электронный ресурс] / Ю. Мктарян. – 2018. – Режим доступа до ресурсу: <https://habr.com/ru/post/419665/>.
 21. Мосеев В. Мережі LPWAN. Історія та перспективи [Электронный ресурс] / В. Мосеев. – 2017. – Режим доступа до ресурсу: https://iot.ru/promyshlennost/seti-lpwan-istoriya-i-perspektivy?sphrase_id=30552.
 22. На ваших экранах - «умный» цифровой город-порт Гамбург – Новости. Блог старшего вице-президента компании Cisco по работе в Центральной и Восточной Европе Майкла Ганзера (MichaelGanser) и исполнительного

вице-президента компанії Cisco по стратегії і розвитку [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/10-102714b.html.

23. Наказ від 16.05.2007 № 93 Про затвердження Положення про державну експертизу в сфері технічного захисту інформації. Зареєстровано в Міністерстві юстиції України 16 липня 2007 р. за № 820/14087 [Електронний ресурс] / Адміністрація державної служби спеціального зв'язку та захисту інформації України – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0820-07>.
24. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. Затверджено наказом ДСТСЗІ СБ України від 28.04.1999 №22.
25. НД ТЗІ 1.4-001-2000. Типове положення про службу захисту інформації в автоматизованій системі. Затверджено наказом ДСТСЗІ СБ України від 04.12.2000 №53.
26. НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу. Затверджено нак. Департаменту спеціальних телекомунікаційних систем та захисту інформації СБ України від 28 квітня 1999 р. №22. зі Зміною №1, затвердженою наказом Адміністрації Держспецзв'язку від 15.10.2008 № 172. : - 2008. - 20с. http://www.dsszzi.gov.ua/dsszzi/control/uk/publish/article?showHidden=1&art_id=101870&cat_id=89734&ctime=1344501089407
27. НД ТЗІ 3.7-001-99. Методичні вказівки щодо розробки технічного завдання та створення комплексної системи захисту інформації в автоматизованій системі. Затверджено наказом ДСТСЗІ СБ України від 28.04.1999 №22.
28. НД ТЗІ 3.7-003-05. Порядок проведення робіт зі створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі.
29. О платформе Всеобъемлющего Интернета / Блог Стюарта Тейлора (Stuart Taylor), директора подразделения компании Cisco по трансформации деятельности сервис-провайдеров [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/12-121814d.html.
30. Ответы на два самых частых вопроса про Всеобъемлющий Интернет / Блог главного футуролога компании Cisco Дэйва Эванса (Dave Evans) [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/c/ru_ua/about/press/2013/04042013f.html.
31. Панеш А. Х. Всеобъемлющий интернет - новый этап развития сети Интернет [Електронний ресурс] / А. Х. Панеш – Режим доступу до ресурсу: <https://cyberleninka.ru/article/n/vseobemlyuschiy-internet-novyuy-etap-razvitiya-seti-internet>.

32. Первый всероссийский хакатон по тематике «Умный» город» [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: <http://www.cisco.com/web/RU/news/releases/txt/2014/11/111114c.html>.
33. Петрівський В.Я. Збільшення часу роботи датчиків шляхом регулювання енерговитрат / В.Я. Петрівський, В.Л.Шевченко, М.Г. Бражиненко // Системи обробки інформації. – 2019. – № 3(158). – С. 36-41.
34. Подключение датчика температуры и влажности DHT11 к Arduino [Электронный ресурс] – Режим доступа до ресурсу: <http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:troyka-dht11>.
35. Подключение фоторезистора к ардуино и работа с датчиком освещенности [Электронный ресурс] – Режим доступа до ресурсу: <https://arduinomaster.ru/datchiki-arduino/photorezistor-arduino-datchik-sveta/>.
36. Правила забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах. Затверджені постановою КМУ від 29.03.06р. №373.
37. Прохоров А. Рынок IoT. Оценок много, нет консенсуса [Электронный ресурс] / Александр Прохоров // Huawei_Russia 17. – 2016. – Режим доступа до ресурсу: <https://habrahabr.ru/company/huawei/blog/312888/>.
38. Разработка и изготовление структур КНИ для производства интегральных схем и микроэлектромеханических систем / [С. П. Тимошенко, Е. П. Светлов-Прокопьев, В. И. Графутин та ін.]. // Материалы Международной научно-практической конференции «Фундаментальные проблемы радиоэлектронного приборостроения (INTERMATIC-2004). – 2004. – №1. – С. 52–55.
39. Рекламна діяльність: організація і практика [Электронный ресурс] – Режим доступа до ресурсу: <https://ru.osvita.ua/vnz/reports/management/14108/>.
40. Санітарні норми виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99.
41. Семашко П. В. Определение вклада метрополитена в суммарную акустическую нагрузку населения путем расчета суммарной дозы звуковой энергии / П. В. Семашко, В. В. Шкуро, А. В. Очеретяная.
42. Сервис-провайдеры могут возглавить внедрение технологий Всеобъемлющего Интернета Блог Стюарта Тейлора (Stuart Taylor), директора подразделения компании Cisco по трансформации деятельности сервис-провайдеров [Электронный ресурс] – Режим доступа до ресурсу: https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/12-121914a.html.
43. Смит М. Глобальный руководитель по развитию стратегии Cisco в области «Интернета вещей». Всеобъемлющий интернет и «умные» города / Мэтью Смит. // Control Engineering. – 2015. – №4. – С. 92–95.
44. Смит М. Глобальный руководитель по развитию стратегии Cisco в области Интернета вещей. Всеобъемлющий Интернет: нужно быть готовым к изменениям [Электронный ресурс] / Мэтью Смит. – 2015. – Режим доступа

- до ресурсу:
https://glavportal.com/materials/vseobemlyushchiy_internet_nuzhno_byt_gotovym_k_izm/.
45. Снитюк В. Є. Прогнозування. Моделі. Методи. Алгоритми. Навч. посібник. / Віталій Євгенович Снитюк. – Київ: Маклаут, 2008. – 364 с.
 46. Собираем показания датчиков с Android смартфона [Електронний ресурс] – Режим доступу до ресурсу: <https://habrahabr.ru/post/137678/>
 47. Технологии современных беспроводных сетей Wi-Fi / [Е. В. Смирнова, А. В. Пролетарский, Е. А. Ромашкина та ін.]. – Москва: Издательство МГТУ им. Н.Э. Баумана, 2017. – 446 с.
 48. ТР ЕОТ-95. Тимчасові рекомендації з технічного захисту інформації у засобах обчислювальної техніки, автоматизованих системах і мережах від витоку каналами побічних електромагнітних випромінювань і наводок. Затверджені наказом ДСТЗІ від 09.06.1995 №25.
 49. ТР ТЗІ-ПЕМВН-95. Тимчасові рекомендації з технічного захисту інформації від витоку каналами побічних електромагнітних випромінювань і наводок. Затверджені наказом ДСТЗІ від 09.06.1995 №25.
 50. Шакиб Т. Подключенное оборудование выводит производство на новый уровень [Електронний ресурс] / Тони Шакиб // Control Engineering. – 2015. – Режим доступу до ресурсу: <http://controlengrussia.com/internet-veshhej/evolyutsiya-iot-cisco/>.
 51. Шалаева О. А. Таксонометрический метод рейтинговой оценки деятельности различных типов и форм сельскохозяйственных организаций / О. А. Шалаева, В. И. Колеснев. // Проблемы экономики (Белорусская гос. с/хоз. акад.). – 2011. – С. 237–244.
 52. Школьники Казани ознакомятся с возможностями Всеобъемлющего Интернета на «золотом уроке» Cisco [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://www.cisco.com/c/ru_ru/about/press/press-releases/2014/11-111714a.html.
 53. Шнайдер С. Протоколы «Интернета вещей»: основные сведения. [Електронний ресурс] / Стэн Шнайдер – Режим доступу до ресурсу: <http://www.rtsoft.ru/press/articles/detail.php?ID=2718>.
 54. Эра всеобъемлющего интернета // презентация Холдинг «Зерде» совместно с компанией «Cisco» на территории Astana EXPO в Уголке «Кремниевая долина» [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://zerde.gov.kz/press/news/era-vseobemlyushchego-interneta/>.
 55. Эра всеобъемлющего интернета и розничной торговли: беспрецедентные возможности для связи продавцов и покупателей [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: <http://www.comprice.ru/articles/detail.php?ID=507306>.
 56. A survey of cyber-physical systems / J. Shi, J. Wan, H. Yan, H. Suo. // Wireless Communications and Signal Processing (WCSP), International Conference.. – 2011. – С. 1–6.

57. A Survey on LoRaWAN Architecture, Protocol and Technologies / M. A. Ertürk, M. A. Aydın, M. T. Büyükakkaşlar, H. Evirgen., 2019. – 216 с. – (Futur. Internet). – (11; 10).
58. Activity Recognition API [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/location-context/activity-recognition/>.
59. Android Open Source Project [Електронний ресурс] – Режим доступу до ресурсу: <https://source.android.com/devices/sensors/>.
60. Arduino [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Arduino>.
61. Arduino Due [Електронний ресурс] – Режим доступу до ресурсу: <https://store.arduino.cc/arduino-due>.
62. Awang Nor S. Simulated performance of TCP, SCTP, DCCP and UDP protocols over 4G network / S. Awang Nor, R. Alubady, W. Abduladeem Kamil. // Procedia Comput. Sci.. – 2017. – №111. – С. 2–7.
63. Baheti R. Cyber-physical systems / R. Baheti, H. Gill. // The Impact of Control Technology. – 2011. – С. 161–166.
64. Betters E. Apple HomeKit explained: What is it, and how does it work? [Електронний ресурс] / Betters – Режим доступу до ресурсу: <http://www.pocket-lint.com/news/129922-apple-homekit-explained-what-is-it-and-how-does-it-work>.
65. Chopard B. Cellular automata and lattice Boltzmann techniques: an approach to model and simulate complex systems / B. Chopard, P. Luthi, A. Masselot. – Geneva: Computer Science Department, University of Geneva, 2001. – P. 137
66. Cisco: Первый Всемирный форум по Интернету вещей [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: http://ip-news.ru/?cat=c_news&key=4441.
67. Corti – Artificial Intelligence that saves lives [Електронний ресурс] – Режим доступу до ресурсу: <https://corti.ai>.
68. Cyber-physical systems – a concept map [Електронний ресурс] – Режим доступу до ресурсу: <http://cyberphysicalsystems.org/>.
69. Data Processing for Emergency Monitoring by Mobile Devices / A. Shevchenko, O. Bychkov, V. Shevchenko, D. Berestov. // 13-th International Conference Perspective Technologies and Methods in MEMS Design. – 2017. – С. 124–128.
70. Detecting Falls Using Accelerometers by Adaptive Thresholds in Mobile Devices [Електронний ресурс] – Режим доступу до ресурсу: <http://www.jcomputers.us/vol9/jcp0907-07.pdf>.
71. Estimation of noise hazards in environmental monitoring tools design in the subway / V. Petrivskiy, V. Shevchenko, O. Bychkov, M. Brazhenenko. // 15th International Conference on the Experience of Designing and Application of CAD Systems. – 2019
72. Evans D. How the Internet of Everything Will Change the World...for the Better #IoE [Електронний ресурс] / Dave Evans // Blog of Dave Evans. – 2012. –

- Режим доступу до ресурсу: <https://blogs.cisco.com/digital/how-the-internet-of-everything-will-change-the-worldfor-the-better-infographic>.
73. Evans D. Internet of Everything in Action: Today and Tomorrow #IoE [Електронний ресурс] / Dave Evans // Blog of Dave Evans. – 2012. – Режим доступу до ресурсу: <https://blogs.cisco.com/digital/internet-of-everything-in-action>.
 74. Evans D. Internet of Everything изменит мир к лучшему [Електронний ресурс] / Dave Evans // Blog of Dave Evans – Режим доступу до ресурсу: <https://www.g-news.com.ua/news/10-it-/13936-internet-of-everything-izmenit-mir-k-luchshemu.html>.
 75. Evans D. Internet of Everything: Harnessing an Exponentially More Powerful Internet #IoE [Електронний ресурс] / Dave Evans // Blog of Dave Evans. – 2012. – Режим доступу до ресурсу: <https://blogs.cisco.com/digital/internet-of-everything-harnessing-an-exponentially-more-powerful-internet-ioe-infographic>.
 76. Evans D. Thanks to IoE, the Next Decade Looks Positively ‘Nutty’ [Електронний ресурс] / Dave Evans // Blog of Dave Evans. – 2013. – Режим доступу до ресурсу: <https://blogs.cisco.com/digital/thanks-to-ioe-the-next-decade-looks-positively-nutty>.
 77. Evans D. The Future of Wearable Technology: Smaller, Cheaper, Faster, and Truly Personal Computing [Електронний ресурс] / Dave Evans // Blog of Dave Evans. – 2013. – Режим доступу до ресурсу: <https://blogs.cisco.com/digital/the-future-of-wearable-technology-smaller-cheaper-faster-and-truly-personal-computing>.
 78. Fowler M. Microservices [Електронний ресурс] / Martin Fowler – Режим доступу до ресурсу: <http://martinfowler.com/articles/microservices.html>.
 79. Gershon R. R. Health and safety hazards associated with subways: a review / R. R. Gershon, K. A. Qureshi, M. A. Barrera., 2011. – 10-20 с.
 80. Google integrates Nest Learning Thermostat with iOS, Android [Електронний ресурс] – Режим доступу до ресурсу: <http://www.firstpost.com/tech/news-analysis/google-integrates-nest-learning-thermostat-with-ios-android-3660291.html>.
 81. Gorlatova M. Networking lowpower energy harvesting devices: Measurements and algorithms / M. Gorlatova, A. Wallwater, G. Zussman., 2013.
 82. Gson API [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/google/gson>.
 83. Hand, belt, pocket or bag: Practical activity tracking with mobile phones [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3972377/>.
 84. ICD-10. International Statistical Classification of Diseases and Related Health Problems. Tenth Revision. Vol.1, 1992. – 698 с. – (Geneva: World Health Organization)

85. Image Quality Assessment: From Error Visibility to Structural Similarity / Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli. // IEEE Trans. Image Process. – 2004. – №13. – С. 600–612
86. ImageMagic JP2 вхідні параметри [Електронний ресурс] – Режим доступу до ресурсу: <https://imagemagick.org/script/jp2.php>.
87. Jebril A. Overcoming Limitations of LoRa Physical Layer in Image Transmission / A. Jebril, A. Sali, M. Rasid. // Sensors. – 2018. – №10. – С. 32–57.
88. Jeschke S. Cyber-physical systems: History, presence, and future [Електронний ресурс] / Jeschke // Industrial Advisory Board, Aachen. – 2013. – Режим доступу до ресурсу: http://www.ima-zlw-ifu.rwth-aachen.de/fileadmin/user_upload/INSTITUTSCLUSTER/Publikation_Medien/Vortraege/download//CPS_27Feb2013.pdf.
89. Jolie coog buildings: Internet of Things for smart building infrastructure supporting concurrent apps utilizing distibuted microservices / [К. М. Gusmanov, К. R. Khanda, D. B. Salikhov та ін.], 2016. – 200 - 207 с. – (Современные информационные технологии и ИТ-образование).
90. Lavric A. LoRa (Long-Range) High-Density Sensors for Internet of Things / Lavric. // J.Sensors. – 2019. – С. 1–9.
91. Lee E. A. Introduction to embedded systems: A cyber-physical systems approach / E. A. Lee, S. Seshia., 2013. – 516 с. – (Lulu.com).
92. Lowe D. G. Distinctive Image Features from Scale-Invariant Keypoints / Lowe. // Int. J. Comput. Vis.. – 2004. – №2. – С. 91–110.
93. M2M [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/M2M>.
94. Mathematical Preliminaries / [D. Baowan, B. J. Cox, T. A. Hilder та ін.] // Modelling and Mechanics of Carbon-Based Nanostructured Materials / [D. Baowan, B. J. Cox, T. A. Hilder та ін.], 2017. – С. 35–58.
95. Mikhaylov K. On LoRaWAN scalability: Empirical evaluation of susceptibility to inter-network interference / K. Mikhaylov, J. Petajajarvi, J. Janhunen., 2017. – 1-6 с. – (European Conference on Networks and Communications).
96. Namiot D. On IoT Programming / D. Namiot, M. Sneps-Sneppe. // International Journal of Open Information Technologies. – 2014. – №10. – С. 25–28.
97. Node.JS [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Node.js>.
98. Noreen U. A study of LoRa low power and wide area network technology / U. Noreen, A. Bounceur, L. Clavier. // 2017 International Conference on Advanced Technologies for Signal and Image Processing. – 2017. – С. 1–6.
99. O'Toole J. Google buys Nest Labs for \$3.2 billion [Електронний ресурс] / James O'Toole – Режим доступу до ресурсу: <http://money.cnn.com/2014/01/13/technology/google-nest/index.html>.
100. OWASP Attack Category [Електронний ресурс] – Режим доступу до ресурсу: <https://www.owasp.org/index.php/Category:Attack>.

101. Performance Comparison between TCP and UDP Protocols in Different Simulation Scenarios / [F. AL-Dhief, N. Sabri, N. Abdul Latif та ін.]. // Int. J. Eng. Technol.. – 2018. – №7. – С. 172–176.
102. Pilot Survey of Subway and Bus Stop Noise Levels / R. R.Gershon, R. Neitzel, M. A. Barrera, M. Akram. – New York, 2012. – (Department of Sociomedical Sciences, Columbia University, Mailman School of Public Health.).
103. Piper B. AWS Certified Solutions Architect Study Guide / B. Piper, D. Clinton., 2020.
104. Prospero M. Best Smart Home Hubs of 2018 [Електронний ресурс] / Mike Prospero – Режим доступу до ресурсу: <https://www.tomsguide.com/us/best-smart-home-hubs,review-3200.html>..
105. Raspberry Pi 4 [Електронний ресурс]. Режим доступу (https://uk.wikipedia.org/wiki/Raspberry_Pi)
106. Raza U. Low Power Wide Area Networks: An Overview / U. Raza, P. Kulkarni, M. Sooriyabandara. // IEEE Commun. Surv. Tutorials. – 2017. – №19. – С. 855–873.
107. React [Електронний ресурс]. Режим доступу: (<https://uk.wikipedia.org/wiki/React>)
108. Shevchenko A. Dynamic Objects Emergency State Monitoring by Means of Smartphone Dynamic Data / A. Shevchenko, O. Bychkov, V. Shevchenko. // 14-th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics. – 2017. – С. 292–294.
109. Shrivastava S. Solutions Architect's Handbook / S. Shrivastava, N. Srivastav, K. Arora. – 2020.
110. Smart Homes and Home Automation [Електронний ресурс] // M2M Research Series from Berg Insight. – Режим доступу до ресурсу: <http://www.berginsight.com/ReportPDF/ProductSheet/bi-sh2-ps.pdf>..
111. Sullivan D. Official Google Cloud Certified Associate Cloud Engineer / Dan Sullivan., 2019
112. Taherizadeh S. A Capillary Computing Architecture for Dynamic Internet of Things: Orchestration of Microservices from Edge Devices to Fog and Cloud Providers. Sensors / S. Taherizadeh, V. Stankovski, M. Grobelnik. // Basel. – 2018.
113. The Gmail API [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/gmail/api/>.
114. The Network Effect Isn't Good Enough [Електронний ресурс] – Режим доступу до ресурсу: <http://techcrunch.com/2012/11/04/the-network-effect-isnt-good-enough/>.
115. The Smart Home Intelligent Home Automation. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.entrepreneurial-insights.com/smart-home-intelligent-home-automation..>
116. The Things Network 2016 Update [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <https://speakerdeck.com/wienke/the-things-network-2016-update>.

117. Transfer of Multimedia Data via LoRa / [R. Kirichek, V. D. Pham, M. Al-Bahri та ін.]. // Springer. – 2017. – С. 708–720.
118. Usevitch B. E. JPEG2000 Image Coding Standard / Usevitch // Wiley Encyclopedia of Telecommunications / Usevitch. – Hoboken, NJ, USA: John Wiley & Sons, Inc, 2003.
119. Wang Y. Survey on mobile social networking in proximity (MSNP): approaches, challenges and architecture / Wang. // Wireless networks. – 2014. – №6. – С. 1295–1311.
120. WHO Historical Collection. Disease classifications and nomenclature documents. [Електронний ресурс] / 2018. – Режим доступу: <http://www.who.int/library/collections/historical/en/index1.html>
121. Wolf-Gladrow D. A. Lattice-gas cellular automata and lattice Boltzmann models: an introduction [Електронний ресурс] / Wolf-Gladrow. – 2000. – Режим доступу до ресурсу: <http://openlb.org/>.
122. Wortmann F. Internet of Things / F. Wortmann, K. Flüchter. // Business & Information Systems Engineering. – 2015. – №57. – С. 221–224.
123. ZXing API [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/zxing/zxing>.