ОПРАЦЮВАННЯ ТА ПЕРЕТВОРЕННЯ ВИМІРЮВАЛЬНИХ СИГНАЛІВ

УДК 536.5; 536.5.081

РОЗУМНІ ВИМІРЮВАЛЬНІ ЗАСОБИ ДЛЯ КІБЕРФІЗИЧНИХ СИСТЕМ
SMART MEASURING INSTRUMENTS FOR CYBER-PHYSICAL SYSTEMS

Національний університет "Львівська політехніка",
кафедра інформаційно-вимірювальних технологій,
вул. С. Бандери, 12, 79013, Львів, Україна,
(e-mail: slav.yat@gmail.com +38 068 50 473 68)

Праця спрямована на розвиток кіберфізичних систем, які стають ключовим фактором повсякденного життя, а розумні вимірювальні прилади вважають невід'ємним компонентом цієї системи. Розглядається верифікація метрологічних підсистем за параметрами, що визначають керованість обладнання та процесів, розробленням, впровадженням та реалізацією конкретних метрологічних методів та інструментів, які успішно описуються термінами "апаратна підтримка, основне і проміжне метрологічне програмне забезпечення".
Ключові слова: розумні вимірювальні інструменти, мережі із розумних сенсорів, сенсорні розподілені мережі, основне і проміжне метрологічне програмне забезпечення, верифікація.

Работа направлена на развитие киберфизических систем, которые становятся ключевым фактором повседневной жизни, а умные измерительные приборы считаются неотъемлемым компонентом этой системы. Рассматривается верификация метрологических подсистем по параметрам, определяющим управляемость оборудования и процессов, путем разработки, внедрения и реализации конкретных метрологических методов и инструментов, которые успешно описываются терминами "аппаратная поддержка, основное и промежуточное метрологическое программное обеспечение".
Ключевые слова: умные измерительные инструменты, сети из умных сенсоров, сенсорные распределенные сети, основное и промежуточное метрологическое программное обеспечение, верификация.

Smart measuring instruments are the prerequisite for CPS design as they constitute the essential units of information-measuring subsystems. There is a set of smart measurement instruments which is divided into the following subsets: smart sensors, smart transducers, their grids etc. that can be joined together in modern wireless sensor networks. The emerging field of cheap and easily deployed sensors offers an unprecedented opportunity for a wide spectrum of various applications. When combined, they offer numerous advantages over traditional networks. These include a large-scale flexible architecture, high-resolution data, and application-adaptive mechanisms as well as a row of metrological specific features and performance (self-check, self-validation, self-verification, self-calibration, self-adjustment). Milestones in everyday work aiming to ensure reliable wireless sensors networks operation lie in the direction of functional and probabilistic verifications. We provide the software and middleware development aiming to reach predetermined behavior. The easiest way to achieve this may be demonstrated on the example of widespread wireless fire detector networks. They are characterized by a number of special algorithms directed on as fast as possible and accurate triggering and actuating the automation of higher level. So, it becomes necessary to research and implement the original operation algorithms for fire sensors and also check algorithms for periodic real-time software examination. Considering their structural complexity (presence of smoke and heat sensitive elements, various principles of elaboration of the received signals, their drift of characteristics, and pollution of translucent elements, etc.) the develop-

ment of such algorithms is a daunting task. Herein, human life may be the price for a bug. Equally important seems to be probabilistic verification that is to boost the probability of reaching wireless sensors network declared goals (estimation of their chances being achieved).

Each network consists structurally of a large number (up to 10^3) of nodes which are individual sensors able to radio communicate with one or several neighboring units. The most common wireless sensors network is the fire alarm sensors network each branch of which has up to 2^6 sensors which was caused by limiting the length of microcontroller register. Topology of every network may differ: star, cluster tree, mesh, up to advanced multi-hop mesh network. Propagation technique between hops of network can be routing or flooding. Nowadays, problem arises to adapt traditional network topologies to contemporary communicating conditions.

If a centralized architecture is used in a sensor network and the central node fails, then the entire network will collapse, however the reliability of sensor network can be increased by using distributed control architecture. Distributed control is used in such networks for the following reasons: sensor nodes are prone to failure; for better collection of data; to provide nodes with backup in case of the central node failure; resources have to be self -organized.

Aiming at the substantial development of Cyber-Physical systems, which are becoming a key element of everyday life, the smart measuring instruments are considered below as an indispensable part of entire systems. Verification of the metrological subsystems for parameters determining the controlled equipment and processes through the development, implementation and realization of specific metrology and standardization methods, instruments, that is successfully described by the terms "metrological hardware, software, and middleware".

Smart sensors are supplied with digital information transmissive means by equipping them with built-in digital controllers to match the universal network interface or by combining technology of analogue and digital transmission in a single measuring channel. According to the structure all smart sensors are divided into 4 groups: sensors of centralized and decentralized types, as well as sensors with digital and analogue buses. According to correction methods the analogue interfaces with smart sensors are divided into the groups: with manual error correction, with auto correction of errors in analogue-digital form, and with digital correction of errors.

Specific measurement consists in evaluating MIs performance reliability, trueness, and other metrological properties, due to the quality of a certain kind of metrological software, or the software linked to metrological features of MIs.

MI software metrological verification raises the problem of appropriate methods choice of software and middleware assessing, testing, and certifying. The metrological validation must result in confirmation or discarding of the studied ware following the requirements indicated in normative documents. Procedures and methods of checking software, and determining its disadvantages are considered below. Software study includes first of all the fulfilling the procedures of inambiguity ensuring the operating functions for generated data. Selection of the procedures is determined by regulation requirements, as well as by the software developer or the user's desires to confirm its compliance with the target specification.

Key words: smart measuring instruments, smart sensors grids, distributed sensors networks, metrological software and middleware, verification.

Вступ. Розумні вимірювальні засоби (далі РВЗ) – передумова для створення кіберфізичних систем (КФС), оскільки вони являють собою основні компоненти інформаційно-вимірювальних підсистем. Існує безліч розумних РВЗ, які поділяються на підкласи: розумні сенсори, розумні перетворювачі, їх мережі тощо, які можуть об'єднуватись у сучасних бездротових сенсорних мережах (БСМ). Розлога сфера дешевих і легковстановлюваних сенсорів надає безпрецедентні можливості для широкого спектра різноманітних застосувань. У поєднанні такі сенсори забезпечують низку переваг порівняно з традиційними мережами. Ці переваги полягають у гнучкості великомасштабного компонування мереж, можливості отримання даних із високою роздільною здатністю, наявності адаптивних механізмів пристосування під варіативні технологічні вимоги, а також у наявності низки метрологічних рис та характеристик (самодіагностика, самоперевірка, самоверифікація, самокалібрування, самонастроювання чи самостійне введення поправок у показання приладів).

Аналіз останніх досліджень у галузі. Особливої уваги, на думку вчених Університету Карнегі–Мел-

лоун, потребує верифікація БСМ відповідно до відомої методології. Терміни верифікації повинні гарантувати виконання завдань, поставлених перед БСМ, які працюють у режимі реального часу, протягом запланованого інтервалу часу. У цій галузі зусилля спрямовуються на розроблення методів дотримання наперед заданого графіка роботи для багатоядерних платформ, де існують нові виклики щодо спільно використовуваних ресурсів, таких як об'єми пам'яті, різні та періодично змінювані алгоритми опрацювання, зберігання, передавання та нагромадження сигналів і, відповідно, всі ці питання мають бути детально розглянуті. Це вкрай необхідно для прийняття правильних рішень у системах реального часу, особливо в галузі охорони здоров'я та безпеки життєдіяльності.

Проблеми. Повсякденна робота з метою реалізації надійної роботи БСМ спрямована на забезпечення функціональної та ймовірнісної верифікацій. Це означає, що необхідно забезпечити розроблення основного та проміжного програмного забезпечення для досягнення наперед прогнозованої поведінки систем. Найпростіший шлях досягнення зазначеної мети можна продемонструвати на прикладі бездротових мереж пожежних сповіщувачів. В їхній роботі застосовується низка спеціальних алгоритмів, спрямованих на якнайшвидшу і точну сигналізацію та на задіювання автоматики вищого рівня. Для цього необхідно вивчити і реалізувати оригінальні алгоритми роботи пожежних сенсорів, а також перевіряти програмне забезпечення для роботи у режимі реального часу й змінювати його уставки залежно від пори дня та сезону. Зважаючи на їхню структурну складність (наявність димових та теплових чутливих елементів, різні принципи опрацювання прийнятих сигналів, дрейф характеристик, забруднення напівпрозорих елементів тощо), створення таких алгоритмів є доволі складним завданням. Ціною дрібної неточності може стати людське життя. Не менш важливо, як видається, здійснити імовірнісну верифікацію, тобто підвищити ймовірність досягнення БСМ заявленої цілі (у разі оцінювання цих шансів). Це можна виконати двома відомими способами, які ґрунтуються на: а) імовірнісній моделі, основаній на марковському підході; б) аналогічній моделі на основі методу Монте-Карло.

Мета роботи – дослідження можливостей розвитку і вдосконалення розумних засобів вимірювання та їх мереж, оснащених вбудованим та додатково імплементованим програмним забезпеченням, причому можливостей, спрямованих на їх подальше поширення, зокрема, у складі кіберфізичних систем.

Виклад основного матеріалу

Організація та експлуатація КФС передбачають приділення уваги автономній взаємодії різних БСМ, що складаються з варіативних у часі автономних груп різнорідних сенсорів. БСМ можуть бути відкритими, закритими або виокремленими; останні стосуються БСМ, що виконують цілеспрямовані завдання.

Розвиток нового підходу до створення КФС полягає у формуванні у БСМ здатності швидкої адаптації до змін умов довкілля і навіть до несправностей ("Самоадаптивна система – це система, здатна змінювати свою поведінку і структуру, з метою адаптації, без втручання людини, до змін у собі та у власному робочому середовищі" [1]). Ми повинні розвивати не тільки загальні стратегії адаптації, серед яких прогнозування еволюції навколишнього середовища, а й конкретні запобіжні метрологічно обґрунтовані адаптивні механізми, зумовлені зазначеними вище змінами. Якщо це стосується метрологічних інструментів, то, зокрема, це можуть бути механізми, пов'язані зі зміною метрологічних характеристик, наслідок чого виникає дисфункція КФС. Завдяки моделюванню така адаптація може виконуватись заздалегідь, активно випереджаючи поточний дрейф або інші види змін характеристик. У результаті це зекономить час, необхідний для упровадження адаптаційної стратегії, яка повинна виконуватись.

Розумний сенсор, відповідно до загальноприйнятих визначень, містить в одному корпусі чутливий елемент, мікросхеми аналогового інтерфейсу, АЦП та інтерфейс шини [2]. Крім цього, у розумному сенсорі передбачено мікропроцесор, який кондиціює сигнали (термін "кондиціювання" відповідає встановленню необхідних амплітуди та форми сигналів перед відправленням їх до подальших вузлів), відфільтровує небажані шуми та виправляє помилки перед відправленням даних. Створення класу розумних сенсорів нового покоління, однак, передбачає необхідність у додаткових функціональних можливостях, зокрема, у самотестуванні, самоідентифікації, самовалідизації або самоадаптації. Зацікавлення проектувальників викликають такі можливості розумних сенсорів, як самокалібрування та самодіагностика, можливість опрацьовувати сигнал, а також мультисенсорні можливості.

Для прикладу, розумний сенсор температури виконується конструктивно, як первинний аналоговий або цифровий термочутлиий перетворювач, оснащений блоком опрацювання сигналу та інтерфейсом [3], який може виконати низку розумних метрологічних операцій завдяки спеціальному метрологічному програмному забезпеченню. Практично це може бути інтелектуальний сенсор температури з низкою спеціалізованих алгоритмів, передбачених на етапі проектування й виготовлення або установлених пізніше, тобто сенсор з такими вбудованими алгоритмами, які необхідні для забезпечення виконання спеціалізованих метрологічних функцій. Зокрема, такі функції передбачають здатність реалізувати: автоматичне перемикання діапазонів вимірювання, залежно від значення вхідного сигналу; автоматичну самовалідизацію; самоперевірку; самодіагностику тощо; введення поправок у покази, коли зафіксовано дію фактора впливу; лінеаризацію номінальних характеристик; компенсацію температури холодних кінців термоелектричного перетворювача тощо.

Згадані автономні розумні сенсори, попередньо просторово розподілені, часто об'єднують у мережі, призначені для моніторингу заданих умов або умов довкілля (***T, p*** тощо) і спільно та узгоджено надсилають отримані й опрацьовані дані від мережі до кінцевого користувача. Набір таких сенсорів називають БСМ, якщо вони сполучені між собою і розподілені в просторі. Окрім того, найсучасніші БСМ можуть працювати двонаправлено, що означає наявність контролю функцій розумних сенсорів та віддаленого оновлення їх програмного забезпечення під час експлуатації.

Розподілені сенсорні мережі та їх розвиток. Кожна БСМ структурно складається зі значної кількості (до 10^3) вузлів, які являють собою індивідуальні сенсори, здатні за наявності радіозв'язку спілкуватися один з одним або з декількома сусідніми одиницями. Прикладом такої БСМ є мережа сенсорів пожежної сигналізації, де кожна гілка налічує до 26 сенсорів, що зумовлено обмеженням довжини регістра мікроконтролера [4]. Топологія кожної БСМ може відрізнятися: зірка, кластерне дерево, сітка, аж до передових багатострибкових БСМ. Поширення сигналів перестрибуванням між вузлами мережі може здійснюватись за зазначеним маршрутом або за технологією фладингу (останнім вважається простий роутинговий алгоритм, за яким кожний вхідний пакет висилається на всі можливі адреси, за винятком того пункту, звідки він надійшов) [5].

Ранні сенсорні мережі використовували прості скручені екрановані подвійні дроти для підлучення кожного сенсора. Пізніше промисловість почала застосовувати шини з багатоточковими підлученнями (наприклад, Ethernet). Тепер вже застосовують веб-мережі (наприклад, World Wide Web) навіть в умовах виробництва. Топології мереж (точка-точка, багатоточкова і веб-мережа) добре відомі. У надійнішої першої кожен вузол сенсора потребує окремого підлучення скрученою екранованою двійкою дротів. У багатоточковій мережі кожен сенсорний вузол розташовує власну інформацію у загальному середовищі, тому треба звернути увагу на протоколи програмно-апаратного забезпечення.

Як приклад, декілька видів ранніх мереж використовували частотну модуляцію сигналів для передавання по дротах сигналів різних сенсорів по FM каналах. Фактично ранні БСМ являли собою прості радіочастотні реалізації зазначених топологій. Тут застосовували радіочастотні модеми для перетворення RS-232 сигналу на радіосигнал і навпаки. Внаслідок простоти кодування та істотного впливу завад не забезпечувалась достатня надійність таких мереж. Тільки за умови подальшого розділення частотних діапазонів роботи різних мереж розвинулися бездротові локальні мережі.

Віддалені системи збирання й нагромадження даних БСМ схожої топології часто реалізуються з винесеними концентраторами даних і радіопередавачем, що гарантує роботу "хостів", де сигнали демультиплексуються у вихідні сигнали сенсорів.

Після того, як промисловість почала перехід на багатоточкові шини, виникли проблеми, зумовлені оцифруванням інформації [6]. Особливо це притаманно системам точко-точкової топології, де один годинник хоста може використовуватись для відліку часу, тоді як аналогові сигнали надходять від декількох сенсорів з їхніми показами часового відліку. У випадку розподілених смарт-сенсорів, потрібних для реалізації багатоточкової мережі, синхронізація їх годинників стає критично важливою для функціонування мережі. Дещо інші проблеми притаманні веб-топології, в якій всі вузли потенційно під'єднані до всіх інших вузлів. Деякі з них можна усунути, застосовуючи ретранслятори і маршрутизатори, що реалізують віртуальні з'єднання.

Істотну зацікавленість викликає ***децентралізований тип БСМ*** (ad hoc мережа – мережа на вимогу, що є перескоковою, оскільки вона не стосується наперед усталеної структури, зокрема відсутні маршрутизатори у дротових мережах чи точки доступу в організованій структурі БСМ). Натомість кожен вузол мережі бере участь у маршрутизації, перекидаючи дані на сусідні вузли, оскільки визначення того, на які вузли здійснюється перекидання, виконується динамічно на основі мапи з'єднань мереж. На додаток до класичної маршрутизації ad hoc мережі можуть використовувати фладинг для перекидання даних. Фладинг поширений у мостах та в таких системах, як Usenet, і у файлообмінних мережах (*peer-to-peer file sharing*), а також як частина деяких роутингових протоколів, зокрема OSPF, DVMRP, що вже застосовуються у бездротових ad hoc мережах). Децентралізований тип БСМ – це багатоланкові мережі, які складаються з бездротових автономних хостів, де кожен хост може слугувати маршрутизатором для передавання трафіку від інших вузлів [7]. Бездротові однорангові ad hoc мережі охоплюють широкий спектр мережевих видозмін, що містять сенсор, мобільний механізм перескоків, персональну специфіку чи інші мережі.

Науково-дослідницька активність у конкретній галузі БСМ передбачає навчання сенсора (переведення його з розряду інтелектуальних до розумних), вивчення безпекових аспектів за рахунок інтелектуальних можливостей вузла, зони покриття сенсора за його випадкового або детермінованого розміщення, розташування об'єкта, визначення впливу розміщення сенсора, енергетичної ефективності передавання інформації та розкладу діяльності, маршрутизації, топологічних зв'язків, поширення даних та їх нагромадження, впливу дерева реконфігурацій і конструювання конкретної топології. Для нормальної роботи мережева топологія передбачає розгляд не тільки сенсорних вузлів, а й базових станцій та кросшарів. Структура останніх стає дуже важливою для БСМ, оскільки їх можна використати для оптимальної модуляції з метою підвищення параметрів передавання, таких як швидкість передавання даних, ефективність використання енергії тощо.

Ідеальний бездротовий сенсор повинен під'єднуватись до мережі та мати змогу бути масштабованим, розумним і програмованим, здатним швидко збирати дані, бути достовірним і точним протягом тривалого часу, а також потребувати мінімального технічного обслуговування [7]. Вибір оптимальних сенсорів та бездротової лінії зв'язку потребує знання механізмів застосування програми та способів розв'язання проблемних завдань. Термін служби батарей, швидкість відновлення характеристик сенсора стають основними для визначення конструкції мережі. Прикладами сенсорів з малою швидкістю передавання даних вважають температурні та вологісні сенсори. Сенсори з високою швидкістю передавання даних – це сенсори деформації, прискорення і вібрації. Сучасні приклади надвисокошвидкісних сенсорів – це сенсори теплових оптичних томографів, сенсори дистанційного керування та інші.

Поступ у мікроелектроніці призвів до появи у сенсорів нової якості, що полягає у можливості створення інтегрованих одиниць. Такі сенсори набувають здатності взаємодіяти між собою бездротово за допомогою принципово інших протоколів маршрутизації даних. БСМ повинна складатися з базової станції ("шлюз"), що може взаємодіяти з декількома бездротовими сенсорами за допомогою радіохвиль. Дані накопичуються в сенсорному вузлі, ущільнюються архівуванням і передаються на основну або проміжну базову станцію.

Протоколи передавання даних у мережах і архітектура мереж. Відомі звичайні мережеві стандарти IEEE802.11x використовуються для локальних мереж для передавання даних з високою пропускною спроможністю у широкій смузі частот; IEEE802.15.1 і 2 відомі як Bluetooth. Їм притаманна менша потужність і відповідно вони призначені для персональних мереж. IEEE802.15.4 спеціально розроблений для вимог БСМ, тут у всьому світі виділено неліцензовану смугу частот 2 .. 4 ГГц. ZigBee підпадає під специфікацію IEEE802.15.4, розширений до мережевої специфікації. Один з найсучасніших стандартів – IEEE1451.5, спрямований на забезпечення роботи робочих груп смарт-сенсорів у бездротовому варіанті.

Якщо у сенсорній мережі застосовується централізована архітектура, а її центральний вузол виходить з ладу, то вся мережа опиняється у стані колапсу. Проте надійність мережі сенсорів можна збільшити за допомогою розподіленої архітектури управління. Остання використовується у БСМ з таких причин: сенсорні вузли схильні до відмов; для покращення якості збирання даних; для забезпечення

вузлів резервуванням у разі виходу з ладу центрального вузла; для самоорганізації ресурсів.

Розумний мережевий сенсор являє собою невеликий легкий вузол, який водночас може слугувати станцією виявлення самої мережі. Такі сенсори дають змогу віддалено моніторити оснащення, таке як трансформатори та лінії електропередач, а також за потреби організувати управління енергетичними ресурсами за допомогою зазначеної смарт-мережі [8]. Сенсори смарт-мережі також можуть використовуватись для моніторингу погодних умов та температури лінії електропередавання, що надалі можна використати для коригування пропускної здатності енергетичної лінії. Цей процес називається динамічним нормуванням енергетичних ліній і дає змогу енергетичним компаніям збільшити потік потужності, який подається чинними лініями електропередавання. Розумні мережеві сенсори також можуть бути використані у будинках й офісах для підвищення їхньої енергоефективності [9]. За інформацією Nano Markets, такі компанії, як GE, LG і Whirlpool, вже оголосили про свої наміри щодо створення побутової техніки зі смарт-підтримкою. Розумні мережеві сенсори об'єднують ці пристрої у смарт-лічильниках і забезпечують облік споживаної потужності у режимі реального часу. Енергетичні компанії зможуть використовувати цю здатність для коригування цін у режимі реального часу, а споживачі – для зменшення енергоспоживання у години пік [10].

Проміжне програмне забезпечення і бездротові сенсорні мережі. Проміжний шар програмного забезпечення (Middleware) являє собою новий підхід до повного задоволення вимог проектування та реалізації завдань БСМ технології. Термін Middleware зазвичай використовується для програмного забезпечення, яке забезпечує комунікування та керування даними у розподілених сенсорних або інших системах. У вужчому сенсі Middleware можна назвати "тире між клієнтом і сервером". ObjectWeb визначає Middleware як: "Шар програмного забезпечення, розміщений між операційною системою і додатками (сенсорними програмами) на кожній стороні розподіленої обчислювальної системи у мережі". Послуги, які можна розглядати, як Middleware, передбачають інтеграцію згаданих додатків, інтеграцію даних, комунікативно-орієнтоване Middleware (MOM), запит об'єкту через брокерів (ORBs – Middleware, що дає змогу основній програмі виконати запит з одного комп'ютера на інший через комп'ютерну мережу, – це забезпечивши прозорість розташування через віддалені процедурні виклики.

Програмно-апаратна підтримка Middleware – це веб-сервери, сервери розділених мереж, системи управління контентом та аналогічні інструменти, які підтримують життєдіяльність, забезпечення та розвиток згаданих систем. Middleware успішно інтегрується в інформаційні технології на основі Extensible Markup Language (XML); простого протоколу доступу до об'єктів (SOAP); веб-служб; SOA (service-oriented architecture) – модульного підходу до розроблення програмного забезпечення; Web 2.0 – інформаційних технологій, які дають змогу користувачам створювати та поширювати власний контент у Всесвітній павутині; інфраструктуру та полегшений протокол служби каталогів (LDAP).

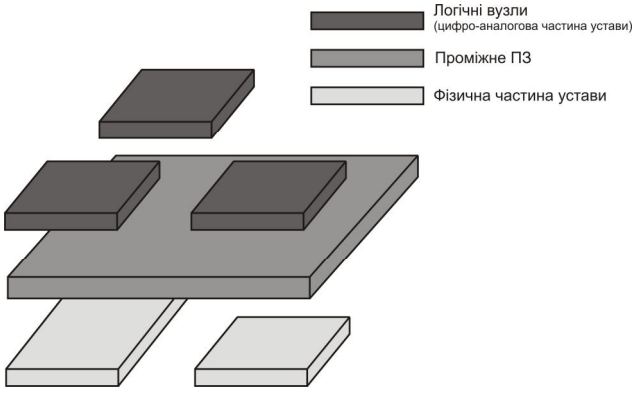
У симуляційній технології проміжне програмне забезпечення, як правило, застосовується у контексті архітектури високого рівня для низки розподілених задач моделювання. Це – прошарок програмного забезпечення, розміщеного між кодом програми та часовиконавчою (run-time) інфраструктурою. Остання – Middleware, необхідна для імплементації Архітектури Високого Рівня (HLA). Middleware здебільшого складається з бібліотеки функцій, а також дозволяє низці додатків (моделювання або федерати в термінології HLA) скопіювати ці функції із загальної бібліотеки, а не заново створювати їх для кожної програми. Бездротові мережеві розробники можуть використовувати проміжне програмне забезпечення для вирішення завдань, пов'язаних з БСМ. Імплементація Middleware дає змогу розробникам інтегрувати операційні системи та апаратні засоби з широким спектром різних додатків, доступних на цей момент. Наприклад, радіочастотні програмні набори інструментальних засобів використовують Middleware для фільтрування шумів та надлишкових даних.

Розподілені обчислювальні системи Middleware можна поділити на дві категорії – ті, які забезпечують обслуговування людини (наприклад, обслуговування веб-запиту), й ті, які виконують подібне, працюючи у машинному часі, себто самостійно, у межах попередньо закладеної програми. Останнє вважається таким, що стандартизоване службою Availability Forum, і зазвичай використовується у комплексних вбудованих системах зв'язку, оборони та аерокосмічної промисловості.

Інші приклади Middleware. Термін Middleware іноді використовують у значенні драйвера програм-

ного забезпечення, абстрактного рівня, який приховує подробиці щодо апаратних пристроїв або щодо іншого програмного забезпечення сенсорів.

У технології моделювання Middleware зазвичай використовується у контексті архітектури високого рівня (HLA), яка застосовується до моделювання багатьох розподілених систем. Це шар програмного забезпечення, розташований між кодом програми та інфраструктурою, що займає наступні цикли виконання програми у часі (див. рисунок). Middleware, як правило, містить бібліотеку функцій і дає змогу низці програм-симуляцій або програм-федератів у HLA-термінології (дружніх програм) скопіювати певні згадані функції із загальної бібліотеки, а не створювати їх знову для кожної програми.



Middleware та його місце в архітектурі програмного забезпечення розподілених систем

Middleware and its place in the architecture of the software of the dispersed systems

Розробники бездротових мереж можуть використовувати Middleware, щоб успішно реалізувати завдання, пов'язані з функціонуванням бездротової сенсорної мережі. Реалізація додатків Middleware дає змогу розробникам мереж об'єднувати операційні системи й устаткування у широке розмаїття різних застосувань, доступних на цей момент.

Розглянемо проміжне програмне забезпечення БСМ як програмне забезпечення інфраструктури, що сполучає мережеве обладнання, операційні системи, мережеві стеки та додатки. Комплексне рішення проміжного шару повинно міститись у заданому середовищі виконання, що підтримує та координує низку додатків, а також стандартні системні сервіси, такі як агрегація даних (агрегування даних – видобування інформації з баз даних з метою підготовлення комбінованих наборів даних для їх подальшого опрацювання), контроль і управління політикою адаптування до цільових програм, а також механізми для досягнення адаптивних та ефективних системних ресурсів, використовуваних для подовження ресурсу роботи сенсорної мережі. Проміжний шар повинен підтримувати моделі низького рівня програмування, щоб реалізувати основне завдання – подолання розриву між потенціалом апаратних технологій і необхідними працемісткими заходами, такими як реконфігурація, виконання та зв'язок. Сенсорний вузол повинен виконувати свої три основні операції – вимірювання, опрацювання даних та їх передавання через інтерфейс, не вичерпуючи енергетичних ресурсів. В енергоекономних middleware, наприклад, більшість електронних компонентів апаратних засобів (зокрема радіо) мають бути вимкненими більшу частину часу.

Релевантні проекти проміжного програмного забезпечення для БСМ вивчено в [11]. У тих випадках, коли фізичний контакт для заміни або технічного обслуговування неможливий, бездротові засоби стають єдиним засобом для реалізації віддаленого доступу. Отже, проміжне програмне забезпечення повинно допомогти залучити механізми для ефективного використання процесора та пам'яті за низькоенергетичного споживання комунікацій.

Більшість додатків БСМ реалізуються у реальному часі, тому проміжне програмне забезпечення повинно надавати послуги у режимі реального часу, щоб адаптуватися до змін і забезпечити несуперечливість та послідовність даних. Застосування принципів проектування диктує ще одну важливу властивість проміжного шару БСМ. Він повинен містити механізми для впровадження відомостей про інфраструктуру БСМ. Це дає змогу розробникам виставити карту вимог до параметрів мережі, які нададуть їм змогу точніше відстежувати налаштування мережі. Інтенсивне впровадження БСМ для жорстких умов довкілля погіршує їх стійкість до вторгнень та хакерських атак, аж до відмови в обслуговуванні. Окрім того, бездротове середовище створює можливості для зняття інформації сторонніми особами та для конкурентного впровадження сторонніх пакетів, що становлять загрозу функціонуванню мережі. Перелічені фактори роблять проблему безпеки надзвичайно важливою. Окрім цього, сенсорні вузли мають обмежену потужність та ресурси опрацювання і стандартні механізми безпеки, які потребують істотних

ресурсів, тут непридатні. Тому необхідно розробити комплексні та безпечні рішення, які дадуть змогу реалізувати надійніший захист за збереження бажаної продуктивності мережі. Саме тому під час розроблення і впровадження проміжного програмного забезпечення, зокрема на початкових етапах роботи, треба зосередити увагу на питаннях розвитку і безпеки, а саме на конфіденційності, аутентичності, цілісності та доступності [12, 13].

Можливість формування великомасштабної гнучкої архітектури, з високою роздільною здатністю даних вимірювань крихітних мережевих сенсорів (з низькою собівартістю й енергоспоживанням, можливістю легкого встановлення на об'єктах) дає змогу реалізувати широкий спектр різних застосувань. Такій БСМ притаманні численні переваги над традиційними сенсорними мережам за наявності адаптивних механізмів застосувань. Створення проміжного програмного рівня можна вважати новим підходом у проектуванні та впровадженні робіт з бездротових мережевих сенсорних технологій. Тому middleware бездротової сенсорної мережі, як проміжна програмна інфраструктура, поєднує мережеве обладнання, операційні системи, мережеві стеки і додатки. Повне вирішення middleware повинне містити середовище виконання, яке підтримує і координує декілька різних застосувань, стандартизовані системні сервіси, такі як нагромадження даних, контроль та управління адаптацією щодо мети застосування, а також механізми для використання адаптивних та ефективних системних ресурсів для продовження ресурсу роботи сенсорної мережі.

Основні характеристики БСМ

Масштабованість та переформатування мережі. Масштабованість визначається так: якщо розміри мережі зростають, мережа має бути достатньо гнучкою, щоб уможливити це зростання у будь-якому місці й у будь-який час без впливу на характеристики мережі. Ефективні middleware послуги повинні підтримувати прийнятний рівень продуктивності (характеристик) за збільшення розмірів мережі. Мережевим топологіям притаманні зміни внаслідок таких факторів, як несправності, відмови пристрою, рухомі перешкоди, завади. Middleware повинні підтримувати надійну роботу сенсорних мереж, незважаючи на зазначені фактори, адаптуючись до змінних зовнішніх умов роботи мережі. Middleware також повинно підтримувати механізми стійкості до відмов, самоконфігурацію та самопідтримку характеристик сенсорних вузлів.

Масштабованість визначається так: якщо завдання мережі збільшуються за обсягом, то БСМ повинна бути достатньо гнучкою, щоб забезпечити це зростання будь-де і в будь-який час без погіршення продуктивності мережі. Ефективні послуги проміжного шару повинні бути здатними підтримувати прийнятний рівень продуктивності, оскільки мережа змінює свої розміри та топологію. Остання схильна до частих варіацій внаслідок таких факторів, як несправність, вихід з ладу пристрою, мобільність тощо.

Гетерогенність. Проміжне ПЗ повинно забезпечувати моделі низькорівневого програмування, щоб подолати розрив між вихідним потенціалом програмного забезпечення та такими заходами, як переконфігурування, виконання та комунікування через інтерфейс.

Організація динаміки мереж. На відміну від традиційних мереж, сенсорні мережі мають справу з динамічними ресурсами, такими як енергопостачання та споживання, змінні ширина смуги пропускання та ресурси опрацювання даних. Сенсорні мережі також повинні підтримувати довготривалі додатки, отже, протоколи маршрутизації повинні бути ефективно розроблені, здатні підтримувати роботу мереж якомога довше. Тому знання мережі має важливе значення для того, щоб працювати належно, а Middleware повинні забезпечити доставлення спеціального мережевого ресурсу. Сенсорні вузли повинні знати своє місце розташування у мережі та у всій топології мережі. У деяких випадках самостійно встановити власне місцезнаходження за допомогою GPS неможливо, недоцільно або високовартісно. Такі важливі параметри системи, як розмір мережі та щільність встановлення сенсорів на одиницю площі, можуть ускладнити пошук компромісів між затримкою, надійністю та енергоспоживанням.

Інтеграція у реальний фізичний світ. Здебільшого сенсорні мережі повинні працювати в режимі реального часу, коли час і простір є надзвичайно важливими. Отже, Middleware повинні забезпечити сервіси режиму реального часу для адаптації послуг до змін і досягнення несуперечності отриманих даних.

Знання засад застосувань. Принципи проектування залежно від галузі застосування мереж диктують ще одну важливу й унікальну здатність Middleware для бездротової мережі. Middleware повинні передбачати і

давати змогу залучати механізми для імплементації інфраструктури бездротової сенсорної мережі. Це дає змогу девелоперам звести вимоги комунікацій до мережевих параметрів, які дозволять точно настроїти мережу з метою моніторингу певних параметрів. Багато Middleware прив'язано до конкретних застосувань, однак, Middleware призначена підтримувати їх широкий спектр. Тому розробники повинні вивчити компроміси між ступенем специфічності Middleware для різних застосувань і ступенем їх спільності.

Об'єднання даних. Більшість застосувань мережевих сенсорів стосується вузлів, що містять надлишкові дані та розміщені у певній локальній зоні. Ці риси уможливлюють об'єднання в межах мережі даних з різних джерел, усунення дублювання і мінімізацію кількості непотрібних повідомлень. Це об'єднання істотно економить енергію і обчислювальні ресурси за умови, що вартість комунікації набагато вища від витрат на обчислення. Ця парадигма зміщує акцент розроблення та проектування мереж від традиційних адресно-орієнтованих до мереж із підходом, більше орієнтованим на дані.

Якість сервісу. Якість обслуговування – це термін, яким зловживають, оскільки у нього доволі багато значень, коли споглядати з перспективи різних досліджень і різних технічних громад. У бездротових сенсорних мережах можемо розглядати якість обслуговування з двох позицій: з боку їх конкретного прикладного застосування та зі сторони мережі. Перша позиція стосується параметрів якості обслуговування, специфічних для прикладного застосування, таких як метрологія сенсорного вузла, його встановлення; охоплення активними сенсорними вузлами заданого регіону та їх кількість. Останнє стосується того, як підтримувальний інтерфейс може задовольнити вимоги поставленого завдання застосування мережі у процесі ефективного використання мережевих ресурсів, таких як ширина смуги пропускання та споживання електроенергії. Традиційні механізми забезпечення якості обслуговування, використовувані у провідних мережах, не збігаються з подібними для бездротових сенсорних мереж через ресурсні обмеження та обмеження у динамічній топології (переконфігуруванні). Тому Middleware повинні надати нові механізми для підтримки якості обслуговування протягом тривалого періоду, і навіть підрегульовувати себе, коли необхідні якість обслуговування та стан мережі підлягають змінам. Middleware повинні розроблятись на основі компромісів між метрологічними показниками, такими як ємність мережі або її пропускна здатність, затримка доставки даних та споживання енергії.

Безпека. Сьогодні бездротові сенсорні мережі широко використовуються в галузях, пов'язаних з вимірювальною інформацією, наприклад, в охороні здоров'я та порятунку. Вільне й об'ємне розгортання бездротових сенсорних мереж у суворих умовах оточення підвищує їх вразливість до вторгнень і атак, зокрема до відмов в обслуговуванні. Крім того, бездротове середовище полегшує прослуховування і можливості цілеспрямованого впровадження чужорідних пакетів з метою спотворити функціонування мережі. Всі ці фактори формують безпекові пріоритети. Крім того, сенсорні вузли характеризуються обмеженими швидкістю та об'ємом пам'яті (RAM, ROM), тому типові безпекові механізми з істотною масою і споживанням ресурсів непридатні.

Ці проблеми збільшують потребу в розвитку комплексних та безпечних рішень, які забезпечують надійніший захист зі збереженням бажаних характеристик мережі. Зусилля Middleware повинні зосереджуватись на розробленні та інтеграції безпекових аспектів на початкових стадіях проектування програмного забезпечення, отже, на досягненні різних безпекових вимог, таких як конфіденційність, аутентичність, цілісність, новизна і доступність.

Конструкція Middleware як шару для сенсорних мереж, що повною мірою відповідає викликам, сьогодні відкрита для обговорення. Споконвічна проблема полягає у дотриманні вимог якості обслуговування за забезпечення високорівневого наближення, що стосується гетерогенності сенсорних вузлів. Іншим визначальним завданням вважається розвиток зручного у використанні, чіткого інтерфейсу програмування, придатного для вирішення проблем різних сенсорних мереж, таких як, наприклад, обмежені апаратні ресурси і вимоги якості обслуговування. Middleware, що використовує автономні обчислювальні потужності, може забезпечити більшу стійкість, надійність і самоуправління. Поки що незрозуміло, наскільки успішне управління мережею і належні програмувальні підходи випливають із описаних у роботі парадигм, або якщо все-таки нові підходи з'являться, то чи будуть вони спеціалізованими для цілей бездротових сенсорних мереж.

Нові завдання оптимізації математичних моделей вимірювальних перетворень. Підвищення вимог до точності вимірювання ставить нові завдання оптимізації математичних моделей вимірювальних перетворень, а також адекватного опрацювання експериментальних даних. Ця задача у сучасних засобах вимірювання – актуальна завдяки можливостям недорогої апаратної реалізації, що уможливлює реалізацію обчислювальної компоненти безпосередньо у вимірювальному тракті. Досягти потрібної точності нерідко можливо лише у разі застосування оптимальних математичних моделей, які забезпечують найкраще у певному розумінні наближення загальної функції перетворення вимірювальних пристроїв. Раціональний вибір математичної моделі функції перетворення вимірювального пристрою у багатьох випадках дає змогу покращити точність вимірювання або розширити діапазон вимірювання із заданою точністю.

У зв'язку з цим важливого значення набуває вибір придатного критерію під час опрацювання дослідних даних. Серед критеріїв, які сьогодні здебільшого застосовують для аналізу дослідних даних, найпоширенішим є метод найменших квадратів, який полягає у мінімізації суми квадратів похибок відхилення. Але середньоквадратичне наближення не забезпечує досягнення найменшого відхилення апроксиманти від наближуваної функції в усіх точках спостереження, що бажано під час опрацювання прецизійних дослідних даних. Тому для розв'язування задач градуювання слід застосовувати мінімаксний (чебишевський) критерій, який забезпечує мінімально можливі похибки відтворення експериментальної статичної характеристики градуювальною залежністю.

Особливості розумних засобів вимірювання та подальший розвиток наукових досліджень

Зазвичай аналоговий інтерфейс, як невід'ємна частина вимірювального каналу між первинним вимірювальним перетворювачем і АЦП, виконує такі функції: перетворення масштабу вимірювальних сигналів, їхню фільтрацію, компенсацію температури холодних злютів термоелектричних перетворювачів, гальванічне розділення передавального та приймального каналів, лінеаризацію характеристик й ініціалізацію пасивних перетворювачів, мультиплексування вимірювальних сигналів, передавання вимірювальної та сервісної інформації.

Розумним сенсорам притаманний цифровий відлік завдяки оснащенню їх вбудованими мікроконтролерами, що відповідає вимогам універсального мережевого інтерфейсу або вимогам об'єднувальної технології аналогового та цифрового передавання по єдиному вимірювальному каналу. За структурою всі розумні сенсори розділено на чотири групи: сенсори централізованого та децентралізованого типів, а також сенсори з цифровими й аналоговими шинами даних. Відповідно до методів корекції аналогові інтерфейси з розумними сенсорами розподілено на групи: з ручним уведенням поправок на похибку, з автоматичним уведенням поправок на похибку в аналого-цифровій формі, а також з цифровим уведенням поправок на похибку.

Подальший розвиток науково-дослідних праць має спрямовуватись на покращення метрологічних характеристик, зокрема, під час вимірювання багатофакторних величин за допомогою багатопараметричних смарт-інструментів. До прикладу, розглянемо коріолісові витратоміри [14] – гнучкі інструменти, здатні швидко переналаштовувати основні процедури. (функція перетворення витратоміра – залежність маси або об'єму рідини, що протікає крізь переріз його трубопроводу, від її швидкості; визначається порівнянням частотно-фазових характеристик двох однакових сенсорів на вході і виході витратоміра). Сенсори витратоміра визначають швидкість потоку й температуру та надають інформацію у формі вихідних сигналів до мікропроцесора, який виконує обчислення за певним алгоритмом і забезпечує доступ до дисплею, головного меню та пристроїв виведення інформації для взаємодії з інформаційними системами, наприклад, заправної станції. Головною перевагою такого засобу вимірювання є можливість роботи із газорідинною, твердофазно-рідинною та найскладнішою твердофазно-газорідинною сумішшю, й отримують дані витрат для кожної фазої (газової, рідинної та твердофазної).

Іншим прикладом необхідності розроблення смарт-інструментів є промислові системи томографії. Їх розвиток полягає в розробленні та реалізації методів і алгоритмів для опрацювання масивів результатів вимірювань за необхідності точного та швидкого отримання високоякісних зображень досліджуваного розподілу параметрів. Зокрема, це може бути просторовий розподіл електропровідності в об'єкті. Точності та швидкості покращення томографічних вимірювань досягли останнім часом [15], зменшивши адитивні складові похибки використанням диферен-

ційного методу; розробленням спеціальної методики розрахунку матриці чутливості з уникненням методичних помилок, притаманних різницевим методам, лише на основі розв'язання прямої задачі; вдосконаленням ітераційної процедури.

Верифікація метрологічного основного і додаткового програмного забезпечення. Метрологія програмного забезпечення зазвичай вважається визначальною для всіх галузей, у яких система програмного забезпечення або процес виконує певні задані функції: оцінювання надійності метрологічних інструментів, їх достовірності та забезпечення високої якості метрологічного програмного забезпечення або програмного забезпечення, пов'язаного з особливостями засобів вимірювання. Процедура проектування і розроблення метрологічних інструментів передбачає перебудову аналогових блоків і розширення функціональності цифрових блоків. До складу останніх входять мікроконтролери та програмовані логічні масиви даних. Ураховуючи збільшення "ваги" програмного забезпечення у засобах вимірювань, неузгодженість їх програмного забезпечення щодо вирішення вимірювальних завдань, випадкова чи зумисна зміна їхніх функцій можуть призвести до неправильного висвітлення результатів вимірювань. Тому доцільно здійснювати перевірку програмного забезпечення для оцінювання його впливу на метрологічні характеристики засобів вимірювання, а також можливість подальшого впровадження розглянутого програмного забезпечення як невід'ємної частини приладу, інструменту або метрологічного засобу [16].

Під програмним забезпеченням метрологічного засобу розуміємо набір програм і процедур, призначених для реєстрації, агрегації, опрацювання, збереження та надсилання результатів вимірювань. Таке програмне забезпечення, як функціональна частина метрологічного засобу, поставляється разом з обладнанням [17, 18]. Згідно з [18] до програмного забезпечення, що може негативно впливати на метрологічні характеристики засобів вимірювання, належать: а) програми і програмні модулі, які опрацьовують результати вимірювань; б) програми, призначені для розрахунку окремих параметрів програмного забезпечення, які впливають, зокрема негативно, на результати; в) програми і програмні модулі, які здійснюють представлення результатів вимірювань, їх зберігання і передавання, оновлення та ідентифікацію програмного забезпечення, формують безпекові аспекти розробленого програмного забезпечення [19] та захист даних; д) компоненти захищеного інтерфейсу повинні гарантувати надійний обмін даними між програмними модулями окремих одиниць кіберфізичних систем.

Останні два пункти стосуються загальної неметрологічної верифікації програмного забезпечення, оскільки тут не виконуються жодні інформаційно-обчислювальні дії, що впливають на результати вимірювань. Ці пункти характеризують коректність функціонування програмного забезпечення загалом. Тут повинні визначатись можливості залучення безпекових практик – аутентифікації, авторизації та аудиту – в кожній фазі верифікації, починаючи від розроблення і впровадження програмного забезпечення – до тестування і розгортання метрологічних засобів [20].

Безпека розроблення програмного забезпечення стала важливою частиною якості програмного забезпечення, зокрема, завдяки розвитку інтернет-технологій. Заходи безпеки можуть запропонувати базовий захист для основних ділянок смарт-систем, тоді як безпечне програмне забезпечення стає вирішальним для гарантування повної безпеки у середовищі конкретної КФС. Кожен розробник програмного забезпечення метрологічних засобів повинен звертати увагу на безпекові аспекти; тому користувачі повинні мати можливість довіряти пропонованому програмному забезпеченню, і основному, і допоміжному. Повний цикл розроблення вищезгаданого передбачає такі вимоги: інжиніринг, довіру і моделювання загроз, безпекові кодування та тестування, реагування у відповідь на загрозу розшифрування коду [21].

Метрологічна перевірка програмного забезпечення засобів вимірювання висвітлює проблему вибору відповідних методів основного і допоміжного програмного забезпечення з оцінкою, тестуванням і сертифікацією. Метрологічна валідизація повинна привести або до підтвердження якості, або до відмови від досліджуваного виробу (конкретного ПЗ) на підставі процедур зіставлення з вимогами, зазначеними у нормативних документах. Дослідження передбачають, передусім, перевірку програмного забезпечення і передусім виконання процедур, які забезпечують робочі функції для спеціально генерованої вибірки даних. Вибір процедур визначається вимогами, які поставили розробник програмного

забезпечення або користувач, щоб підтвердити його відповідність призначенню.

Для валідизації типу метрологічного інструменту процедура випробувань повинна передбачати ідентифікацію та оцінку впливу програмного забезпечення на метрологічні характеристики, а також запобігання несанкціонованій зміні конфігурації програмного забезпечення і усунення впливу завад, які можуть погіршити достовірність отриманих результатів вимірювань. Отже, розроблений під час попередніх випробувань запланований опис типу інструменту повинен містити додатково до метрологічних характеристик опис програмного забезпечення, його ідентичності, оцінки впливу завад, а також рівня захисту від ненавмисних чи навмисних змін. Сертифікація основного і проміжного програмного забезпечення метрологічного інструменту – це дослідження, метою якого є визначення характеристик, ідентифікаційних даних та підтвердження відповідності поставленим вимогам. Відповідно до методології сертифікації програмного забезпечення, для визначення однієї або декількох характеристик (аналіз документації і вихідного коду, функціональне обстеження у контрольованих умовах тощо) згідно з [22] повинні проводитися випробування.

Розрізняють ***два види сертифікації: загальну і метрологічну***. Об'єктом загальної оцінки є повне програмне забезпечення, яке вивчають, щоби пересвідчитись у коректності застосування алгоритму (програми) в межах конкретних завдань. Для оцінювання впливу програмного забезпечення на неточність даних вивчається лише те, чи перевірюване програмне забезпечення є невід'ємною частиною програмного забезпечення конкретного метрологічного інструменту. Сертифікація програмного забезпечення є переважно добровільною, за винятком програмного забезпечення, яке виконує особливо відповідальні функції, коли відсутність якості, помилки або збої можуть істотно порушити роботу з небезпечними для життя і здоров'я наслідками (в авіації, ядерній енергетиці, управлінні банками тощо) [23]. Важливо вибрати правильний метод перевірки для основного та проміжного програмного забезпечення метрологічного інструменту.

Метод порівняльних випробувань із застосуванням зразкового програмного забезпечення використовують для конкретного програмного забезпечення, що дає змогу ідентифікувати його особливості. Як зразкове програмне забезпечення можуть застосовуватись: спеціально створене або/і сертифіковане програмне забезпечення, функціональність якого ідентична тестованим функціям перевірюваного програмного забезпечення; спеціально розроблене програмне забезпечення з функціями, ідентичними функціям програмного забезпечення, яке має перевірятись; програмне забезпечення для вирішення обчислювальних завдань (наприклад, електронні таблиці, програмне забезпечення для математичних та статистичних розрахунків тощо).

До розроблення зразкового програмного забезпечення можна вдаватися у тих випадках, коли програмне забезпечення не занадто складне, а алгоритм його впровадження порівняно простий. Зразкове програмне забезпечення не повинно відтворювати всі функціональні можливості тестованого програмного забезпечення, а може містити лише функції та параметри, які впливають на метрологічні характеристики контрольованих засобів. У певних випадках окремі функції користувача інтерфейсу, які не беруть участі в опрацюванні результатів вимірювань (наприклад, функції індикації, зберігання даних тощо) не враховують. Недоліком цього методу є те, що часто складність імплементації такого програмного забезпечення призводить до висновку про недоцільність його застосування внаслідок значних витрат на розроблення зразкового програмного забезпечення.

За відсутності зразкового програмного забезпечення пріоритет віддається методу порівняльних випробувань із застосуванням моделей вихідних даних та порівняльного тестування з генеруванням "еталонних" даних. Останній рекомендується для сертифікації алгоритмів опрацювання даних результатів вимірювань. Метод дає змогу оцінити можливості алгоритму, порівнявши результати опрацювання вихідних даних для моделі, отриманої за допомогою згаданого алгоритму, стосовно конкретних параметрів цієї моделі. Метод моделей вихідних даних – це свого роду метод генерації "стандартних" даних, а не тільки даних, отриманих за допомогою спеціально розроблених програм, тобто тільки тих даних, що не генеруються спеціально розробленою програмою, а програмними засобами встановлюються на вході програмного забезпечення, яке перевіряється. Моделі вихідних даних вибирають так, щоб вони повною мірою аналізували вимірювальні завдання, які охоплюють найширший діапазон можливих значень. Останні можуть містити: дані, які повністю охоплюють

діапазон можливих значень; дані, близькі до найбільших та найменших значень, а також декілька проміжних значень; конкретні значення вхідних змінних – ділянки стрімкого зростання або розриву похідних, нуль, один, гранично малі числові значення змінних тощо.

Якщо значення змінної залежать від значень іншої змінної, то випробування виконується для певних комбінацій цих змінних, таких як тотожність двох змінних або їх значна чи мала різниця, або нульові значення змінних. Метод випробування моделей вихідних даних простіше реалізувати, ніж класичний метод створення "стандартних" даних. Проте розвиток цього методу вимагає апріорної інформації про програмні алгоритми та способи реалізації програми, що не завжди відома.

Метод генерування "стандартних" даних як метод моделей вихідних даних застосовується як альтернатива методу зразкового програмного забезпечення або якщо неможливо перевірити окремі реалізовані функції програмного забезпечення. Однією з передумов створення методу "стандартних" даних є наявність апріорної інформації про відповідну вимірювальну задачу. "Стандартні" дані продукує спеціально розроблена генерувальна програма, тобто генератор "стандартних" даних оснований на заданих вихідних даних. Останні реалізуються мовою програмування або за допомогою стандартного математичного (статистичного) пакета програмного забезпечення. Вихідні дані для тестування, зокрема для генерації "еталонних" даних, формуються з урахуванням властивостей програмних алгоритмів. Метод генерації "стандартних" даних є альтернативою методу використання зразкового програмного забезпечення. З іншого боку, методу "стандартних" даних треба віддавати перевагу, оскільки він дешевший порівняно з іншими методами [24].

Якщо існує декілька програмних реалізацій того самого алгоритму вимірювання за відсутності зразкового програмного забезпечення, то доцільно виконувати перевірки, порівнюючи реалізації методу. Відповідно до цього, той самий набір "стандартних" даних подається на входи програмних продуктів і виконується порівняння відповідних результатів випробувань. Метод порівняння простий і не потребує додаткових програм. Проте програми з такими функціями доволі рідкісні.

Під час тестування або аналізу вихідного коду програмного забезпечення перевіряють такі пункти: відповідність структури алгоритмів наданій документації; відповідність записаних алгоритмів вибраній мові програмування; відповідність вибраних алгоритмів вимірювальним завданням (виявлення нестабільних алгоритмів).

Для перевірки відповідності структури алгоритмів наданій документації блок-схему алгоритмів можна скласти та порівняти з алгоритмами, описаними в документації. У разі розбіжностей між структурами алгоритмів виконується додатковий аналіз елементів блок-схем. Під час перевірки відповідності алгоритмів вибраній мові програмування встановлюють відповідність коду правилам програмування, наявність невизначених змінних і операторів, правильність організації циклів тощо. Адекватність вибраних алгоритмів вимірювальному завданню можна оцінити за допомогою математичного аналізу імплементованих програмних алгоритмів. Алгоритми реалізованих характеристик можуть додатково вивчатись, зокрема, на виконання оптимального аналізу числових методів розв'язання задач вимірювання. Для реалізації методу тестування на аналізі вихідного коду експерти з індустрії програмного забезпечення і з метрології повинні залучатись одночасно. Тому витрати на його імплементацію значно вищі порівняно з іншими методами.

Висновки

1. Упровадження кіберфізичних систем неможливе без легковстановлюваних розумних вимірювальних інструментів, що створюють можливості для реалізації широкого спектра бездротових сенсорних мереж. Переваги останніх – масштабована гнучка архітектура, висока роздільна здатність даних, наявність аплікаційно-адаптивних механізмів завдяки застосуванню у мережах розумних сенсорів, а також низка специфічних рис (самоперевірка, самодіагностика, самовалідизація, самоверифікація, самокоректування метрологічних характеристик тощо), які істотно покращують їхні метрологічні характеристики.

2. Подальше вивчення і вдосконалення бездротових сенсорних мереж досягається не лише завдяки основному програмному забезпеченню (яке встановлює розробник у розумних мережевих сенсорах) і проміжному програмному забезпеченню (отриманому через Інтернет і встановленому користувачем), запровадженню метрологічних перевірок та досліджень якості програмного забезпечення, але і забезпеченням масштабованості елементів мереж, гнучкості їхньої

топології та низки інших характеристик розумних метрологічних інструментів.

1. Platzer A. Carnegie Mellon University. Logical Foundations of Cyber-Physical Systems. http://symbolaris. com/logic/lfcps.html. 2. Mathas C. Smart Sensors – Not Only Intelligent, but Adaptable, Contributed By Electronic Products, 2011-09-29. 3. S. Y. Yurish, Sensors: Smart vs. Intelligent, Sensors & Transducers, Vol. 114, Issue 3, pp. 1–6, March 2010. 4. Deb B., Bhatnagar S., Nath B. A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management" // CiteSeerX, 2002. 5. Wayne W. Manges, Oak Ridge National Laboratory. 6. Ad-hoc Sensor Networks http://www.brunel.ac.uk/cedps/electronic-computer-engineering/research-activities/ wncc/ad-hoc-sensor-networks. 7. Sensor technology handbook, Editor-in-Chief Jon S.Wilson, SA, 2005. 8. Wobschall D. Smart Sensors for Smart Grid // Sensor Tech Forum, 2011. 9. What is smart grid sensor? – Definition from WhatIs.com. internet-ofthingsagenda. techtarget.com/definitio. 10. Change into UA. Start-up in energetics, 2015. 11. Salem H., Nader M., Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks // IEEE DISTRIBUTED SYSTEMS ONLINE 1541–4922, Published by the IEEE Computer Society, Vol. 7, No. 3, March 2006. 12. What is Middleware? // Middleware.org. Defining Technology, 2008, Retrieved 2013-08-11. 13. Hadim S., Mohamed N., "Middleware challenges and approaches for wireless sensor networks", IEEE Distributed Systems Online, vol.7, Issue, 2006, Retrieved March 4, 2009 from IEEE Distributed Systems. 14. Kazahaya M. A Mathematical Model and Error Analysis of Coriolis Mass Flowmeters // IEEE Transactions on Instrum. and Measurement/ – 2011. – Vol. 60, Issue 4. – P. 1163–1174. 15. Spitzer D. The Consumer Guide to Coriolis Mass Flowmeters // Seminar. Spitzer and Boyes, LLC, 2004. 16. The procedure for certification of software of measuring instruments. – Access to information: http://www.uazakon. com/. 17. WELMEC 7.1, Edition 2 Information document. Development of software requirements, Vienna, 48 p., 2005. 18. MI 3286 – 2010. Testing of software protection, with the determination of its level at the measuring instruments tests in order to type approval. – Moscow, 2010. – 33 p. (in Russian). 19. Microsoft Secure Software Development // The 3-rd Conference on minimization of software vulnerabilities in its development, www.microsoft.com/ru-ru/./mssdcon2013/. 20. What is CSSLP (certified secure software lifecycle. Searchsecurity.techtarget.com/./CSSLP-certifie. 21. ISSECO, the International Secure Software Engineering Council. www.isseco.org/ 22. Zlygosteva G. V., Muravyev S. V. The generalized model of a test procedure of measuring software // Bulletin of the Tomsk Polytechnic. – Univ, 2011. – Vol. 318, No. 4. – P. 62–67 (in Russian). 23. Kovalevskaya E. V. Metrology, software quality and certification. – Moscow, 2004 (in Russian). 24. Oleskiv O., Kunets I., Mykytyn I. Review of Techniques and Methods of Software Verification of Metrological Means. – Lviv: Publishing House of Lviv Politechnic, 2014. – No. 75 http://vlp.com. ua/node/12707 (in Ukrainian). 25. Yatsyshyn S., Stadnyk B., Lutsyk Ya., Buniak L. Handbook of Thermometry and Nanothermometry // IFSA Publishing, Spane, 2015. 26. Measurement Science Roadmap for Metal-Based Additive Manufacturing, May 2013, sponsored by NIST. 27. Ko S., Pan H., Grigoropoulos C. et al. All-inkjet-printed flexible electronics fabrication on a polymer substrate by low-temperature high-resolution selective laser sintering of metal nanoparticles // IOP Publishing, Nanotechnology. – 2007, Vol. 18, 345202, 8 pp., 28. Concept Laser Introduces QA Tool for In Process AM: QMmeltpool 3D Available Next Year, July 2, 2015.