

АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ УПРАВЛІННЯ ЕНЕРГОЕФЕКТИВНІСТЮ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ

В сучасному світі не володіння енергоресурсами займають домінуюче місце, а інноваційні технології ефективного використання енергоресурсів, відновлювані джерела енергії, засоби інтелектуалізації систем енергопостачання та енергоспоживання. Змінилась філософія конкурентного світу: від суспільства безлімітного споживання ресурсів – до сталого розвитку, «зеленого переходу», «розумного» енергоспоживання та в перспективі – активного застосування штучного інтелекту. В цих умовах зростає роль підвищення енергоефективності як головного чинника, який “тягне” за собою модернізацію всього підприємства або енергоефективне проектування нового, підвищення продуктивності виробництва та якості продукції, поліпшення екологічної ситуації. Принцип «Енергоефективність понад усе» повинен стати основним гаслом при відновленні економіки України та переході від сировинної моделі розвитку до галузей з високим ступенем переробки (машинобудування), які дозволяють, в тому числі, розвивати вітчизняний ВПК.

Найбільший вплив на ефективне використання електроенергії підприємством спричиняє енергоефективність електромеханічної системи (ЕМС) як основного споживача електроенергії (понад 70% у промисловості). Інтелектуалізація споживання електричної енергії передбачає створення цифрової системи управління енергоефективністю ЕМС в реальному часі. На жаль, на сьогодні утвердилась практика оцінювання енергоефективності ЕМС в кращому випадку аперіодично, наприклад, під час проведення енергетичного аудиту. Тому актуальним завданням є створення інтелектуальних систем управління ефективністю споживання електроенергії ЕМС та контролю за чинниками, які на неї впливають: якість електричної енергії, завантаження обладнання та технічний стану. Сучасний розвиток вимірювальної та обчислювальної техніки дозволяють створювати подібні системи, які загалом складаються з технічної (давачі струму, напруги, технологічних параметрів виробничого процесу) та програмної частин.

Ключові слова: енергетичний менеджмент, програмне забезпечення, електромеханічна система, енергоефективність, клієнт-серверна архітектура.

Вступ. Довоєнна сировинна модель економіки України, що базувалась на експорті непереробленої продукції сільського господарства, залізної руди та металевих напівфабрикатів не дає можливості реалізувати високий науковий та технічний потенціал України, де понад 80% населення охоплено вищою освітою. До речі, за цим показником Україна випереджає переважну кількість країн розвинутого Заходу, зокрема, навіть Німеччину [1]. Сировинна модель диктує повну залежність від тенденцій світової економіки і не потребує залучення великої кількості професійної та високоосвіченої робочої сили. Будь-яка фінансова криза спричиняє суттєве зниження попиту на сировинні товари та глибоке економічне падіння в державі, знецінення національної валюти. При цьому сировинна економіка продукує товари з “негативним потенціалом”, коли собівартість кожної наступної видобутої тони залізної руди (тільки експорт якої у 2021 році сягнув 7 млрд. доларів) чи кам’яного вугілля, дорожче попередньої, і головне – не може забезпечити розвиток ВПК України. Дисбаланс між занепадом сектору глибокої переробки сировини (машинобудування) та довоєнним розвитком, пов’язаними із експортом ІТ-послуг, сфери якісних побутових послуг “24/7”, торгівлі та електронної комерції не є винятком із загальної сировинної моделі – це, як правило, – атрибут країн з бідним населенням, таких як Україна чи Індія.

Післявоєнне відновлення України повинно базуватись передусім на розвитку машинобудівного комплексу із залученням закордонних інвестицій та передових технологій, що включають повну інтелектуалізацію виробничих процесів.

Постановка задачі дослідження. Дотепер в енергетичному менеджменті деталізоване управління раціональним використанням енергії не поширювалося на конкретного технологічного споживача, його режими роботи. В першу чергу розглядалось підприємство в цілому або окрема технологічна лінія. Приймались до уваги загальні показники по підприємству - питомі витрати енергоресурсів на виробництво товарів або надання послуг. Контроль конкретного обладнання, наприклад, ЕМС, не був постійним і безперервним та відбувався за фактом аварійної ситуації, а не в реальному часі, коли неефективне споживання електроенергії мало місце значний проміжок часу.

Останніми роками енергетичний менеджмент набув нових завдань, а саме, – він став поширюватися на кінцевого споживача. Це є сучасним підходом у методології енергетичного менеджменту – запроваджується високий ступінь деталізації, контроль енергетичної ефективності ЕМС і, відповідно, – оперативне реагування на погіршення енергоефективності. Мета полягає у забезпеченні енергоефективної роботи кожної окремої одиниці обладнання в реальному часі. Ці нові завдання пов'язані з постійним суттєвим удорожчанням електричної енергії, коли основна складова вартості життєвого циклу ЕМС – витрати на електричну енергію – становлять в середньому 85% від усіх витрат за життєвий цикл обладнання [2, 3]. Розв'язати такі задачі можна лише шляхом охоплення споживачів електроенергії сучасними засобами інтелектуалізації: цифровими давачами струму, напруги, технологічних параметрів виробництва, мікроконтролерів, потужними робочими станціями для збору та обробки інформації з використанням можливостей штучного інтелекту, сучасного спеціалізованого програмного забезпечення (рис. 1).

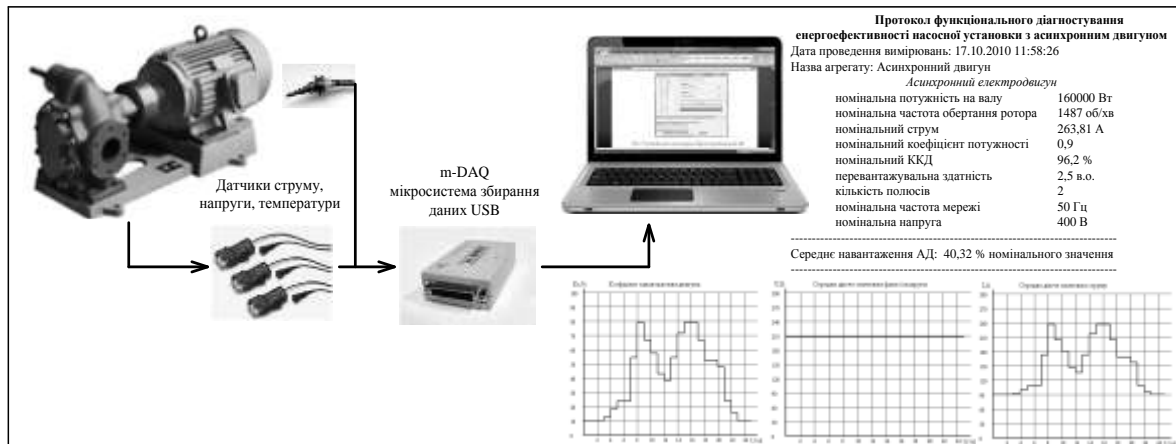


Рисунок 1 - Структурна схема мобільного програмно-апаратного комплексу для контролю енергоефективності ЕМС

Основні матеріали дослідження. Система інтелектуалізації управління споживанням електроенергії ЕМС має п'ять компонент: вимірювальна (давачі), обчислювальна (мікроконтролери, сервери, ПК, смартфони), методична (методика розрахунків параметрів ЕМС), комунікаційна (мережа по якій передаються дані по протоколу HTTPS) та програмна. Вимірювальна та методична, та часткова обчислювальна компоненти висвітлені у монографії [4].

Що стосується програмної частини, то найкраще завданням створення систем інтелектуалізації споживачів електричної енергії відповідає клієнт-серверна архітектура – як один із архітектурних шаблонів програмного забезпечення та домінуюча концепцією у створенні розподілених мережних застосунків, що передбачає взаємодію та обмін даними між ними (клієнт-сервер). Така архітектура передбачає три основні компоненти [5]:

- сервер або набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них (сервер баз даних та сервер обробки запитів / відповідей);
- клієнт або набір клієнтів, які використовують сервіси, що надаються сервером обробки запитів / відповідей (браузер ПК, застосунок у смартфоні);
- мережа, яка забезпечує взаємодію між клієнтами та серверами (протоколи HTTP або HTTPS).

Клієнти функціонують паралельно і незалежно один від одного. Типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів. Клієнти мають знати адресу серверу обробки запитів / відповідей, але можуть не мати жодного уявлення про існування інших клієнтів.

Модель клієнт-серверної взаємодії визначається перш за все проблемою розподілу функцій між клієнтом та сервером: надсилання та обробки запитів / відповідей, математичних розрахунків, керування поточними даними застосунку, їх кешування, аутентифікації, авторизації та ідентифікації користувачів, верифікації даних на різних етапах роботи програмного забезпечення, відслідковування змін вимірювальних даних та реагування на ці зміни, формування зовнішнього інтерфейсу для взаємодії з користувачем в залежності від прав та дозволів на доступ до інформації тощо.

Логічно можна відокремити три рівні операцій (рис.2):

- рівень подання даних, який по суті являє собою інтерфейс користувача і відповідає за представлення інформації користувачеві (енергоменджеру) і введення від нього запитів (шар клієнта);
- прикладний рівень, який реалізує основну логіку застосунку і на якому здійснюється необхідна обробка інформації (шар бізнес-логіки).
- рівень управління даними, який забезпечує зберігання даних та доступ до них (шар даних) [6].

Таким чином постає задача розподілення складних математичних обчислень (рис.3) та інших завдань між клієнтською та серверною частинами програмного забезпечення. Одним з вирішенням цієї задачі є архітектура застосунків за програмним паттерном Model-View-Controller (MVC). MVC – це шаблон у розробці програмного забезпечення, який зазвичай використовується для реалізації та відокремлення інтерфейсів: користувача (енергоменеджера), даних і та логіки управління застосунком та даними. Це підкреслює розмежування між бізнес-логікою програмного забезпечення та зовнішнім відображенням інформації [7].

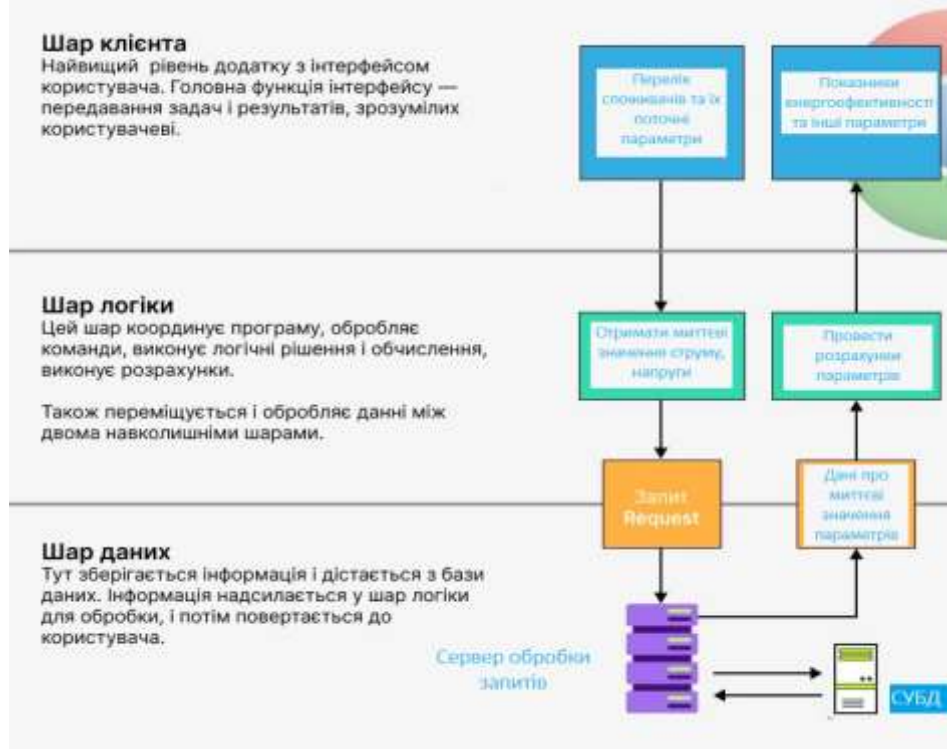


Рисунок 2 - Принципова схема тришарової структури клієнт-серверної архітектури

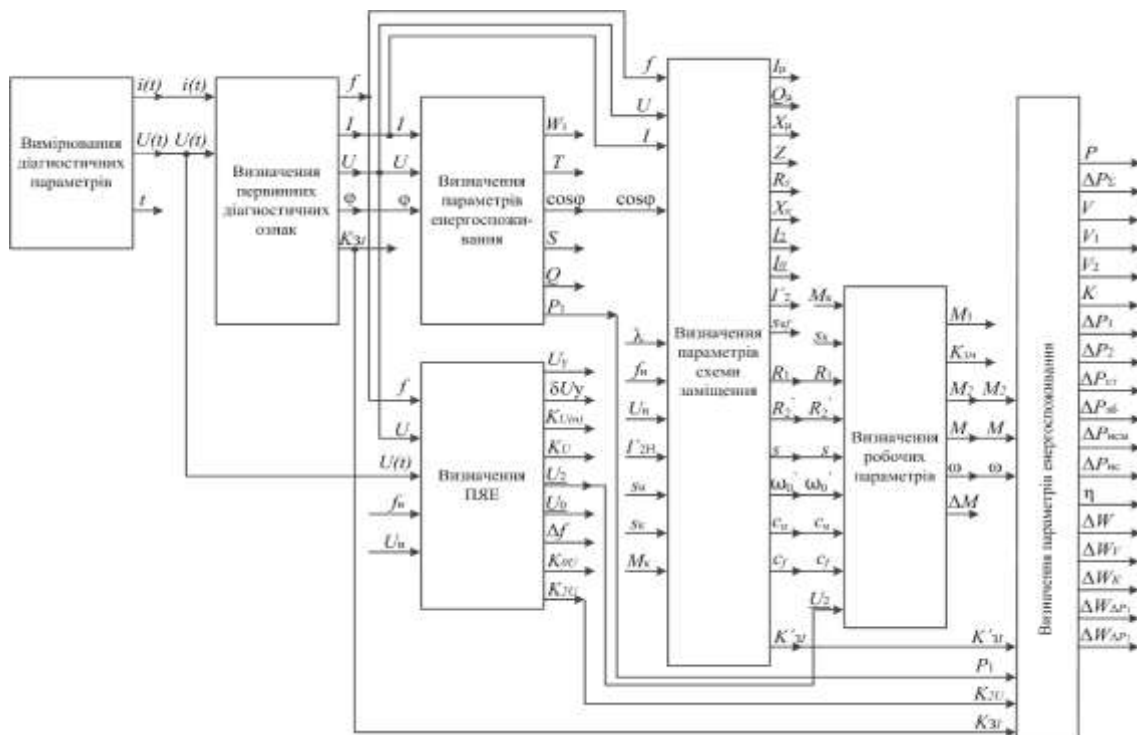


Рисунок 3 - Структурна схема визначення параметрів процесу перетворення електроенергії ЕМС

Три шари шаблону проектування програмного забезпечення MVC можна описати наступним чином (рис. 4):

1. “Модель” (Model): управляє даними та бізнес-логікою.
2. “Вигляд” (View): керує макетом web-сторінки та її відображенням.
3. “Контролер” (Controller): направляє http запити (request) від користувача (енергоменеджера) до необхідного модуля (програмної функції, методів класу) “Моделі” та верифікує відповіді від сервера (response) для модуля “Вигляд”, який відповідає за відображення результатів користувачеві.

“Контролер” забезпечує зв'язок між користувачем та системою, контролює та спрямовує дані від користувача до системи та навпаки, використовує компоненти “Модель” та “Вигляд” для реалізації необхідної дії [6, 8, 9].

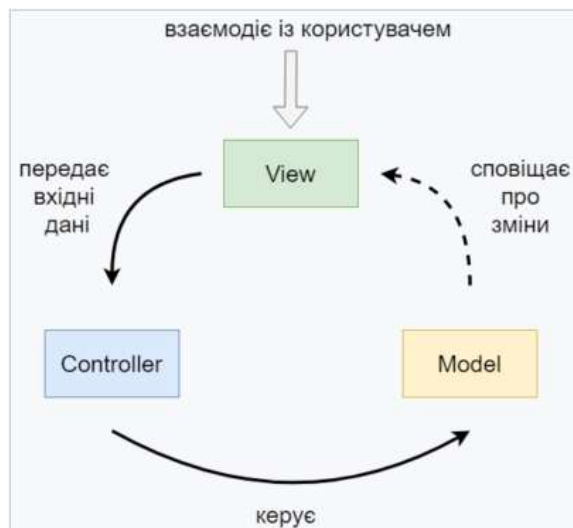


Рисунок 4 - Діаграма взаємодії між компонентами шаблону MVC

У випадку застосування такої моделі при написанні програмного забезпечення для мікроконтролерів (рис. 1), розподіл задач виглядатиме наступним чином:

1. “Контролер” відповідає за отримання та опрацювання сигналів від вхідних інтерфейсів
2. Компонента “Модель” у випадку мікроконтролерів по суті реалізує поведінку пристрою в залежності від поточного стану та інформації отриманої з компоненти “Контролер”.
3. Компонента “Вигляд”, - у випадку мікроконтролерів, - це пристрій візуалізації інформації (рідкокристалічний дисплей) чи певний виконавчий пристрій.

Енергоменеджер для отримання інформації щодо стану енергоефективності споживачів на підприємстві взаємодіє з інтерфейсом web-сторінки застосунку - клієнтською складовою в браузері, що написана на мові програмування JavaScript з використанням бібліотеки React, Angular чи Vue. В результаті чого браузер надсилає HTTP запити: GET, POST, PUT та інші.

Основна програмна функція “Контролера” в цьому випадку – викликати функції, методи в необхідних об’єктів, управляти доступом до ресурсів виконання завдань, які поставлені користувачем - енергоменеджером. За поточні дані застосунку, які зберігаються в оперативній пам’яті комп’ютера та станом програми відповідає окрема бібліотека, наприклад, Redux. За зовнішнім виглядом web-сторінки користувача може відповідати також зовнішня бібліотека стилей, наприклад Bootstrap.

“Модель” – набір алгоритмів, які використовуються для роботи з даними, що становлять бізнес-логіку програми. Проектування програми завжди починається з побудови моделей об’єктів, якими вона оперує. Схематичне зображення прикладу алгоритму моделі наведено на рис. 2

“Модель” оброблює для “Контролера” набір даних, на які зробив запит енергоменеджер (поточне ККД споживача, спожита ним електроенергія, графік електричного навантаження тощо). Таким чином “Модель” містить найбільш важливу частину логіки програмного забезпечення – методи розрахунку та вирішує завдання, поставлені енергоменеджером (розрахунок втрат, рішення оптимізаційних завдань щодо споживання електроенергії в залежності від технологічних параметрів виробництва тощо). “Контролер” містить переважно організаційну логіку та оброблює запити користувача програми - енергоменеджера, тому ці завдання виконуються, як правило, на стороні сервера з використанням програмного забезпечення, наприклад, Node.js, бібліотеки Express та хмарних сервісів провідних іт-компаній світу. Важливо зазначити, що клієнтська та серверна частина пишеться на одній мові програмування - JavaScript (Typescript) за допомогою спеціалізованих фреймворків та бібліотек, що значно

спрощує розробку. Мова програмування JavaScript (Typescript) останні роки є найбільш розповсюдженою у світі, достатньо гнучкою та дозволяє писати програмний код як в функціональній так і в об'єктно-орієнтовній парадигмі програмування, або у змішаному стилі.

Шар програмного забезпечення "Вигляд" відповідає за візуалізацію та представлення кінцевих результатів користувачеві, які отримані з "Моделі". Він може бути шаблоном HTML-коду, що заповнюється даними на стороні сервера. Але найефективніше це робити на стороні клієнта та, відповідно, повертати з сервера дані у форматі JSON (JavaScript Object Notation), з якими може працювати як браузер комп'ютера так і застосунок для смартфона. Передача саме даних, а не кінцевого HTML коду є значною перевагою, адже зменшує навантаження на сервер, мережу та є універсальним рішенням як для ПК так і смартфона.

Наприклад, якщо користувач застосунку відкриє список споживачів електричної енергії та вибере опцію для перегляду поточних значень ККД, всю послідовність дій можна описати у вигляді наступних кроків:

1. Користувач натискає на посилання «споживачі» і браузер надсилає GET-запит на endpoint (сервер обробки запитів / відповідей), наприклад, "/consumptions/efficiency".

2. "Контролер" верифікує запит користувача, який повинен пройти авторизацію та аутентифікацію (як правило, за допомогою cookies або jwt-токенів). Потім "Контролер" звертається до відповідного модуля "Моделі" та надсилає запит провести розрахунки, які необхідні користувачеві.

3. "Моделі" верифікує http-запит, звертається до серверу баз даних (наприклад, MongoDB, що працює у зв'язці з Mongoose - засобом моделювання програмних структур даних - об'єктів та їх зв'язку з сервером на Node.js), в якому зберігаються результати вимірювання та отримує набір даних в форматі JSON про миттєві значення струму, напруги та технологічних параметрів виробничого процесу. На основі цих даних "Моделі" проводить розрахунки параметрів, що запитує користувач.

4. "Контролер" одержує кінцевий результат розрахунків та верифікує його.

5. "Контролер" визначається з тим, які дані надати користувачеві у форматі JSON (текстовий файл з даними представленими у вигляді програмної структури даних – об'єкт) - зі списком споживачів та їх поточного ККД, або сторінку з помилкою, якщо вона мала місце.

6. Сервер віддає клієнту JSON файл, сформований "Контролером" за певним шаблоном.

7. Компонента "Вигляд" створює та заповнює HTML-сторінку (назва споживача, ККД, інші показники, елементи інтерфейсу сторінки) вже на стороні клієнта, наприклад, з використанням бібліотеки React та відображає користувачеві кінцевий результат (енергоменеджеру).

У мобільних програмах та веб застосунках з використанням, наприклад, бібліотеки, Redux використовуються події та механізми підписки на ці події (програмні паттерни "диспетчер", "спостерігач"). При такому підході "Вигляд" підписується на зміни "Моделі" (наприклад, зміни вимірювальних даних). Далі, коли відбувається подія (наприклад, оновлення вимірювальних даних), викликається "Контролер". Він надає "Моделі" команду щодо проведення нових розрахунків та їх відображення засобами компоненти "Вигляд".

Таким чином, якщо вимірювальні дані оновилися, компонента "Моделі" генерує подію про зміну цих даних. Всі складові інтерфейсу "Вигляд", які підписалися на цю подію (для яких важлива зміна саме цих даних), отримують оновлені результати розрахунків та оновлюють дані у інтерфейсі користувача.

Висновки. Перспективним інструментом енергоменеджменту є впровадження інтелектуальних систем управління енергоефективністю ЕМС на основі клієнт-серверної архітектури з програмним розділенням шарів бізнес-логіки, даних та представлення результатів - визначення енергоефективності, інших параметрів для оперативного реагування на збільшення електроспоживання, погіршення технічного стану чи порушення технологічного режиму. Оптимізація електроспоживання ЕМС, прийняття обґрунтованих рішень щодо подальшої їх експлуатації шляхом виявлення неекономічних, неефективних, аварійних режимів роботи, прихованих дефектів засобами інтелектуальних систем управління енергоефективністю передбачає також наявність інформації про миттєві значення струму й напруги ЕМС, розрахунок параметрів за діагностичними й еталонними моделями в режимі реального часу; виявлення відхилень параметрів від встановлених.

Інтелектуальні систем управління енергоефективністю ЕМС з використанням програмного архітектурного паттерну MVC дозволяє впровадити сучасні інформаційні smart-технології у систему енергетичного менеджменту підприємства та автоматизувати виконання таких складових стандарту ДСТУ ISO 50001:2020, як встановлення енергетичної базової лінії, контроль індикаторів енергоефективності, здійснення моніторингу та необхідних вимірювань для визначення рівня енергоефективності, проведення енергоаналізу, енергопланування та формування технічних звітів тощо.

Список використаної літератури

1. Стратегія розвитку вищої освіти України на 2021-2023 роки. Міністерство освіти на науки України. -Київ: 2020., -71 с. URL: <https://mon.gov.ua/storage/app/media/rizne/2020/09/25/rozvitku-vishchoi-osviti-v-ukraini-02-10-2020.pdf>
2. Прокопенко В.В. Енергетичний аудит: Навчальний посібник / В.В. Прокопенко, О.О. Закладний, П.В. Кульбачний. – Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2018. – 400 с. URL: <https://ela.kpi.ua/handle/123456789/56296>
3. Закладний О.М. Електропривод: навч. посіб. / О.М. Закладний, В.В. Прокопенко, О.О. Закладний. –К.: НТУУ «КПІ», 2007. – 316 с URL: <https://ela.kpi.ua/handle/123456789/56294>
4. Закладний О.О. Функціональне діагностування енергоефективності електромеханічних систем: Монографія / О.О. Закладний – К.: Видавництво «Лібра», 2013. – 195 с.: ил URL: <https://ela.kpi.ua/handle/123456789/50554>
5. Клієнт-серверна архітектура. Вікіпедія: Вільна Енциклопедія URL: https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура
6. Підхід MVC. Модуль 3.Java Professional. URL[^]<https://javarush.com/ua/quests/lectures/ua.questservlets.level14.lecture02>
7. MVC. MDN Web Docs Glossary: Definitions of Web-related terms. URL: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>
8. Триярусна архітектура. Вікіпедія: Вільна Енциклопедія. URL: https://uk.wikipedia.org/wiki/Триярусна_архітектура
9. Модель-вид-контролер. Вікіпедія: Вільна Енциклопедія. URL: https://uk.wikipedia.org/wiki/Модель_вид_контролер

O. Zakladnyi¹, Cand.Sc. (Eng.), Assoc. Prof., ORCID 0000-0003-2813-3692

V. Prokopenko¹, Cand.Sc. (Eng.), Assoc. Prof., ORCID 0000-0002-5518-5802

¹**National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”**

SOFTWARE ARCHITECTURE OF INTELLIGENT SYSTEMS FOR MANAGING ENERGY EFFICIENCY OF ELECTROMECHANICAL SYSTEMS

In the modern world, it is not the possession of energy resources that occupy a dominant place, but innovative technologies for the efficient use of energy resources, renewable energy sources, means of intellectualization of energy supply systems and energy consumption. The philosophy of the competitive world has changed: from a society of unlimited consumption of resources to sustainable development, "green transition", "smart" energy consumption and, in the future, the active use of artificial intelligence. Under these conditions, the role of increasing energy efficiency is growing as the main factor that "entails" the modernization of the entire enterprise or the energy-efficient design of a new one, increasing production productivity and product quality, and improving the environmental situation. The principle of "Energy efficiency above all" should become the main slogan in the recovery of Ukraine's economy and the transition from the raw material model of development to industries with a high degree of processing (mechanical engineering), which will allow, among other things, the development of the domestic military-industrial complex.

The greatest impact on the efficient use of electricity by the enterprise is caused by the energy efficiency of the electromechanical system (EMS) as the main consumer of electricity (more than 80% in industry). Intellectualization of electrical energy consumption involves the creation of a digital EMS energy efficiency management system in real time. Unfortunately, today the practice of assessing the energy efficiency of EMS has been established at best periodically, for example, during an energy audit. Therefore, the urgent task is to create intelligent systems for managing the efficiency of EMS electricity consumption and monitoring the factors that affect them: the quality of electrical energy, equipment loading and the technical condition of EMS. Modern developments in measuring and computing technology make it possible to create such systems, which consist of technical (current, voltage sensors, technological parameters of the production process) and software parts.

A promising tool for energy management is the introduction of intelligent EMS energy efficiency management systems based on client-server architecture with software separation of layers of business logic, data and presentation of results - determination of energy efficiency, other parameters for prompt response to an increase in power consumption, deterioration of technical condition or violation of the technological regime. Optimization of EMC power consumption, making informed decisions on their further operation by identifying

uneconomical, inefficient, emergency operating modes, hidden defects by means of intelligent energy efficiency management systems also provides for the availability of information on instantaneous values of EMS current and voltage, calculation of parameters according to diagnostic and reference models in real time; detection of deviations of parameters from the established ones.

Intelligent EMC energy efficiency management systems using the MVC software architectural pattern allows you to introduce modern smart information technologies into the energy management system of the enterprise and automate the implementation of such components of the ISO 50001:2020 standard as the installation of an energy baseline, control of energy efficiency indicators, monitoring and necessary measurements to determine the level of energy efficiency, energy analysis, energy planning and formation of technical reports, etc.

Keywords: *energy management, software, electromechanical system, energy efficiency, client-server architecture.*

References

1. Strategy for the Development of Higher Education in Ukraine for 2021-2023. Ministry of Education and Science of Ukraine. -Kyiv: 2020., -71 p. URL: <https://mon.gov.ua/storage/app/media/rizne/2020/09/25/rozvitku-vishchoi-osviti-v-ukraini-02-10-2020.pdf>
2. Prokopenko, V.V., Zakladnyi, P.V. Kulbachnyi. – Kyiv: KPI them. Igor Sikorsky, Polytechnic Publishing House, 2018. – 400 p. URL: <https://ela.kpi.ua/handle/123456789/56296>
3. Zakladnyi O.M. Elektroprivod [Electric drive]. / O.M. Zakladnyi, V.V. Prokopenko, O.O. Zaladnyi. – K.: NTUU "KPI", 2007. – 316 URL: <https://ela.kpi.ua/handle/123456789/56294>
4. Zakladnyi, O.O. Functional diagnostics of energy efficiency of electromechanical systems: Monograph / O.O. Zakladnyi – K.: Libra Publishing House, 2013. – 195 p.: ill URL: <https://ela.kpi.ua/handle/123456789/50554>
5. Client-server architecture. Wikipedia: The Free Encyclopedia URL: https://uk.wikipedia.org/wiki/Client-server_architecture
6. MVC approach. 3.Java Professional module. URL: <https://javarush.com/ua/quests/lectures/ua.questservlets.level14.lecture02>
7. MVC. MDN Web Docs Glossary: Definitions of Web-related terms. URL: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>
8. Three-tiered architecture. Wikipedia: The Free Encyclopedia URL: https://uk.wikipedia.org/wiki/Трияруснаа_архітектура
9. Model-view-controller. Wikipedia: The Free Encyclopedia URL: https://uk.wikipedia.org/wiki/Модель_вид_контролер

Надійшла: 7.04.2024

Received: 7.04.2024