

Шафоренко С.Ю.

<https://orcid.org/0009-0009-6677-8548>

Приватний заклад вищої освіти «Харківський технологічний університет «ШАГ»

Аврята А.В.

<https://orcid.org/0000-0002-5100-2298>

Приватний заклад вищої освіти «Харківський технологічний університет «ШАГ»

ВИБІР АРХІТЕКТУРИ ДЛЯ СИСТЕМ РЕАЛЬНОГО ЧАСУ: КРИТЕРІЇ ТА МЕТОДОЛОГІЇ

В дослідженні описані особливості архітектурного проектування систем реального часу (СРЧ) з визначенням переваг та обмежень архітектурних підходів. Проведений аналіз останніх досліджень і публікацій дозволив сформулювати чітку та логічну структуру цього дослідження, що охоплює: особливості архітектурного проектування з впливом архітектурних рішень на характеристики СРЧ; архітектурні підходи щодо побудови СРЧ; критерії вибору архітектури СРЧ та практичне застосування на прикладі Інтернет речей та інтелектуальних систем, штучного інтелекту, систем прогнозування та вбудованих промислових систем керування; запропоновану методологію вибору архітектури СРЧ.

Систематизовано та обґрунтовано критерії вибору архітектури СРЧ, таких як детермінованість виконання, часові затримки, масштабованість, ефективність використання апаратних ресурсів, надійність і стійкість до відмов, складність розробки та супроводу, а також відповідність типу системи реального часу. На основі цих критеріїв наведено приклади практичного застосування різних архітектурних підходів у відеоспостереженні, інтелектуальній аналітиці поточкових даних, вбудованих промислових системах та системах Інтернету речей реального часу.

Запропоновано поетапну методологію вибору архітектури систем реального часу, яка включає формалізацію часових і нефункціональних вимог, визначення пріоритетних критеріїв, попередній відбір архітектурних альтернатив, їх порівняльну оцінку та верифікацію обраного рішення шляхом моделювання або прототипування. Застосування запропонованої методології дозволяє знизити ризики порушення часових вимог, підвищити обґрунтованість проектних рішень і забезпечити відповідність архітектури специфіці сучасних систем реального часу.

Отримані результати можуть бути використані при проектуванні та модернізації розподілених, інтелектуальних і вбудованих систем реального часу, а також слугувати основою для подальших досліджень у галузі архітектурного проектування СРЧ.

Ключові слова: системи реального часу, архітектура програмного забезпечення, архітектурне проектування, критерії, архітектурні підходи, детермінованість, часові характеристики.

Постановка проблеми. Сучасні можливості цифровізації та розвиток вбудованих систем реального часу (СРЧ) відіграють важливе значення для забезпечення функціонування критично важливих процесів. Такі системи набули широкого застосування в промисловій автоматизації, транспортних і енергетичних комплексах, медичному обладнанні, телекомунікаціях, а також у розподілених інформаційно-керуючих системах [1]. Специфіка застосування СРЧ повинна гаран-

тувати не лише правильність обчислювальних результатів, але й також їх отримання у визначені часові межі, що супроводжується ускладненням процесів проектування та реалізації. У випадках, коли порушуються часові обмеження у СРЧ це призводить до зниження якості функціонування, а в окремих ситуаціях і до аварійних ситуацій [2].

Одним з визначальних чинників, який впливає на здатність СРЧ виконувати задані часові обмеження є вибір архітектури, на основі якої мож-



ливо визначити ефективність та надійність роботи СРЧ. По-перше, архітектурні рішення визначають організацію програмних і апаратних компонентів, характер їхньої взаємодії, механізми обробки подій, синхронізації та розподілу ресурсів. По-друге, архітектурні рішення здатні впливати на детермінованість системи, її масштабованість, стійкість на відмову, складність розробки або супроводу тощо. У випадках, коли вибір архітектури є невдалим це може призвести до зростання затримок, втрати детермінованості виконання, ускладнення забезпечення стійкості в сценаріях відмови, а також до підвищення складності розробки й супроводу системи [3].

З практичної точки зору існує безліч архітектурних підходів, які можуть застосовуватися в СРЧ, наприклад, монолітна, шарова, клієнт-серверна, орієнтована на події, сервіс-орієнтована та мікросервісна архітектури. Кожен із цих підходів має певні переваги та обмеження щодо детермінованості, масштабованості, надійності та чутливості до затримок. Водночас відсутність універсальної архітектури, яка була б однаково ефективною для всіх типів СРЧ зумовлює необхідність індивідуального підходу до вибору архітектурних рішень [1-3].

Додатковою складністю є те, що вимоги до СРЧ часто мають суперечливий характер. Наприклад, підвищення масштабованості або гнучкості архітектури може негативно впливати на передбачуваність часових характеристик, тоді як прагнення до максимальної детермінованості може обмежувати можливості розвитку системи. У таких умовах процес вибору архітектури потребує чіткого формулювання критеріїв оцінювання та застосування методологічних підходів, які дозволяють врахувати як функціональні, так і нефункціональні вимоги [4].

Актуальною науково-практичною проблемою цього дослідження є обґрунтування критеріїв та методології вибору архітектури для систем реального часу, що дозволить підвищити ефективність проектування, забезпечити дотримання часових обмежень та зменшити ризики, пов'язані з експлуатацією таких систем.

Аналіз останніх досліджень і публікацій. Сучасні дослідження в галузі СРЧ охоплюють безліч важливих питань, які взаємопов'язані з широким спектром наукових напрямів. Деякі дослідження спрямовані на аналіз архітектурних підходів та часових характеристик СРЧ [5], проектування архітектури програмного забезпечення (ПЗ) систем керування реальним часом [6], роз-

робку застосунків СРЧ на основі монолітної та мікросервісної архітектури, а також виявлення переваг та недоліків [7, 8], дослідження моделей з об'єднанням мережних та обчислювальних ресурсів для СРЧ на основі сервіс-орієнтованої архітектури [9] тощо. Не менш важливими є сучасні підходи, які пов'язані з інтеграцією СРЧ у поєднанні з розширеними та розподіленими системами (Edge, Fog, Cloud) [10], методами штучного інтелекту (ШІ) [11-12], Інтернет речей [13] та операційними системами [14].

В одному з нещодавніх досліджень [5] запропоновано методи визначення часових характеристик СРЧ з використанням моделей і формалізацій на етапі проектування системи. Метод враховує послідовність появи задач, їхні типи та залежності при визначенні часових характеристик, що підсилює можливості моделювання передбачуваності виконання задач у СРЧ. Це свідчить про зростаючу увагу дослідників до кількісного аналізу часових параметрів ще на ранніх стадіях проектування архітектури СРЧ.

Іншим не менш важливим аспектом є проектування архітектури ПЗ систем керування реальним часом. Етап проектування є важливим, оскільки саме на цьому етапі враховується складність таких систем з одного боку, та нагальний попит на високоякісні системи в промисловості з іншого боку. Головним завданням на етапі проектування є створення архітектури ПЗ, яка описує можливу реалізацію програми з дотриманням як функціональних, так і нефункціональних вимог, які пов'язані з часом. Авторами в роботі [6] досліджуються різні альтернативні варіанти проектування з використанням методів спроб та помилок, що може слугувати для створення прийнятної архітектури. З цією метою авторами запропоновано використання шаблонів реального часу, що може допомогти інженерам та програмістам при проектуванні визначати відповідну архітектуру за допомогою стратегії спроб і помилок. Така концепція дозволяє застосовувати запропоновані шаблони щодо реального випадку і можуть бути корисними для різних проектів ПЗ СРЧ.

В публікації [7] наголошується на тому, що розробка застосунків СРЧ обумовлена зростанням інтересу використання мікросервісної архітектури, яка є альтернативним варіантом замість монолітної архітектури. Це пов'язано з тим, що СРЧ характеризуються підвищеними вимогами до часових показників обробки запитів, що ускладнює використання монолітної архітектури через обмежену масштабованість і конфліктне вико-

ристання спільних апаратних ресурсів. У таких умовах мікросервісний підхід розглядається як перспективний засіб підвищення продуктивності та гнучкості систем, оскільки дозволяє ізолювати функціональні компоненти, масштабувати їх незалежно та зменшувати взаємний вплив навантажень. При цьому підкреслюється доцільність застосування мікросервісної архітектури для веб-орієнтованих систем реального часу, зокрема інформаційних сервісів, чутливих до затримок взаємодії з користувачем. Водночас з цим у роботі [8] авторами розглядаються обмеження мікросервісної архітектури, які пов'язані зі зростанням складності взаємодії між сервісами, підвищеними вимогами до інфраструктури, а також труднощами налагодження та забезпечення безпеки в СРЧ.

У публікації [9] сервіс-орієнтована архітектура розглядається для побудови розподілених систем із слабо зв'язаними сервісами та стандартизованими інтерфейсами. У контексті систем реального часу така архітектура здатна забезпечувати гнучкість і масштабованість, однак супроводжується додатковими затримками, що ускладнює гарантування детермінованості. Тому її застосування є доцільним переважно для м'яких СРЧ, де часові обмеження мають допустимий діапазон.

В сучасних дослідженнях та публікаціях розглядаються різні архітектурні вимоги з різних наукових напрямів. Наприклад, в роботі [10] розглядаються розширені та розподілені системи (Edge, Fog, Cloud) при побудові складних СРЧ за допомогою відеоспостереження, де часові обмеження поєднуються з вимогами до обробки великих даних та комунікаційної взаємодії між компонентами. Підтвердженням тому у роботі [11] розглядається вплив інтеграції методів ШІ на архітектуру великих даних з підтримкою обробки в реальному часі, де автором наголошено, що використання масштабованих розподілених архітектур, зокрема хмарних та периферійних обчислень, є критичним чинником для забезпечення реального часу обробки потокових даних. Водночас у дослідженні зазначаються обмеження, пов'язані з якістю даних, масштабованістю моделей ШІ та складністю гарантування передбачуваних часових характеристик, що є важливим для систем реального часу.

Враховуючи архітектурні особливості, фундаментальні основи проектування та практичні аспекти реалізації деякі з досліджень охоплюють ретельний аналіз СРЧ на базі ШІ. Огляд архітектурних підходів щодо інтеграції ШІ в СРЧ

з орієнтацією застосування для критично важливих галузей представлено в роботі [12]. Авторами проведена класифікація архітектури ШІ СРЧ на основі централізованих, периферійних та гібридних моделей, оцінюючи їхню придатність з огляду на часові обмеження та вимоги до надійності. Дослідження спрямоване на застосування методів ШІ, (машинне навчання, глибоке навчання та навчання з підкріпленням) у системах, які потребують відповідей з низькою затримкою та високою надійністю.

Розвиток технології Інтернет речей (ІР) не залишився поза межами цього дослідження, оскільки саме ця технологія почала широко впроваджуватися за останні роки у різні сфери обслуговування, які потребують вимог до СРЧ. В роботі [13] пропонується нова методологія проектування систем ІР СРЧ, де в якості критеріїв розглядається підтримка параметрів якості обслуговування (ЯО). Критерій ЯО розглядається з метою забезпечення доставки пакетів даних у найкоротші терміни, запобігаючи перевищенню критичних параметрів, що стосуються передачі в реальному часі. Методологія базується на запропонованій авторами моделі функціональної специфікації СРЧ та її відображенні в архітектурі ІР.

Проаналізувавши вищезазначені дослідження та публікації необхідно врахувати також те, що більшість галузей промисловості покладаються на вбудовані додатки на основі операційних систем реального часу (ОСРЧ), які забезпечують детерміновану поведінку та точний час у найважливіших системах. У роботі [14] описуються концепції ОСРЧ, основні компоненти зі шаблонами проектування та найкращі практики для оптимізації та тестування.

Проте в огляді літератури також трапляються значущі прогалини. В існуючих дослідженнях все менше пропонуються комплексні методології вибору архітектури, які враховують одночасно часові вимоги, надійність, масштабованість та особливості апаратної платформи. Найчастіше архітектурні підходи розглядаються окремо, без узагальненої процедури порівняння або формалізованих критеріїв для обґрунтованого вибору. Це створює необхідність для систематизації критеріїв та розробки методології, що дозволять інтегрувати результати різних напрямів досліджень в єдину процедуру прийняття архітектурних рішень для СРЧ, що є предметом цього дослідження.

Постановка завдання. Метою дослідження є обґрунтування критеріїв та розробка методоло-

гії вибору архітектури для систем реального часу з урахуванням часових, функціональних і нефункціональних вимог, а також особливостей сучасних вбудованих, розподілених та інтелектуальних систем. Для досягнення поставленої мети цього дослідження необхідно сконцентрувати увагу на виконання таких завдань:

- проаналізувати роль та особливості архітектурного проектування СРЧ з визначенням їхнього впливу на часові характеристики;
- охарактеризувати основні архітектурні підходи, які можуть застосовуватися для побудови СРЧ з визначенням їхніх переваг та обмежень;
- систематизувати критерії вибору архітектури СРЧ з врахуванням детермінованості, масштабованості, надійності та ресурсних обмежень;
- на основі архітектурних підходів та критеріїв вибору архітектури СРЧ проаналізувати практичні приклади застосування архітектурних підходів у різних класах СРЧ;
- сформулювати поетапну методологію вибору архітектури СРЧ та обґрунтувати її практичну доцільність.

Виклад основного матеріалу. На основі проведеного аналізу останніх досліджень і публікацій в роботі наведена логічна структура, яка охоплює особливості архітектурного проектування з метою виявлення впливу архітектурних рішень на характеристики СРЧ, архітектурні підходи щодо побудови СРЧ, критерії вибору архітектури СРЧ з практичними прикладами застосування, а наприкінці роботи описується методологія вибору архітектури для СРЧ. Така структура дозволить краще зрозуміти, як архітектурні рішення, підходи та критерії впливають на роботу СРЧ.

1. Особливості архітектурного проектування та вплив архітектурних рішень на характеристики СРЧ.

Архітектурне проектування СРЧ містить безліч принципів особливостей, що відрізняється від проектування традиційних інформаційних або бізнес-орієнтованих систем. Характерною відмінністю є домінування часових обмежень над іншими вимогами, оскільки архітектура СРЧ повинна гарантувати виконання критичних функцій у визначені часові інтервали за всіх допустимих умов експлуатації [3]. Забезпечення належного рівня поведінки системи є однією з головних практичних задач архітектурного проектування. Це свідчить про те, що для кожного сценарію роботи необхідне визначення верхньої межі часу для виконання операцій, наприклад, час виконання за найгіршого сценарію (Worst-Case Execution

Time, WCET). З огляду на це архітектура повинна забезпечувати мінімізацію з використання недедетермінованих механізмів (наприклад, динамічне виділення пам'яті, неконтрольовані черги повідомлень або складні ланцюги асинхронних викликів).

На практиці проектування СРЧ показує, що архітектурні рішення безпосередньо впливають на можливість аналізу часових характеристик. Наприклад, монолітні або модульні архітектурні з фіксованими інтерфейсами значно спрощують статичний часовий аналіз, тоді як розподілені або сервіс-орієнтовані архітектури можуть ускладнювати прогнозування затримок через мережеві взаємодії [2, 4].

Ще однією важливою особливістю є тісний зв'язок архітектури з механізмами планування задач. Архітектурне планування повинно враховувати модель виконання (періодичні, аперіодичні та спорадичні задачі), механізми пріоритету, а також підтримку реального часу з боку операційної системи. Невідповідність архітектури обраній моделі планування може призвести до неконтрольованих затримок і порушення дедлайнів.

З практичної точки зору важливим аспектом є ізоляція компонентів і локалізація відмов. Архітектура СРЧ повинна забезпечувати таку організацію компонентів, за якої збій у некритичному модулі не впливає на виконання часоно критичних функцій. Це досягається шляхом використання ізольованих процесів, чітко визначених інтерфейсів взаємодії та механізмів деградації функціональності.

Крім того, архітектурне проектування СРЧ відбувається в умовах обмежених апаратних ресурсів. Тому архітектурні рішення повинні бути узгоджені з можливостями апаратної платформи, зокрема з кількістю обчислювальних ядер, обсягом пам'яті та характеристиками систем вводу-виводу [1]. Надмірно складна архітектура може призвести до перевищення допустимих часових і ресурсних обмежень навіть за правильно реалізованих алгоритмів.

Для систематизації практичних вимог до архітектури СРЧ доцільно зіставити архітектурні рішення з їх впливом на ключові властивості СРЧ (табл. 1).

Проведене узагальнення показує, що архітектурне проектування СРЧ має виражений компромісний характер. Забезпечення високої детермінованості та мінімальних затримок, як правило, досягається за рахунок зменшення гнучкості та обмеження можливостей масштабування системи.

Вплив архітектурних рішень на характеристики СРЧ

Архітектурний аспект	Практичне значення для СРЧ	Потенційні ризики
Монолітна або модульна структура	Мінімальні затримки, висока детермінованість	Обмежена масштабованість
Чітко визначені інтерфейси компонентів	Спрощена аналізу WCET та тестування	Зростання складності проектування
Пріоритетна модель виконання	Гарантоване виконання критичних задач	Ризик пріоритетної інверсії
Ізоляція компонентів	Локалізація відмов, підвищення надійності	Додаткові накладні витрати
Обмеження асинхронних взаємодій	Передбачуваність часових характеристик	Зменшення гнучкості системи
Узгодження з апаратною платформою	Оптимальне використання ресурсів	Обмеження на розвиток системи

Джерело: сформовано авторами

Натомість архітектурні рішення, які орієнтовані на модульність і розширюваність потребують додаткових механізмів контролю часових характеристик, що ускладнює процес проектування.

Архітектурні підходи щодо побудови СРЧ.

Архітектура СРЧ визначає принципи організації програмних і апаратних компонентів, характер їхньої взаємодії, механізми обробки подій та розподілу обчислювальних ресурсів, що впливає на здатність системи дотримуватися заданих часових обмежень. Вибір архітектурного підходу є одним із ключових етапів проектування СРЧ, оскільки саме на цьому рівні закладаються властивості детермінованості, передбачуваності та стійкості системи до змін навантаження і відмов. На практиці розробка СРЧ вимагає архітектурних підходів, кожен з яких має власні переваги та обмеження з погляду часових характеристик, масштабованості та складності реалізації. Розглянемо деякі архітектурні підходи щодо побудови СРЧ (рис. 1), які найчастіше використовуються в інженерній практиці та проаналізуємо їхню придатність для різних класів СРЧ [1-4].

Монолітна архітектура передбачає реалізацію основної функціональності СРЧ у вигляді єдиного програмного модуля або тісно пов'язаного набору модулів, що виконуються в одному адресному просторі. Взаємодія між компонентами здійснюється через прямі виклики функцій без використання проміжних шарів абстракції [3]. З точки зору практичності монолітна архітектура забезпечує мінімальні накладні витрати на обмін даними та високу передбачуваність часових характеристик, що дозволяє ефективно виконувати аналіз гіршого часу виконання (WCET) та гарантує дотримання жорстких дедлайнів. Архітектура набула широкого застосування у жорстких СРЧ

(в програмованих логічних контролерах, системах керування технологічними процесами та вбудованих пристроях). До обмежень архітектури можна віднести низьку гнучкість та складність масштабування через зміни в системі, які можуть вимагати модифікації та повторного тестування значної частини коду, що ускладнює розвиток системи протягом життєвого циклу [8].

Шарова архітектура на відміну від монолітної архітектури відрізняється розподілом системи на функціонуючі рівні (апаратний та системний рівні, рівень сервісів, прикладний рівень), кожен з яких виконує чітко визначені завдання. До переваг архітектури можна віднести підвищення структурованості системи та спрощення супроводу, що дозволяє в контексті СРЧ відокремити часові критичні компоненти для проведення аналізу та тестування. На противагу цьому кожен додатковий шар архітектури створює накладні витрати, що в свою чергу негативно впливає на часові характеристики. Відповідно така архітектура застосовується обмежено, що вимагає суворого контролю кількості рівнів та механізмів взаємодії між шарами [2].

Архітектура орієнтована на події (подієво-орієнтована архітектура) ґрунтується на обробці подій, що надходять з зовнішнього середовища або генеруються всередині системи. Реакція системи супроводжується ініціюванням настання події, а не послідовним виконанням заздалегідь визначеного сценарію. Такий підхід добре узгоджується з СРЧ, які працюють з датчиками, перериваннями та асинхронними сигналами. Така архітектура забезпечує високу реактивність і дозволяє ефективно обробляти велику кількість подій. Разом з тим асинхронний характер виконання ускладнює аналіз часової поведінки сис-

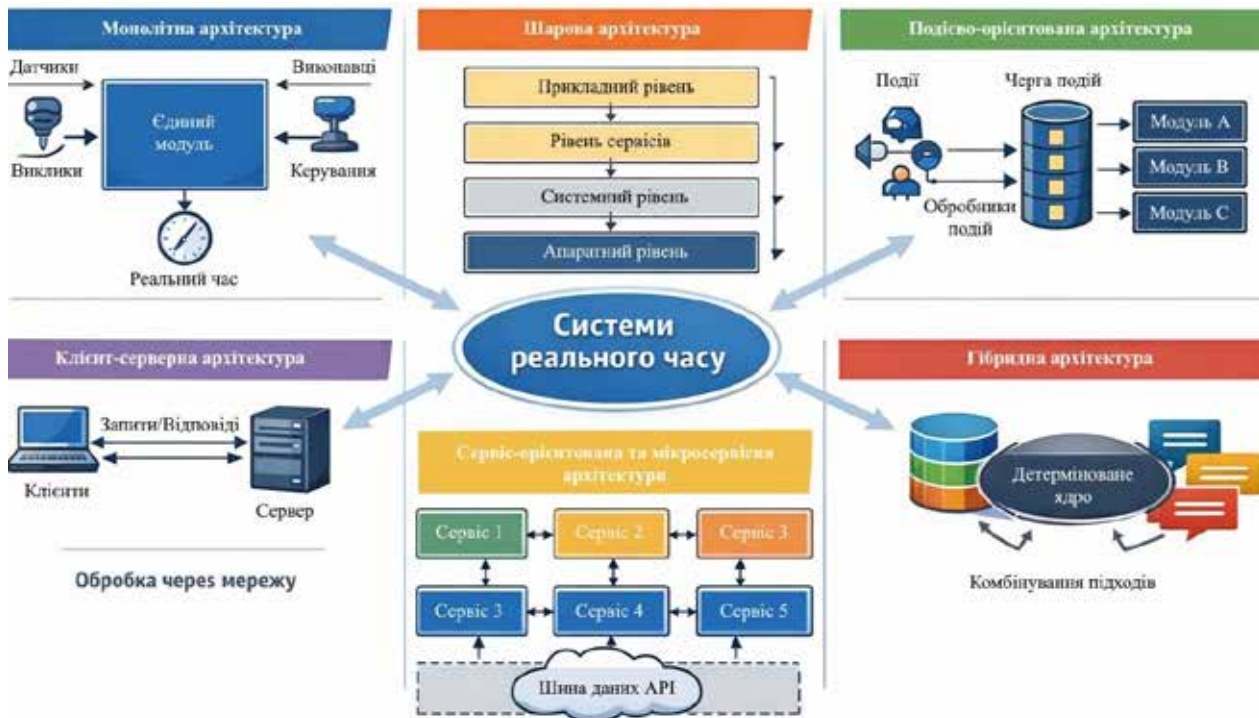


Рис. 1. Архітектурні підходи щодо побудови СРЧ

Джерело: сформовано авторами

теми. Без додаткових механізмів контролю черг подій і пріоритетів існує ризик непередбачуваних затримок, що обмежує застосування цього підходу в жорстких СРЧ [15].

Клієнт-серверна архітектура розрахована на розподіл функцій між клієнтськими компонентами та сервером, який виконує основну обробку даних або керування. Обмін між компонентами здійснюється через мережу або інші комунікаційні канали. У СРЧ цей підхід застосовується переважно в розподілених або мережових СРЧ, наприклад у диспетчерських системах або системах моніторингу, який дозволяє централізувати управління та спростити інтеграцію з іншими системами [1-2]. Обмеженням архітектури висувається залежність через затримки передачі даних і стабільність каналів зв'язку, що ускладнює забезпечення жорстких часових обмежень і обмежує застосування такого підходу в критично важливих СРЧ.

Сервіс-орієнтована та мікросервісна архітектури базуються на декомпозиції системи на незалежні сервіси, які взаємодіють між собою через стандартизовані інтерфейси. Основною перевагою є висока гнучкість, масштабованість і можливість незалежного розвитку компонентів. У контексті СРЧ такі архітектури мають обмежене застосування через значні накладні витрати на взаємодію між сервісами та складність забез-

печення детермінованості. Зазвичай вони використовуються для некритичних підсистем, таких як аналітика, візуалізація або інтеграція з зовнішніми інформаційними сервісами [8, 16].

Гібридні архітектури поєднують кілька архітектурних підходів у межах однієї системи. Найпоширенішим практичним рішенням є виділення детермінованого ядра системи, побудованого за монолітним або подієво-орієнтованим принципом, та використання більш гнучких архітектур для допоміжних компонентів. Такий підхід дозволяє збалансувати вимоги до детермінованості, надійності та масштабованості, що робить його перспективним для складних систем реального часу з тривалим життєвим циклом [2, 4].

Критерії вибору архітектури СРЧ та їхнє практичне застосування.

Вибір архітектури СРЧ є багатокритеріальним завданням, оскільки архітектурні рішення частковою або повною мірою здатні впливати не лише на функціональні можливості системи, але й на здатність забезпечувати задані часові обмеження в умовах змінного навантаження, а також обмежених апаратних ресурсів [2, 8]. Відповідно з чим постає питання правильного архітектури, що повинно базуватися на чітко сформульованій системі критеріїв, на основі яких можна об'єктивно оцінити доцільність застосування того або іншого архітектурного підходу.

Розглянемо табл. 2, в якій будуть наведені основні критерії, які впливають на вибір архітектури СРЧ.

На основі проведеного аналізу останніх досліджень та врахувавши вищезазначені критерії розглянемо практичні аспекти з використанням критеріїв для різних галузей. В табл. 2 наведені архітектурні підходи для деяких прикладів практичного застосування.

Методологія вибору архітектури для СРЧ.

На основі проведеного аналізу архітектурних підходів та критеріїв оцінювання доцільно застосовувати поетапну методологію вибору архітектури СРЧ, яка забезпечує системність прийняття проєктних рішень, а також враховує функціональні та нефункціональні вимоги. Такий підхід дозволяє зменшити суб'єктивність вибору та підвищити обґрунтованість архітек-

Таблиця 2

Критерії вибору архітектури СРЧ

Критерій	Зміст та значення
Детермінованість	Базовий критерій для СРЧ, який характеризує здатність системи забезпечувати передбачуваний час реакції на зовнішні події. Архітектура повинна дозволити точно оцінювати гірші часові випадки виконання задач та мінімізувати джерела недетермінованої поведінки, (наприклад, неконтрольований доступ до спільних ресурсів, непередбачувані черги повідомлень або мережні затримки. Критерій охоплює жорсткі СРЧ, де порушення часових обмежень є неприпустимим.
Часові характеристики та затримки	Архітектура СРЧ повинна забезпечувати виконання задач у межах встановлених дедлайнів, включаючи обробки подій, передачу даних і взаємодію між компонентами. При оцінюванні цього критерію враховуються середні та максимальні затримки, а також вплив контекстних перемикачів і взаємодії між процесами. Критерій охоплює архітектуру з багаторівневою взаємодією або розподіленими компонентами, які потребують особливої уваги до аналізу часових накладних витрат.
Масштабованість	Визначає здатність архітектури адаптуватися до зростання навантаження або розширення функціональності без суттєвого погіршення часових характеристик. Для СРЧ важливо, щоб масштабування не призводило до втрати детермінованості або неконтрольованого зростання затримок. Критерій охоплює м'які СРЧ та розподілені системи з динамічною кількістю джерел подій.
Надійність і стійкість до відмов	Архітектура повинна забезпечувати безперервність функціонування системи в умовах часткових відмов компонентів або каналів зв'язку. Критерій надійності охоплює підтримку резервування, ізоляцію відмов, механізми відновлення та деградації функціональності без порушення часових вимог. Критерій охоплює СРЧ, що застосовуються у критично важливих галузях, який є не менш важливим, ніж детермінованість.
Ефективність використання апаратних ресурсів	СРЧ здебільшого функціонують в умовах обмежених обчислювальних ресурсів, зокрема у вбудованих системах. Архітектура повинна забезпечувати раціональний розподіл процесорного часу, пам'яті та пропускну здатності каналів зв'язку. Надмірна модульність або складні механізми взаємодії можуть призвести до перевитрат ресурсів та зниження продуктивності, що негативно впливає на часові характеристики.
Складність розробки та супроводу	Архітектурні рішення впливають на складність реалізації, тестування та подальшого супроводу СРЧ. Простота архітектури повинна сприяти зменшенню кількості помилок і полегшувати верифікацію часових властивостей, що може обмежувати гнучкість системи. З іншого боку, складні архітектури потребують високої кваліфікації розробників і збільшують витрати на супровід.
Гнучкість та можливість модернізації	Сучасні можливості СРЧ здебільшого можуть потребувати адаптації до змін вимог або умов експлуатації. Архітектура повинна забезпечувати можливість розширення або модифікації функціональності без повного перепроєктування системи. При цьому важливо, щоб внесення змін не призводило до порушення критичних часових обмежень, що може вимагати ретельного контролю впливу модифікацій на часові характеристики.
Відповідність типу СРЧ	Вибір архітектури має здійснюватися з урахуванням типу СРЧ (жорстка, м'яка або умовно жорстка). Для жорстких СРЧ пріоритетними є детермінованість і передбачуваність, тоді як для м'яких СРЧ допускається компроміс між часовими характеристиками та гнучкістю. Ігнорування цього критерію може призвести до невідповідності архітектури реальним вимогам системи.

Джерело: сформовано авторами на основі [2, 4, 8]

Приклади практичного застосування архітектурних підходів та критеріїв у СРЧ

Приклади практичного застосування	Архітектурний підхід	Ключові критерії вибору архітектури	Особливості забезпечення реального часу
Інтелектуальні системи відеоспостереження	Розподілена архітектура Edge–Fog–Cloud	Низька затримка, масштабованість, пропускна здатність, обробка великих даних	Попередня обробка відеопотоків на периферії (Edge) для зменшення затримок; агрегація та аналітика на рівні Fog; довгострокове зберігання та навчання моделей у хмарі
Аналітика поточкових даних з використанням ШІ	Хмарно-розподілена та гібридна архітектура	Масштабованість, підтримка обробки в реальному часі, гнучкість, продуктивність	Використання розподілених платформ потокової обробки; складність гарантування детермінованості через динамічне масштабування та варіативність навантаження
Критично важливі системи з елементами ШІ (автономні системи, робототехніка)	Периферійна або гібридна (Edge в поєднанні з централізованим контролем)	Детермінованість, надійність, мінімальні затримки, безпека	Перенесення логіки прийняття рішень ближче до джерел даних; обмеження складності моделей ШІ для збереження передбачуваних часових характеристик
Системи ІР реального часу	Багаторівнева архітектура ІР (пристрої, шлюзи, хмара)	ЯО, затримка, масштабованість, енергоефективність	Забезпечення гарантованої доставки пакетів даних у визначені часові межі; використання пріоритетів і політик ЯО для трафіку реального часу
Вбудовані промислові системи керування	Централізована або слабо розподілена архітектура на базі ОСРЧ	Детермінованість, ефективність використання ресурсів, надійність	Використання операційних систем реального часу для забезпечення точного часу, прогнозованого планування задач і контролю доступу до ресурсів
Системи прогнозного обслуговування	Гібридна архітектура (Edge в поєднанні з Cloud)	Баланс між затримкою та обчислювальною потужністю, масштабованість	Локальна обробка критичних сигналів у реальному часі; складні аналітичні моделі виконуються асинхронно у хмарному середовищі

Джерело: сформовано авторами

турних рішень у складних умовах реального застосування.

Першим етапом методології є визначення типу СРЧ (жорстка, м'яка або умовно жорстка) та формалізація часових вимог, зокрема граничних значень затримок, дедлайнів та вимог до передбачуваності виконання задач. Паралельно визначаються ключові нефункціональні вимоги, такі як надійність, масштабованість, безпека, енергоефективність та обмеження апаратної платформи. Чітка формалізація вимог на цьому етапі є критично важливою, оскільки саме вона визначає допустимий простір архітектурних рішень.

Другим етапом є ідентифікація та ранжування пріоритетних критеріїв вибору архітектури, які мають вирішальний вплив на функціонування СРЧ. Залежно від сфери застосування, такими критеріями можуть бути детермінованість виконання, мінімізація затримок, стійкість до відмов, підтримка розподіленої обробки або можливість

інтеграції методів ШІ. На цьому етапі доцільно визначити допустимі компроміси між окремими критеріями, зокрема між гнучкістю архітектури та передбачуваністю часових характеристик.

Третім етапом є попередній відбір архітектурних рішень, які потенційно здатні задовольнити сформульовані вимоги та обрані пріоритетні критерії. На цьому етапі розглядаються як класичні архітектурні підходи (монолітна, шарова, клієнт–серверна), так і сучасні розподілені або гібридні архітектури (подієво-орієнтована, сервіс-орієнтована, мікросервісна). Архітектури, які не відповідають критичним часовим виключаються з подальшого розгляду.

Наступним етапом є порівняльна оцінка відібраних архітектурних альтернатив за визначеними критеріями з урахуванням взаємних компромісів між детермінованістю, масштабованістю, гнучкістю та складністю реалізації. Оцінювання може здійснюватися із застосуванням експертних

методів, матриць критеріїв або формалізованих моделей аналізу часових характеристик. Особлива увага на цьому етапі приділяється аналізу джерел затримок, механізмів синхронізації та впливу взаємодії між компонентами на часову передбачуваність системи.

Завершальним етапом методології є прийняття архітектурного рішення та його верифікація шляхом моделювання, імітаційного аналізу або прототипування. Метою цього етапу є підтвердження здатності обраної архітектури забезпечувати дотримання часових обмежень у реальних або наближених до реальних умовах експлуатації. За результатами верифікації можуть бути внесені коригування до архітектури або повторно переглянуті окремі критерії вибору.

Застосування запропонованої поетапної методології вибору архітектури дозволяє систематизувати процес проектування СРЧ, знизити ризики порушення часових вимог, підвищити ефективність архітектурних рішень та забезпечити їх відповідність специфіці сучасних систем реального часу, зокрема розподілених, інтелектуальних та вбудованих систем.

Висновки. Проведене дослідження дало змогу з'ясувати низку можливостей, які пов'язані з архітектурними підходами, критеріями та методологією для СРЧ. Особливості архітектурного проектування дозволили визначити вплив архітектурних рішень на характеристики СРЧ, де було оцінено практичне значення та потенційні ризики

з архітектурними аспектами. Серед потенційних ризиків при проектуванні розглянуто обмежену масштабованість систем, зростання складності проектування, ризики пріоритетної інверсії, додаткові накладні витрати, зменшення гнучкості системи та обмеження розвитку систем.

Розглянуто архітектурні підходи щодо побудови СРЧ, де визначено, що монолітна архітектура є найбільш придатною для жорстких СРЧ, а сервіс-орієнтована, мікросервісна та розподілені архітектури доцільно застосовувати переважно для більш м'яких СРЧ. Визначено, що гібридні архітектури можуть бути перспективним напрямом для складних СРЧ, які можуть поєднувати детерміноване ядро з гнучкими допоміжними підсистемами.

Систематизовано критерії вибору архітектури СРЧ, серед яких визначальними є детермінованість, часові характеристики, надійність, масштабованість та ефективність використання апаратних ресурсів. На основі розглянутих архітектурних підходів та критеріїв запропонована поетапна методологія вибору архітектури для структурування процесу прийняття архітектурних рішень, що зменшує суб'єктивність вибору та підвищує ймовірність дотримання часових вимог у реальних умовах експлуатації. В якості перспективи майбутніх досліджень доцільним буде проведення аналізу можливостей архітектурних рішень СРЧ для розробки ПЗ на програмному рівні.

Список літератури:

1. Пасічник М. Ю. Методи диспетчеризації завдань в системах реального часу. *Computer-Integrated Technologies: Education, Science, Production*. 2024. № 56. С. 39–43. DOI: <https://doi.org/10.36910/6775-2524-0560-2024-56-04>
2. Журавель І. В., Сайківська Л. Ф. Особливості використання операційних систем реального часу у мікроконтролерних системах. *Радіоелектроніка та молодь у XXI столітті : матеріали 24-го Міжнар. молодіж. форуму (Харків, 7–9 квіт. 2020 р.)*. Харків : ХНУРЕ, 2020. Т. 3. С. 190–191. URL: <http://openarchive.nure.ua/handle/document/13854>
3. Чихіра І. В., Микитишин А. Г. Конспект лекцій з дисципліни «Програмування систем реального часу». Харків, 2016. С. 17–23. URL: <http://openarchive.nure.ua/handle/document/13854>
4. Проектування систем реального часу : конспект лекцій для студентів спец. 123 «Комп'ютерна інженерія» [Електронний ресурс] / уклад. С. О. Ковальов, В. В. Шамаєв. Луцьк : ДонНТУ, 2023. 85 с. URL: <http://ea.donntu.edu.ua/jspui/handle/123456789/34584>
5. Юрович І., Зайцев В. Метод визначення часових характеристик систем реального часу. *Управління розвитком складних систем*. 2024. № 59. С. 148–154. DOI: <https://doi.org/10.32347/2412-9933.2024.59.148-154>
6. Mzid R., Abid M. Software architecture design for real-time control systems: how design patterns could help? *Innovations in Systems and Software Engineering*. 2025. P. 1–13. DOI: <https://doi.org/10.1007/s11334-025-00600-w>
7. Здоренко Ю. М. Розробка програмних додатків реального часу на основі використання мікросервісної архітектури. Тези 76-ї наук. конф. проф., викл., наук. працівників, аспірантів та студентів ун-ту (Полтава, 14–23 трав. 2024 р.). Полтава : Нац. ун-т ім. Ю. Кондратюка, 2024. Т. 1. С. 466. URL: <https://reposit.nupp.edu.ua/handle/PolNTU/16607>

8. Киселевич В. В., Усата О. Ю., Сікора Я. Б., Вербівський Д. С., Іванов Д. Є. Мікросервісна архітектура: переваги та недоліки її практичного застосування. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2024. № 2. С. 50–59. DOI: <https://doi.org/10.32782/IT/2024-2-7>
9. Музичук І. В. Сервіс-орієнтована архітектура. Сучасні інфокомунікаційні технології : зб. тез наук.-техн. конф. (Київ, 5 груд. 2019 р.). Київ : ДУТ, 2019. С. 161–162.
10. Бабарика А., Басараб О., Площик А., Табенський С. Вибір архітектури побудови систем відеоспостереження Державної прикордонної служби України на основі хмарних та туманних технологій. 2024. DOI: <https://doi.org/10.58254/viti.5.2024.01.05>.
11. Li A. AI-Driven Big Data Analytics: Scalable Architectures and Real-Time Processing. *European Journal of AI, Computing & Informatics*. 2025. Vol. 1, No. 1. P. 33–41. DOI: <https://doi.org/10.71222/pw8kw891>.
12. Miya M., Islam M., Saleh M. A., Mia M. M. H., Faruk M. O. An overview of architectures, difficulties, and uses for artificial intelligence in real-time systems. *Scientia. Technology, Science and Society*. 2025. Vol. 2, No. 9. P. 41–50. DOI: [https://doi.org/10.59324/stss.2025.2\(9\).05](https://doi.org/10.59324/stss.2025.2(9).05).
13. Deniziak S., Plaza M., Arcab Ł. Approach for designing real-time IoT systems. *Electronics*. 2022. Vol. 11, No. 24. Art. 4120. DOI: <https://doi.org/10.3390/electronics11244120>.
14. Ahmad A. B., Prabowo S. Real-time operating systems for embedded applications: design and implementation. *Journal of Integrated VLSI, Embedded and Computing Technologies*. 2025. Vol. 2, No. 1. P. 37–45. DOI: <https://doi.org/10.31838/JIVCT/02.01.05>.
15. Event-Driven Architecture (EDA): a complete guide to real-time systems. CortexFlow. 2024. Medium. URL: <https://medium.com/the-software-frontier/event-driven-architecture-eda-a-complete-guide-to-real-time-systems-974f612dc6b5>
16. Товма О. Основні типи архітектури програмного забезпечення. *Art of Business Analysis*. 2024. URL: <https://www.artofba.com/uk/post/main-types-of-software-architecture>

Shaforenko S.Yu., Avriata A.V. CHOOSING AN ARCHITECTURE FOR REAL-TIME SYSTEMS: CRITERIA AND METHODOLOGIES

This study describes the features of real-time system (RTS) architectural design, identifying the advantages and limitations of architectural approaches. An analysis of recent studies and publications has made it possible to form a clear and logical structure for this study, covering: the features of architectural design with the influence of architectural solutions on the characteristics of RTS; architectural approaches to building RTS; criteria for selecting RTS architecture and practical application using the example of the Internet of Things and intelligent systems, artificial intelligence, forecasting systems, and embedded industrial control systems; the proposed methodology for selecting RTS architecture.

The criteria for selecting RTS architecture, such as determinism of execution, time delays, scalability, efficiency of hardware resource utilization, reliability and fault tolerance, complexity of development and maintenance, as well as compliance with the type of real-time system, are systematized and justified. Based on these criteria, examples of the practical application of various architectural approaches in video surveillance, intelligent stream data analytics, embedded industrial systems, and real-time Internet of Things systems are provided.

A step-by-step methodology for selecting real-time system architectures is proposed, which includes formalization of time and non-functional requirements, definition of priority criteria, preliminary selection of architectural alternatives, their comparative evaluation, and verification of the selected solution through modeling or prototyping. The application of the proposed methodology allows reducing the risks of violating time requirements, increasing the validity of design decisions, and ensuring the compliance of the architecture with the specifics of modern real-time systems.

The results obtained can be used in the design and modernization of distributed, intelligent, and embedded real-time systems, as well as serve as a basis for further research in the field of RTS architectural design.

Keywords: *real-time systems, software architecture, architectural design, criteria, architectural approaches, determinism, time characteristics.*

Дата першого надходження статті до видання: 02.01.2026

Дата прийняття статті до друку після рецензування: 28.01.2026

Дата публікації (оприлюднення) статті: 08.04.2026