

ІНФОРМАЦІЙНА СИСТЕМА КОНТРОЛЮ ЯКОСТІ ПОЛІЕТИЛЕНУ В ЦИРКУЛЯРНІЙ ЕКОНОМІЦІ

Микола Баран¹, Мирослава Бублик¹, Любомир Чирун², Марина Шевченко³, Софія Чирун¹

¹ Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж, Львів, Україна

² Львівський національний університет імені Івана Франка,
кафедра прикладної математики, Львів, Україна

³ Університет Оснабрюка, кафедра міжнародної економічної політики,
Оснабрюк, Німеччина

E-mail: mykola.baran.kn.2017@lpnu.ua, ORCID: 0000-0002-8199-0530

E-mail: Myroslava.I.Bublyk@lpnu.ua, ORCID: 0000-0003-2403-0784

E-mail: Lyubomyr.Chyrun@lnu.edu.ua, ORCID: 0000-0002-9448-1751

E-mail: mshevchenko@uni-osnabrueck.de, ORCID: 0000-0003-2165-9907

E-mail: sofia.chyrun.sa.2022@lpnu.ua, ORCID: 0000-0002-2829-0164

© Баран М. І., Бублик М. І., Чирун Л. В., Шевченко М. М., Чирун С., 2023

Система контролю якості поліетилену – це сукупність методів і засобів перевірки й регулювання компонентів, що визначають рівень якості продукції, а також технічного моніторингу на всіх стадіях виробничого процесу поліетилену. Уперше полімеризацію поліетилену в кінці XIX ст. здійснив учений Густавсон. Протягом довгих років поліетилен створювали в малих обсягах, але в 1938 р. процес промислового виробництва освоїли в Англії. Тоді метод полімеризації був ще недосконалий та потребував додаткових засобів контролю за виробництвом. У наш час кожне підприємство застосовує різні методи для контролювання виробництва, однак, з огляду на складні хіміко-технологічні процеси, необхідність ефективнішої та ціліснішої системи, яка контролюватиме якість кінцевої продукції та забезпечуватиме економію сировини, все ще актуальна. Мета дослідження: створення інформаційної системи контролю якості управління системами автоматизації із використанням віддаленого доступу та системи штучного інтелекту. Об’єкт дослідження: технічні особливості впровадження інформаційної системи контролю якості для підприємства із виробництва поліетилену. Предмет дослідження: інформаційна система контролю якості виробництва поліетилену із елементами технології штучного інтелекту. В роботі розглянуто питання створення інформаційної системи контролю якості поліетилену, яка уможливить ефективне і безпечне управління виробничим процесом, забезпечить високий функціонал та кросплатформність за допомогою вебтехнологій та засобів штучного інтелекту. У першому розділі розглянуто теоретико-методологічні основи дослідження систем контролю якості, у другому виконано системний аналіз об’єкта дослідження. Із урахуванням результатів дослідження створено інформаційну систему контролю якості виробництва поліетилену, призначення якої полягає у встановленні відповідності продукції та процесів вимогам нормативно-технічної документації, зразкам-еталонам, наданні інформації про перебіг виробничого процесу та підтримку його стабільності, захисті даних підприємства, виявленні дефектної продукції на ранніх етапах, запобіганню випуску недоброякісної продукції.

Ключові слова: інформаційна система контролю якості; контрольна панель; вебтехнології; засоби штучного інтелекту; поліетилен; хімічна промисловість.

Вступ

Проблема контролю якості виробництва та економії сировини все ще не до кінця вирішена. Тому виникла необхідність побудови інформаційної системи контролю якості виробництва із застосуванням елементів технології штучного інтелекту, які допоможуть не тільки завчасно повідомити про проблеми, але і контролювати технологічний процес, для постійної підтримки якості виробів на певному рівні. Інформаційні системи контролю якості виробництва широко використовують для процесу виробництва поліетилену високого тиску, оскільки він передбачає складні хіміко-технологічні процеси, запобігання поломкам технологічного обладнання, зменшення виробничих витрат і підвищення рівня безпеки праці на підприємстві. Впровадження інформаційних систем у виробничий процес дуже корисне для підприємств хімічної промисловості, вирішує комплекс завдань, пов'язаних із контролем якості продукції, економією сировини і управлінням виробництвом. Система контролю якості виробництва поліетилену дає змогу підвищити продуктивність реактора, забезпечити високу швидкість процесу, знизити шкідливість виробничої зони, зменшити собівартість поліетилену, збільшити вихід готового продукту і поліпшити його якість. Задіяння інформаційних систем, які на основі штучного інтелекту контролюють складні хіміко-технологічні процеси, чутливі до будь-яких відхилень, робить виробництво поліетилену керованим, передбачуваним і контрольованим.

Мета статті – аналіз процесів створення інформаційної системи контролю якості виробництва для виконання завдань сповіщення, контролю, управління системами автоматизації з використанням віддаленого доступу та системами штучного інтелекту.

Завдання дослідження:

- Розглянути основні елементи базової системи контролю якості виробництва й інформаційні системи, визначити можливість їх використання для створення комплексної системи контролю якості продукції.
- Аналіз інженерних та програмних рішень зі створення системи контролю якості виробництва поліетилену, без прив'язки до платформи.
- Визначення необхідних функцій та команд системи.
- Описання системних вимог для розроблення системи контролю якості виробництва.
- Визначення необхідних програмних засобів реалізації інформаційної системи.

Об'єктом дослідження є технічні особливості впровадження системи контролю якості продукції для підприємства із виробництва поліетилену. Предмет дослідження – інформаційна система контролю якості продукції для виробництва поліетилену із елементами технології штучного інтелекту. Новизна роботи полягає в розробленні принципово нової системи контролю якості поліетилену на основі вебтехнологій, яка за допомогою елементів штучного інтелекту буде самостійно контролювати процес виробництва для створення якісного кінцевого продукту та завдяки цьому забезпечуватиме економію сировини. Окрім цього інформаційна система не буде прив'язана до конкретного робочого місця користувача, що надасть їй підвищену мобільність. Практичне значення одержаних результатів полягає у тому, що вони становлять як науково-теоретичний, так і практичний інтерес. Матеріали виконаного аналізу стануть у нагоді для подальшого вивчення специфіки залучення інформаційних технологій у галузь хімічної промисловості. Розроблювана інформаційна система буде ефективною для подальшої оптимізації виробничого процесу, поліпшення якості продукції та економії сировини. Цю інформаційну систему також можна буде використовувати у взаємодії із іншими інформаційними системами або для підготовки відповідних спеціалістів.

Аналіз останніх досліджень та публікацій

Система контролю якості продукції – це цілісна сукупність процесів, ресурсів, відповідних процедур, організаційної складової, що уможливають підтримання міцних зв'язків між всіма ланками управління та підприємств на всіх рівнях реалізації та виробництва, а також здійснення загального управління якістю продукції та послуг. Для забезпечення необхідної високої якості продукції чи послуг потрібна певна чітка система управління якістю, яка урахуватиме науково-технічні досягнення, орієнтуватиметься на запити ринку і стимулюватиме сумлінну працю всього колективу підприємства [1].

Система контролю якості продукції – узагальнювальний термін для декількох типів інформаційних систем. Є різні підходи до побудови систем контролю якості виробництва, а саме: системи, побудовані за принципом світлосигнальної панелі із застосуванням механічних пристроїв контролю (перемикачів, кнопок, сигнальних ламп, маячків); системи із застосуванням панелі оператора (комбіноване використання механічних пристроїв контролю та панельних комп'ютерів), а також системи, що являє собою НМІ на базі автоматизованих робочих місць (АРМ), тобто персональний комп'ютер, телефон або планшет з розгорнутою системою контролю якості продукції [2, 3]. У роботі розглянуто розроблення системи контролю якості поліетилену за принципом НМІ систем, із підтримкою мультиплатформності та використанням web-технологій.

Робіт, які стосуються розроблення і дослідження інформаційних систем контролю якості продукції та систем віддаленого управління, доволі багато як в міжнародному просторі, так і в національному сегменті. У статті [4] описано наглядний контроль якості продукції, технології збирання даних, вимоги до інтерфейсу, розглянуто нові та відкриті технології, їх вплив на збільшення можливостей розробки SCADA/HMI. Наведено практичні програми SCADA/HMI для DCAI. У статті [5] подано огляд функцій SCADA та основ функціонування систем SCADA, зокрема короткий опис ключового інтерфейсу людина-машина. Обговорено кілька ключових питань та проблем сучасних систем SCADA, тобто стандарти повідомлень, тестування їх продуктивності та застарілості системи. Проаналізовано сучасні тенденції розподіленого опрацювання, вдосконалення інтерфейсу людина-машина, стандартних систем, розумних RTU та стандартних програмних забезпечень. Автор статті [6] зосереджує увагу на тому, що якість продукції, хоч і не має оцінювального критерію, однак є конкретною перевагою і залежить насамперед від кваліфікації персоналу і його обізнаності в технологіях виробництва. Якість формує репутацію і діловий імідж як фахівця, так і компанії, представником якої він є. У статті [7] виконано чіткий, детальний аналіз сучасного стану справ організації контролю якості продукції на підприємствах, передбачено необхідність аналізу всього організаційного процесу контролю якості продукції, дослідження системи, а також чіткого дослідження інформаційних потоків контролю якості продукції, що виготовляється на підприємстві. Проте у всіх згаданих вище роботах не вирішено завдання якісного моніторингу та запущеного контролю процесу виробництва без задіяння персоналу, зменшення затрат на сировину та фінансових витрат, а також зменшення складності навчання роботи із даними системами. Також, із урахуванням цих робіт, актуалізується розроблення системи, яка не прив'язана до певного пристрою чи панелі, а побудована на принципі мультиплатформності, що надає мобільності роботі оператора системи.

Для забезпечення потрібної якості продукції чи послуг необхідна ефективна система управління якістю, що враховуватиме усі науково-технічні досягнення та орієнтуватиметься на запити ринку. Починаючи із 50-х років XX ст. у світі розпочалося запровадження систем управління якістю продукції на підприємствах різних галузей. Однією з перших систем була система бездефектного виробництва продукції (БВП). Метою системи було створення необхідних умов виробництва, за яких продукція відповідала б усім зазначеним нормативним документам і її приймали із першого подання. Одним із основних критеріїв оцінювання якості продукції був заданий відсоток продукції, зданої з першого разу. Відповідно, чим більший відсоток зданої продукції, тим більші були моральні, а головне матеріальні заохочення [8].

Прогресивнішою була Львівська система бездефектної роботи (СБП), яка ставила за мету випуск максимально бездефектного продукту відмінної якості. Варто зауважити, що система передбачала заохочення роботи всіх працівників: від робітника виробництва до працівника управління. Система враховувала якість праці, а також прорахунки кожного працівника підприємства. У наш час системи контролю якості продукції використовують у різних сферах людської діяльності. Їх активно застосовують для автоматизації окремих агрегатів, установок або цілого технологічного процесу [9].

Сучасна НМІ-система – це спеціалізоване програмне забезпечення чи мікропроцесорний пристрій із дисплеєм. Вона поставляється з попередньо встановленим процесом виробництва і середовищем виконання проєктів, призначеними для користувача НМІ. Цим панель оператора відрізняється від панельних промислових комп'ютерів, на які необхідно додатково встановлювати програмні пакети й інші застосунки – як правило, платні. Для того, щоб краще розуміти всі переваги та недоліки різних способів реалізації систем контролю якості, доцільно ознайомитись з основними засобами реалізації систем та виділити у них найпопулярніші системи, пропонувані на ринку. Серед найефективніших та популярних систем, які застосовують на підприємствах хімічної промисловості, можна виділити такі:

1. Кнопкові панелі SIMANTIC можна використовувати скрізь, де потрібні кнопки і лампи, наприклад, у панелях управління, машинах і установках. Вони можуть також входити в конструкції спеціальних машин, для розроблення стандартних панелей оператора, які за рахунок цього можна доволі гнучко і модульно розширити із використанням додаткових кнопок і світлодіодів, що підключаються до вбудованих у панель дискретних входів і виходів. Готові до використання кнопкові панелі підключаються до контролерів SIMATIC через мережі MPI, Profibus DP або Profinet. Стандартна конфігурація гарантує безпосередню готовність для використання відразу ж після підключення і конфігурування, завдяки цьому можна зекономити до 90 % часу порівняно зі стандартним монтажем. Зміна параметрів може активізувати різні опції PP. Управління машиною стає простішим і швидшим для інженера-розробника, оператора і кінцевого користувача [10].

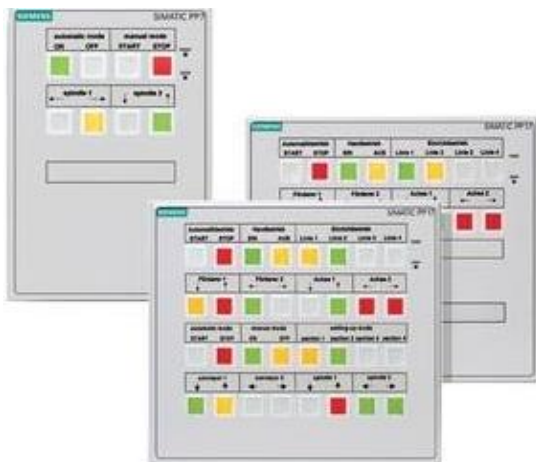


Рис. 1. Кнопкова панель SIMATIC PP



Рис. 2. Сенсорна панель YOKAGAWA

Переваги SIMATIC PP:

- Кнопкова панель (PP) забезпечує стандартні функції, які необхідні для традиційно розробленої панелі оператора.
- Повністю або частково “налаштовані” кнопки, завдяки чому спрощується розгортання системи на підприємстві.
- Двоколірні світлодіоди у всіх кнопках.
- Додаткові цифрові входи і виходи для розширень системи.

- Заготовки для 22 мм додаткових елементів типу кнопка, лампа тощо.
- Централізований вхід, для запобігання несанкціонованому введенню.
- Вбудована перевірка ламп і кнопок, вибрана через параметри.
- Висока надійність управління системою, що надзвичайно важливо під час виробництва.
- Легкі обслуговування та ремонт системи.

Недоліками кнопових панелей контролю SIMATIC як електронно-механічних панелей є:

- Істотний вплив навколишнього середовища та людського фактора на зношення та вихід із ладу елементів системи.
- Відсутність надійних засобів недопущення та виправлення помилок дій оператора під час роботи із системою.
- Необхідність постійного перебування біля системи оператора.
- Прив'язка панелі контролю до однієї локації на підприємстві, що унеможливорює її мобільність.
- Відсутність збереження звітів щодо роботи системи.
- Відсутність виведення графічної інформації, що ускладнює роботу із системою, знижує її продуктивність та рівень входу для роботи з нею.
- Необхідність постійного коригування якості процесу оператором, що створює додаткові загрози на основі людського фактора.
- Надання системі тільки функцій попередження та зміни вхідних даних із дозволу користувача. Відсутність прямого залучення системи до контролю та її самонавчання.

2. Сенсорна панель оператора є засобом людино-машинного інтерфейсу, що дає змогу користувачеві ефективно обмінюватися даними із обладнанням. Простий та інтуїтивно зрозумілий графічний інтерфейс забезпечує максимально зручне управління промисловими агрегатами. На сучасних панелях Yokogawa практично немає фізичних елементів – кнопок, перемикачів, тумблерів, індикаторів тощо [11].

Взаємодія відбувається через сенсорний екран. На ньому ж відображається вся необхідна оператору інформація. Така технологія дає змогу не тільки швидко і зручно контролювати обладнання, але і позбавляє необхідності використовувати додаткові електричні сполуки, наявні в традиційних кнопових системах управління. Надійність таких з'єднань часто сумнівна, тому вони можуть відмовити в найвідповідальніший момент. За сенсорного керування ризик виникнення такої ситуації зводиться до мінімуму. Переваги YOKAGAWA:

- Завдяки високошвидкісному інтерфейсу USB можна швидко завантажувати і вивантажувати робочі програми. Для копіювання і резервування даних, а також перенесення їх з однієї панелі на іншу можна використовувати USB-флешку або SD-карту.
- Торгова марка Yokogawa пропонує великий вибір яскравих і барвистих дисплеїв з високою роздільною здатністю, аж до 65535 кольорів, що дають змогу відтворювати чітку і реалістичну картинку.
- Програму ПЛК можна обслуговувати й оновлювати за допомогою панелі. Це забезпечує оперативну і зручну роботу з ПЛК з меншою кількістю кабелів і з'єднань.
- Панелі можуть бути з'єднані з різними типами струменевих і матричних принтерів через порт USB.
- Надається багато можливостей для комунікації через різні інтерфейси зв'язку.

Недоліки YOKAGAWA: вища вартість пристрою порівняно з кноповими варіантами або з панелями, які не мають елементів управління; деякі типи сенсорних дисплеїв не можуть ефективно працювати в умовах негативних температур, що робить маловживаним пристрій, коли температура нижча від нуля; працюючи із системою, завжди необхідно стежити за чистотою дисплея для забезпечення високої надійності роботи; все ще не вирішена проблема мобільності системи, що призводить до часткових втрат якості та швидкості вирішення проблем чи запобігання їм; надання

системі тільки функцій попередження та зміни вхідних даних із дозволу користувача. Відсутність прямого залучення системи до контролю та її самонавчання.

3. Панелі оператора Harmony ST6 мають розмір від 4,3 до 15,6 дюйма і поєднують виняткову якість з хорошою графікою (16 млн кольорів), стильним дизайном завдяки металевій передній панелі корпусу.



Рис. 3. Сенсорна вебпанель Schneider Electric Harmony STW6

Програмування Harmony ST6 здійснюється за допомогою нового функціонального ПО EcoStruxure Operator Terminal Expert, що є частиною пропозиції EcoStruxure Machine і об'єднує ключові технології для підключення продуктів, локального управління, а також хмарні технології для надання аналітичних і цифрових послуг. Постійний розвиток функціоналу EcoStruxure Machine дає змогу пропонувати клієнтам більше інновацій та додаткових переваг протягом усього життєвого циклу обладнання.[12]. Переваги Schneider Electric Harmony STW6:

- Металева основа забезпечує корпусу жорсткість, що в поєднанні з якісною прокладкою ущільнювача по периметру корпусу забезпечує гарантовано високий ступінь захисту з боку лицьової панелі – IP65.
- Корпус Harmony ST6 має додаткові зачепи-фіксатори, які запобігають випаданню терміналу з монтажного вікна до моменту його повної фіксації. Комплектні фіксатори уможливають монтаж панелі оператора без використання додаткового інструменту.
- Всі моделі Harmony ST6 мають на корпусі QR-код, за допомогою якого можна отримати швидке посилання на документацію із обслуговування панелі оператора, використовуючи смартфон із виходом в інтернет.
- Всі моделі терміналів Harmony ST6 містять широкоформатні дисплеї з високою роздільною здатністю, забезпечуючи відображення 16 млн кольорів, покращену графіку з використанням шрифтів TrueType, градієнта, прозорості та згладжування.
- Моделі Harmony ST6 відповідають вимогам ІоТ на рівні підключення продуктів, забезпечуючи безпечний зв'язок, підключення і ефективність для необхідного програмного забезпечення і застосунків.
- Вбудований WEB-браузер з підтримкою HTML5 дає змогу використовувати можливості доступу до меню для налаштування різних серверів і вебтерміналів.

Недоліки Schneider Electric Harmony STW6: висока ціна придбання системи та її розгортання. Необхідність кваліфікованого персоналу для виправлення програмних помилок із функціонування системи; необхідність додаткових засобів зв'язку, для безпечного передавання даних; відсутність підтримки системи на популярних платформах (смартфон, планшет, персональний комп'ютер); прив'язаність програмного забезпечення системи до вебпанелі; надання системі функцій лише попередження та зміни вхідних даних з дозволу користувача. Відсутність прямого залучення системи до контролю та її самонавчання. Як зрозуміло із наведеного опису систем контролю якості, основною їхньою проблемою є відсутність самонавчання системи, що призводить до значних фінансових витрат на придбання сировини та додаткову оплату праці працівникам. Також серед основних недоліків можна виділити високу ціну, необхідність навичок роботи зі схожими системами та

прив'язку систем до одного пристрою чи місця розташування. Для забезпечення функціонування інформаційної системи контролю якості поліетилену на базі НМІ-інтерфейсу необхідно визначити основні атрибути та функції, вказані в табл. 1:

- Графічний інтерфейс – містить низку візуальних засобів для покращення навігації користувача в інформаційній системі, а також, крім технічних, виконує естетичні функції, забезпечуючи презентабельність системи та її подальше просування на аналогічних підприємствах.
- Обмін повідомленнями – реалізується з метою комунікації між працівниками підприємства для подальшого узгодження дій, надсилання звітів та створення нотаток.
- Виробничий режим із застосуванням контролю якості – функцію реалізовано з метою моніторингу процесу виробництва, перегляду вихідних даних та оцінювання роботи системи. У цьому режимі користувач системи вводить тільки необхідні початкові дані, передбачені технологією виробництва, а надалі виконує роль лише спостерігача. Усі операції із подальшого контролю якості, нівелювання похибок та аналізу критичних ситуацій виконує система, яка на основі штучного інтелекту опрацьовує вихідні параметри і на їх основі задає подальші вхідні.
- База даних для автентифікації виконує захисну роль у системі, обмежує доступ до системи незареєстрованим користувачам. Зберігає всю інформацію про користувача та надає весь необхідний функціонал, прив'язаний до його ID.
- Режим діагностики – реалізується з метою перевірки справності обладнання та давачів, які закріплені за ними.
- База даних для виведення звітності щодо сеансів роботи системи необхідна для аналізу технологічних процесів, ефективності роботи кожного користувача, виконання економічних розрахунків та на їх основі формування подальших прогнозів.

Таблиця 1

Функціональність програмної системи

Атрибут Функція	Статус	Пріоритет	Трудо- ємкість	Ризик	Стабіль- ність	Цільо- ва версія	Призначення	Прийняте рішення
1	2	3	4	5	6	7	8	9
Графічний інтерфейс	Увімкнена v 1.0	Важлива	Високий рівень	Низький	Середня	v 1.0	Графічне зображення функцій для навігації системою	Реалізува- ти негайно
Обмін повідомленнями	Пропонована v 1.0	Корисна	Середній рівень	Низький	Висока	v 1.0	Обмін повідомлен- нями між користувач- чами	Реалізува- ти негайно
Виробничий режим із застосуванням контролю якості	Увімкнена v 1.0	Важлива	Високий рівень	Високий	Середня	v 1.0	Контроль виробничого процесу та якості кінцевого продукту	Реалізува- ти негайно

Продовження табл. 2

1	2	3	4	5	6	7	8	9
База даних для проведення автентифікації	Увімкнена v 1.0	Критична	Високий рівень	Середній	Середня	v 1.0	БД для проведення реєстрації та автентифікації	Реалізувати негайно
Режим діагностики	Увімкнена v 1.0	Корисна	Високий рівень	Високий	Висока	v 2.0	Контроль справності користувачем роботи обладнання та давачів	Реалізувати негайно
База даних для виведення звітності щодо сеансів роботи системи	Увімкнена v 1.0	Корисна	Низький рівень	Низький	Висока	v 3.0	Виведення всієї наявної інформації щодо кожного із сеансів системи	Розглянути можливі варіанти реалізації

Проаналізувавши галузь, ми розглянули основні засади досліджень, з яких випливає, що ця галузь доволі затребувана, однак все ще потребує модернізації та покращень. Детально вивчивши роботи провідних науковців у галузі, отримали загальну картину цільової аудиторії, її потреб, а також порівняння того, що очікують користувачі від використання інформаційної системи, і того, що вона може надати для технічного розвитку галузі. Окрім цього, для поглибленого розуміння необхідності систем контролю якості продукції здійснено короткий аналіз створення та впровадження перших систем на підприємствах. Порівнявши найпопулярніші системи контролю якості продукції, ми визначили вимоги до функціоналу даних систем. Чітко простежується недосконалість кожної з проаналізованих систем, оскільки розширення функціоналу прямо підвищує вартість системи і не дає можливість користуватися її даними, а те, що штучний інтелект не застосовується для контролю якості у виробництві, робить систему складнішою та потребує задіяння більшої кількості як персоналу, так і сировини. З вищесказаного випливає висновок про необхідність створення інформаційної системи, яка поєднуватиме мобільність, низьку вартість та надійний функціонал, а всі складні хіміко-технологічні процеси та контроль якості вихідного продукту здійснюватиме система із елементами штучного інтелекту, щоб унеможливити помилки на основі людського фактора.

Формулювання мети та постановка задачі

Для проектування інформаційної системи контролю якості виробництва поліетилену однією з важливих вимог є надання рекомендацій, що ґрунтуються на принципах побудови системи підтримки прийняття рішень та методології системного аналізу [13]. Серед усіх принципів можна виділити такі:

- *Принцип остаточної (глобальної) мети* – передбачає визначення мети та характеризується абсолютним пріоритетом, тобто визначає наявність певної основної цілі для інформаційної системи – надання інструментів для вирішення проблеми контролю якості виробництва поліетилену.

- *Принцип єдності* – сумісний розгляд системи як сукупності компонентів (елементів, підсистем, системотворчих відношень). У цьому випадку є: підсистема, яка надає можливість внесення необхідної інформації у програму (нових вхідних даних); підсистема надання рекомендацій на основі введених даних; підсистема самостійного контролю вихідних даних; підсистема, що дає змогу вивести всі вихідні дані у попередньо створеній відповідній формі (система виведення).

- *Принцип зв'язності* – містить необхідні сформовані дані, що ґрунтуються на внесених та збережених даних у базі даних.
- *Принцип модульності* – передбачає реалізацію декомпозиції на складові (модулі) різного ступеня загальності та розглядає систему як певну сукупність модулів та зв'язків між ними. Проектовану систему можна розділити на такі модулі: головний модуль логічного виведення, модуль логічного контролю вихідних даних; модуль виведення сформованої системи рекомендацій, модуль внесення даних користувача системи.
- *Принцип функціональності* – сприяє забезпеченню загальної функціональності системи, а саме функціональності внесення необхідних даних користувачів, збереження їхніх звітів.
- *Принцип розвитку* – повинен закладатися під час побудови штучних систем як здатність до вдосконалення, розвитку системи за умови збереження якісних особливостей, тобто повинен полягати у внесенні певних нових знань упродовж функціонування системи.

Наведені вище принципи надають змогу відобразити характерні особливості під час побудови системи та врахувати усі взаємоважливі зв'язки між елементами. Наведемо опис сутності системи відповідно до мети, яка визначає певні призначення системи та конкретизує за допомогою застосування дерева цілей [14]. Згідно із загальноприйнятою методологією системного аналізу, основною метою є створення інформаційної системи контролю якості виробництва поліетилену. Можливість виконання цього завдання з'являється тільки тоді, як досягнуто всі наявні підцілі. Від вершини дерева цілей, що уособлює головну ціль, виходять три гілки, які ділять головну ціль на три відповідні підцілі.

Перша підціль – це “Перевірка кожного етапу виробництва”. На цьому етапі необхідно виділити характеристики підцілі та сформувані структуру знань, що потрібно для ефективної роботи системи. Виділена підціль також поділяється на дві нижчі підцілі, завданнями яких є проектування засобів для перевірки етапів виробничого процесу та створення логічної моделі бази знань із відомостями про стан роботи системи.

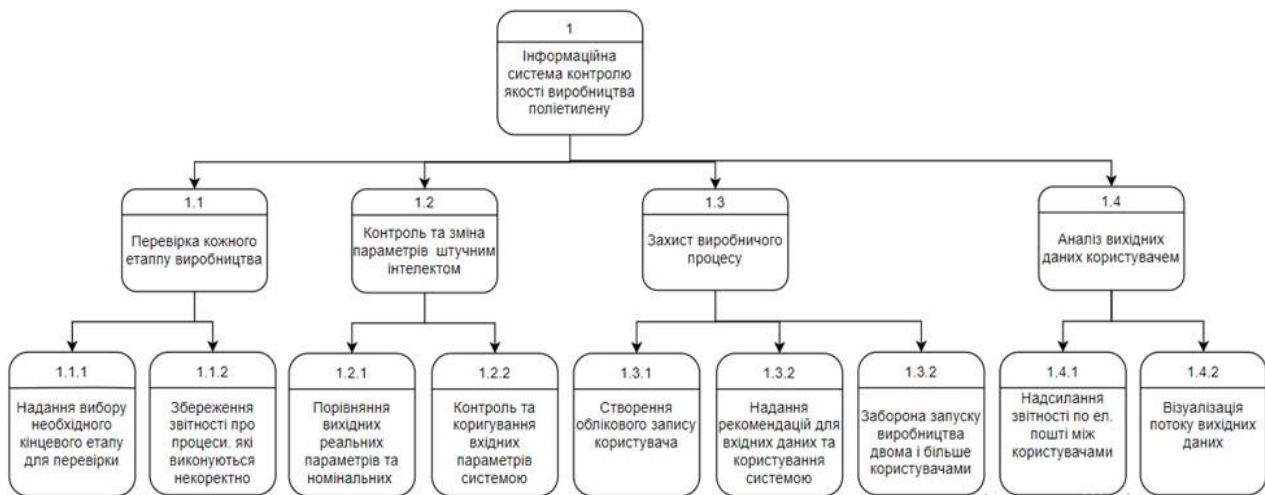


Рис. 4. Дерево цілей

Другою підціллю є “Контроль та зміна параметрів штучним інтелектом”. Метою цієї гілки є аналіз якості та її подальший контроль. Ця підціль також поділяється на дві нижчі підцілі, а саме: порівняння вихідних реальних параметрів та номінальних (відстежує зміни даних, виконує їх аналіз, а також на їх основі здійснює самонавчання, для визначення відхилення від номінальних параметрів), контроль та коригування вхідних параметрів системою (на основі проаналізованих даних приймає рішення про зміну вхідних параметрів, для підтримки необхідного параметра якості продукту). Третя підціль – це “Захист виробничого процесу”. Мета цієї підцілі – забезпечити безпечну роботу із інформаційною системою та зменшити вплив людського фактору на неї. Ця підціль поділяється на три нижчі підцілі: створення облікового запису користувача (слугує для прив’язки користувача і

виробничого процесу, аналізування його дій), надання рекомендацій для вхідних даних та користування системою (зменшує інтелектуальний поріг входу для користування системою), заборона запуску виробництва двома і більше користувачами (необхідне для унеможливлення виникнення нелогічних ситуацій на виробництві, які відповідно можуть призвести до критичних наслідків). Четвертою підциллю є “Аналіз вихідних даних користувачем”. Мета цієї підцилі полягає у наданні зручних засобів користувачам для обміну даними та подальшого їх аналізу. Ця підциль поділяється також на дві підцилі: надсилання звітності електронною поштою (надає змогу звітувати про роботу системи) та візуалізація потоку вихідних даних (надає інструменти для спрощеного виведення окремої інформації користувачу системи).

Виклад основного матеріалу

Для конкретизації функціонування інформаційної системи необхідно задіяти структурну методологію із реалізацією у вигляді ієрархії DFD чи еквівалентної IDEF0. Для кожної без винятку методології необхідне спеціальне середовище їх побудови. На основі аналізу засобів моделювання необхідних діаграм було вирішено використовувати середовище AllFusion Process Modeler, оскільки воно має зрозумілий інтерфейс для користування, всі необхідні функції для роботи із діаграмами і підтримує три основні методології моделювання: IDEF0, IDEF3 та DFD. Зі списку цих методологій ми використовували для моделювання конкретизації функціонування системи методологію IDEF0, тому що вона надає змогу досліджувати функції системи, не пов’язуючи їх з об’єктами, які забезпечують їх реалізацію. Саме цей підхід у реалізації діаграм, в нашому випадку, дає змогу найширше описати функціонування системи [15]. Спочатку було створено контекстну діаграму “Інформаційна система контролю якості поліетилену” в методології IDEF0 (рис. 5).



Рис. 5. Контекстна діаграма IDEF0

В управлінні увійшли:

- *Технологія виготовлення* – конкретні, чіткі вказівки користувачу інформаційної системи, які можуть видаватися на основі як загальноприйнятих стандартів із технічної документації, так і персональних стандартів, які визначає керівник.
- *Алгоритм програми* – це певна послідовність команд виконавцеві цього алгоритму (у нашому випадку штучному інтелекту системи), спрямованих на розв’язання певної задачі.

Механізмами здійснення процесу є:

- *Інтерфейс системи* – певна розроблена сукупність правил, методів і засобів взаємодії (контролю, управління тощо) між розробленими елементами системи, з дотриманням вимог до інтерфейсів систем контролю якості.

- *Давачі* – певні вимірювальні пристрої, до складу яких входять пристрої механічного контролю за виробництвом та ті, які виробляють вихідні сигнали, зручні для дистанційного зберігання, передавання та використання у системах керування чи контролю.

- *Система* – логічні команди та засоби штучного інтелекту, які можуть виконувати необхідні розрахунки та на основі даних, отриманих під час розрахунків чи від давачів, здійснювати аналіз, контроль та вносити відповідні зміни у виробничий процес.

Вхідними даними:

- *Дані користувача* – електронна пошта та пароль, які необхідні для входу в систему та подальшої роботи з нею.

- *Час роботи* – кількість часу, упродовж якого триватиме активна стадія виробничого процесу.

- *Дані фракції* – основна одиниця сировини, яка застосовується для виготовлення продукції.

- *Дані каталізатора* – друга за значущістю одиниця сировини, яку за технологією прийнято застосовувати у відношенні 1:1 із фракцією.

- *Дані розчинника* – дані, необхідні для проведення хімічних реакцій.

- *Дані пристроїв* – величини, які надходять із давачів або інших пристроїв.

- *Дані метанолу* – дані, необхідні для промивання сировини.

- *Дані добавки* – дані, необхідні для задавання кінцевої класифікації готового продукту відповідно до стандартів.

Результатом роботи системи будуть:

- *Звіт роботи* – сформований звіт про роботу системи на період виробничого процесу, із усіма необхідними даними про користувача, сеанс та кінцевий продукт.

- *Характеристики поліетилену* – основні величини, які важливі в кінцевому результаті для оцінювання продукції та її класифікації.

- *Якість поліетилену* – це визначена якість готового кінцевого продукту, яка ґрунтується на розрахованих та отриманих даних щодо характеристики кінцевої продукції, процесу виробництва і технології виготовлення.

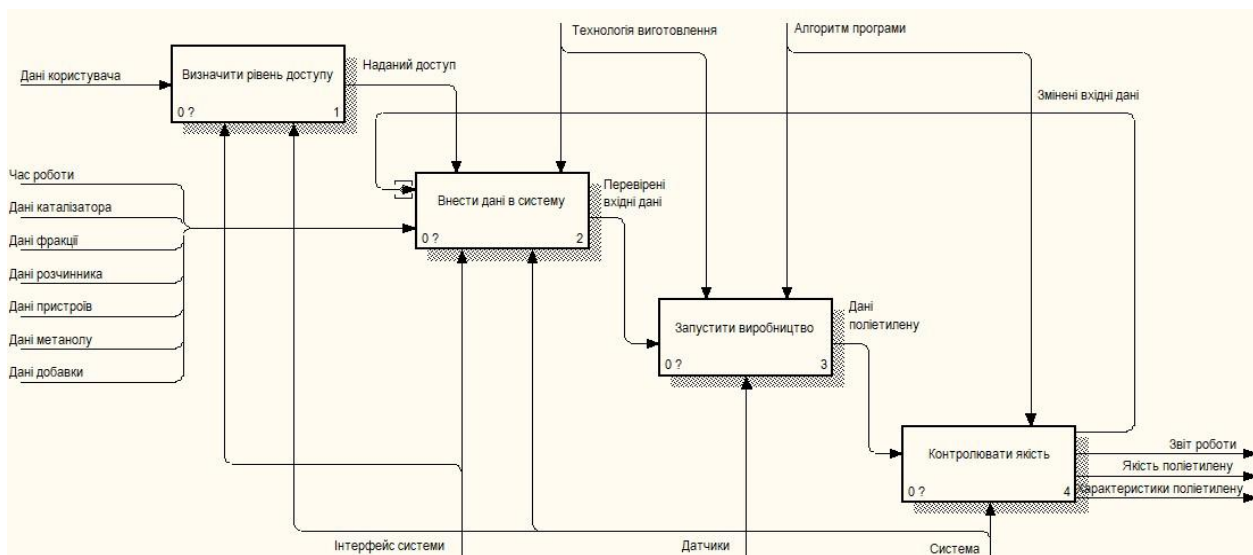


Рис. 6. Декомпозиція контекстної діаграми

Після створення та опису контекстної діаграми було здійснено функціональну декомпозицію, тобто поділ системи на підсистеми. Кожну підсистему опишемо окремо.

Контекстну діаграму ми поділили на чотири блоки (рис. 6):

- *Визначити рівень доступу* – блок, який відповідає за захист системи, надання доступу авторизованим користувачам та верифікацію для незареєстрованих користувачів.

- *Внести дані в систему* – призначений для налаштування виготовлення поліетилену відповідно до певної технології, яку задав керівник, та ґрунтується на введенні даних, необхідних як для тестування системи, так і для повноцінного запуску виробництва, з метою виготовлення готової продукції.
- *Запустити виробництво* – блок, під час якого відбувається весь виробничий процес. Під час виконання цього процесу всі етапи відбуваються лінійно, для того щоб не виникало ситуації невизначеності, яка може мати критичні наслідки як для системи, так і для матеріального забезпечення підприємства.
- *Контролювати якість* – блок, все управління і аналіз якого ґрунтуються на штучному інтелекті, який контролює якість кінцевого продукту, збираючи та аналізуючи уже готові дані, й на їх основі задаючи нові параметри, які містяться в межах, відповідно до попередньо заданої технології виробництва. Окрім цього також приймає рішення про екстрене завершення виробництва, якщо якість виходить за межі мінімальних допустимих показників [16].

Після подальшого поділу діаграми отримуємо ще чотири діаграми декомпозиції, що детально описують кожен із наявних блоків, зображених у декомпозиції контекстної діаграми. Декомпозиція блока “Визначити рівень доступу” (рис. 7) детально ілюструє послідовність виконання входу в систему. Після надходження даних користувача із інтерфейсу системи виконується запит до СД, для перевірки наявності користувача. Якщо користувач наявний у СД, система виконує додаткову перевірку наявності повноважень для роботи (визначаючи, чи систему запустив інший користувач), і після цього надається доступ до користування системою. Якщо ж даних користувача немає у СД, користувачу пропонують здійснити реєстрацію і подальшу верифікацію. Етапи, що відбуваються надалі, аналогічні до етапів для зареєстрованих користувачів.

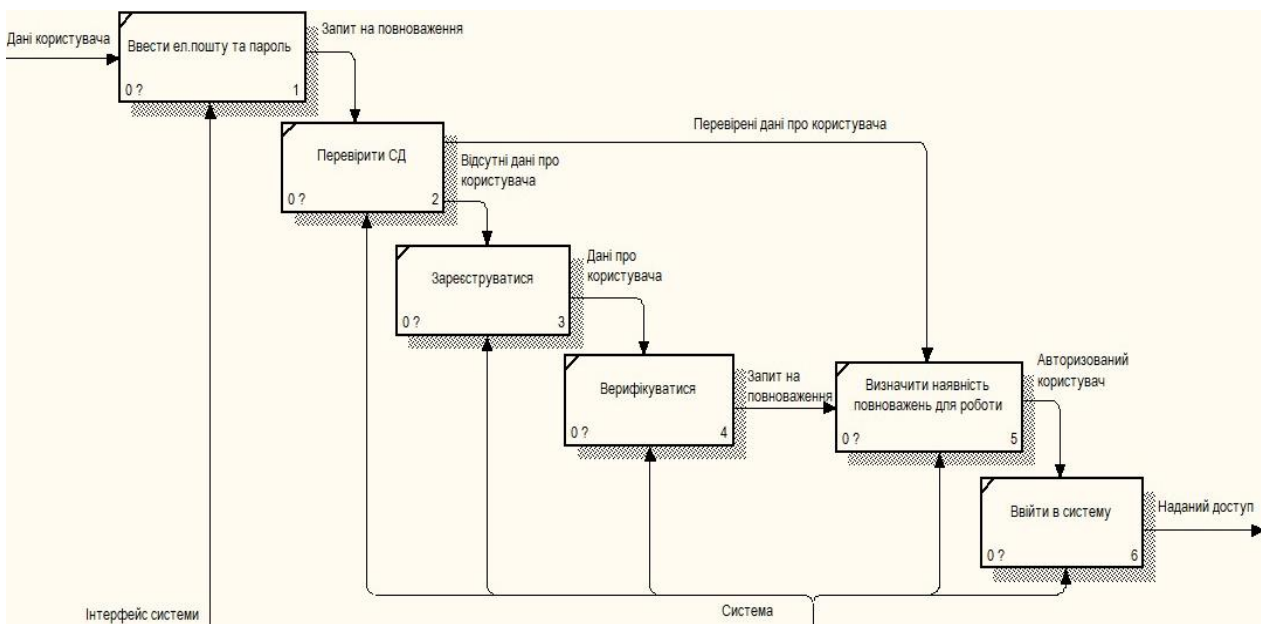


Рис. 7. Діаграма декомпозиції IDEF0 “Визначити рівень доступу”

Декомпозиція блока “Внести дані в систему” (рис. 8) деталізує процес задавання необхідних даних для виробничого процесу із виготовлення поліетилену або їх редагування. Якщо надано дозвіл працювати із системою, користувач отримує можливість вводити першочергові дані для виготовлення поліетилену або запуску перевірки справності виробничого процесу. Після внесення першочергових даних надається можливість вносити дані добавки, для запуску повноцінного виготовлення. У випадку введення як першочергових, так і другорядних даних продукції, яка буде брати участь у виробничому процесі, користувачу необхідно увімкнути пристрої, після чого всі опрацьовані дані, із кожного блока, надходять на перевірку. Відповідно до результату аналізу перевірки система розпочинає виробництво із перевіреними даними або надає вказівки щодо

додавання чи редагування вхідних даних. Всі операції, здійснені й описані в цій діаграмі декомпозиції, можуть здійснювати як користувач, так і система, за винятком блока “Увімкнути пристрої”, який реалізується тільки за допомогою користувача, тобто інтерфейсу системи [17].

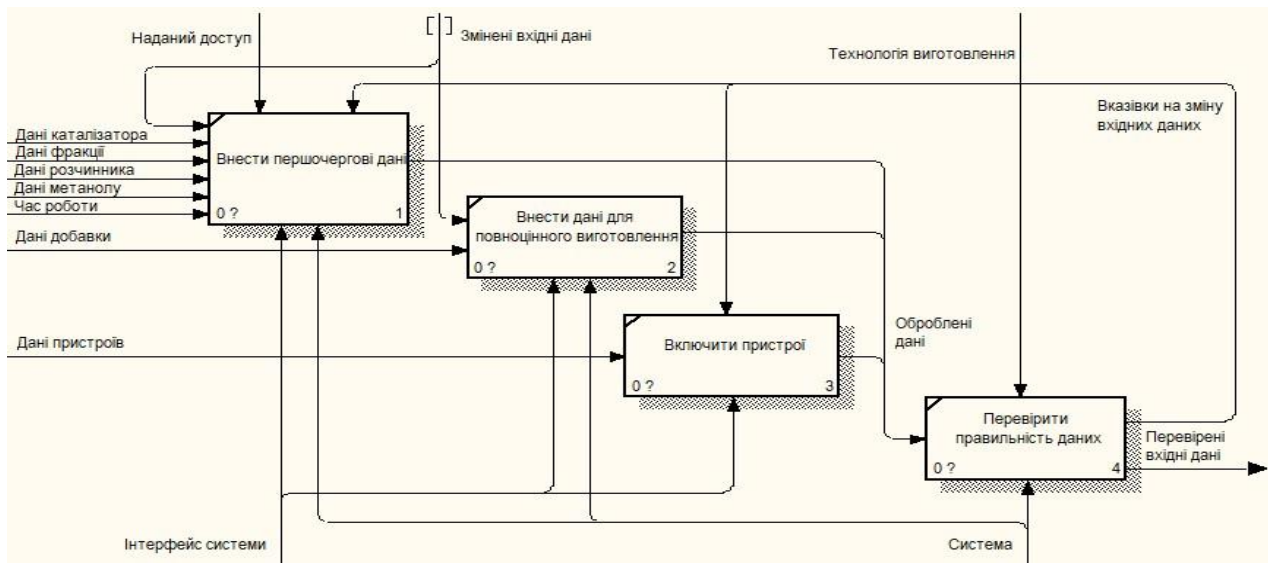


Рис. 8. Діаграма декомпозиції IDEF0 “Внести дані в систему”

Декомпозиція блока “Запустити виробництво” (рис. 9) ілюструє процес виробництва, який є строго лінійним і не передбачає ніяких можливостей розгалуження. Після запуску виробництва перевірені дані надходять до процесу полімеризації, а проаналізовані дані полімеру відповідно надходять до процесу відмивання від каталізатора. Після виконання цього процесу дані про відкритий ПО надходять до блока висушування, який своєю чергою надає обчислені дані поліетилену. Механізмом здійснення процесів у цьому блоці “Запустити виробництво” є тільки давачі [18].

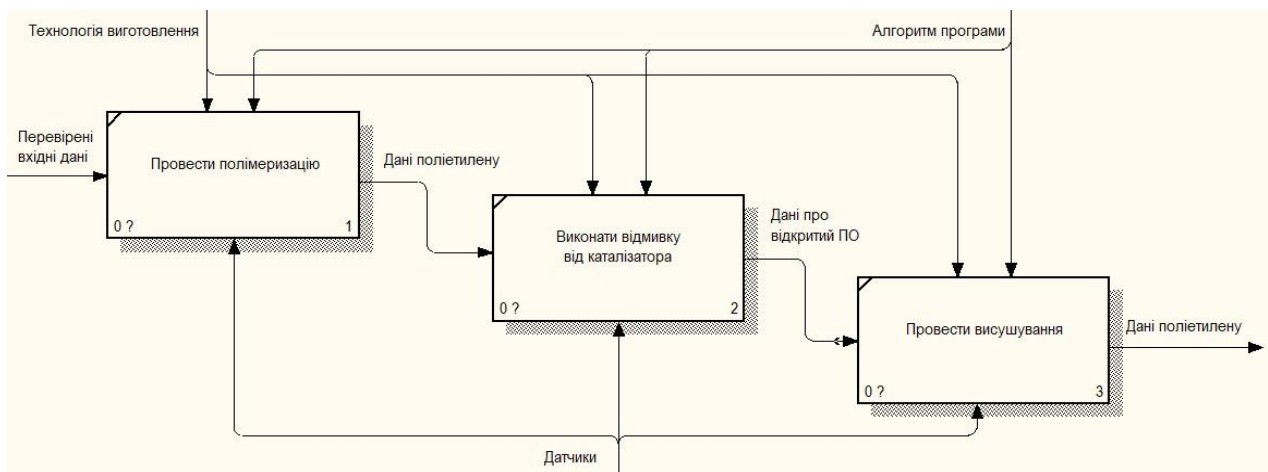


Рис. 9. Діаграма декомпозиції IDEF0 “Запустити виробництво”

Декомпозиція блока “Контролювати якість” (рис. 10) необхідна як для контролю якості, так і для контролю виробничого процесу загалом. Коли надходять дані щодо поліетилену, відбувається порівняння реальних та номінальних параметрів якості. Після перевірки даних, зокрема про якість поліетилену та його характеристики, інформація про поліетилен виводиться для користувача. Окрім цього, всі дані про поліетилен записуються у СД.

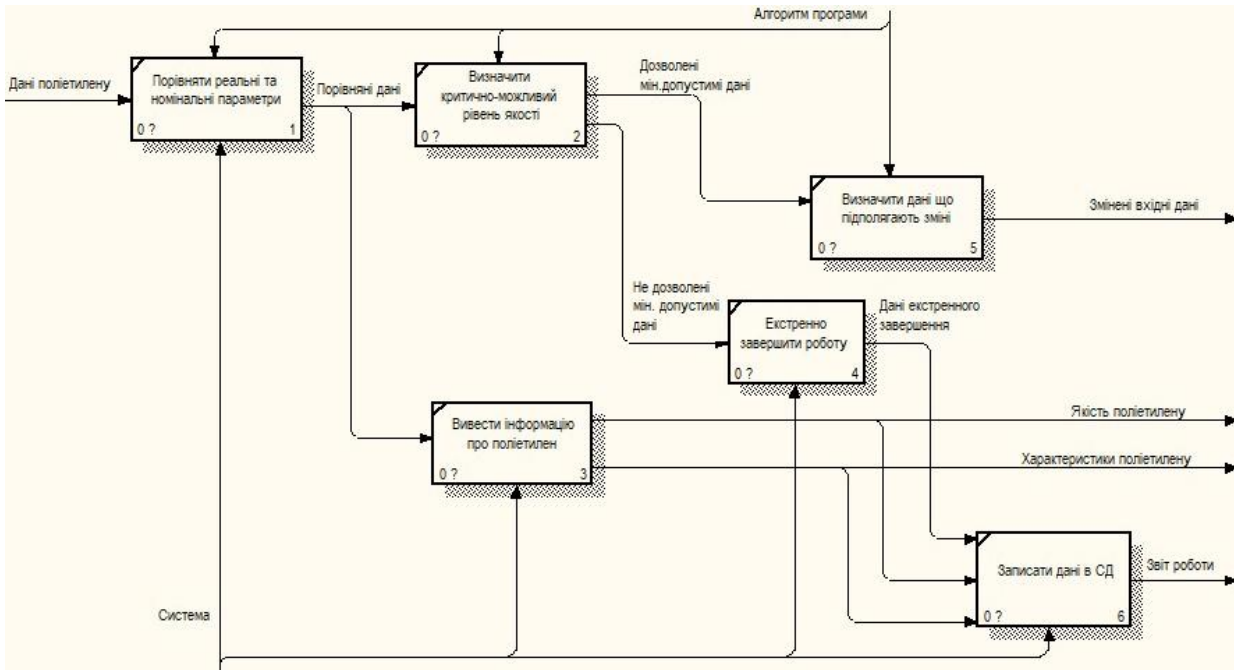


Рис. 10. Діаграма декомпозиції IDEF0 “Контролювати якість”

Паралельно із цим ті самі перевірені дані аналізує штучний інтелект і на їх основі визначає критично можливий рівень якості продукції. Якщо система робить висновок про те, що рівень якості нижчий від критичного, відбувається екстрене завершення роботи та подальший запис у СД причини завершення. Якщо ж рівень якості система оцінила як такий, що є дозволеним, штучний інтелект визначає дані, що підлягають зміні для підтримки чи поліпшення якості кінцевої продукції. Після цього змінені вхідні дані переадресовуються у блок “Внести дані в систему”. Всі розрахунки, аналіз та прийняття рішень у блоці “Контролювати якість”, а також зміну даних у блоці “Внести дані в систему” здійснює штучний інтелект системи. Для того, щоб не заплутатися і не загубитися у великій кількості інформації, а також перевірити коректність декомпозиції, необхідно після кожного внесення змін створювати діаграму ієрархії процесів (дерева вузлів). Подання інформаційної системи у вигляді функціональної схеми – поширений спосіб моделювання функціональності системи. Діаграма зображає усю ієрархію робіт у моделі та надає змогу розглянути абсолютно всю модель, однак не може показувати взаємозв’язки між наявними роботами (стрілки). Цей процес створення моделі робіт є ітеративним, отже, місце розташування роботи може змінюватись на створеному дереві вузлів багаторазово.

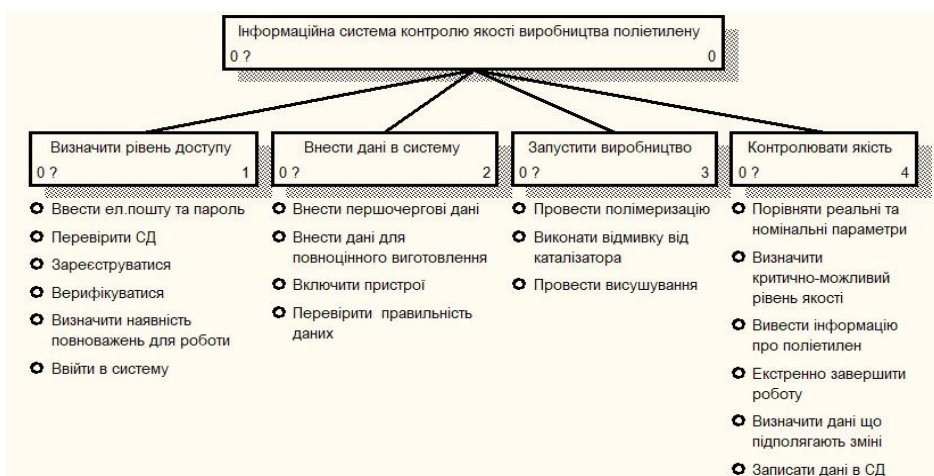


Рис. 11. Діаграма ієрархії процесів

У цій діаграмі, що зображає ієрархію процесів, глибина дерева дорівнює трьом, а корінь дерева – “Інформаційна система контролю якості поліетилену”, яка, своєю чергою, є батьківською роботою діаграми. Нижній рівень декомпозиції показано за допомогою списку, однак решту робіт зображено у вигляді прямокутників. Для повного відображення ієрархії за допомогою прямокутників необхідно вимкнути опцію *Bullet Last Level*. Для побудови інформаційної системи контролю якості виробництва поліетилену доводиться задіювати кілька галузей для максимально ефективної роботи системи. Умовно систему можна розділити на: частину, яка реалізовує дизайн, функціональну частину та частину зі сховищем даних. Для цього надається низка доступних рішень. Перше та найпримітивніше рішення – реалізувати систему на зв’язці HTML+CSS+JS із частиною бекенду на Python чи PHP та підключеною до нього базою даних. Однак це рішення надзвичайно складне, оскільки потребує великих затрат часу, додаткового підключення до локальної бібліотеки хостингу, навичок із оптимізації різних системних підходів під цей код, а також є дещо застарілим у плані швидкодії системи, її зовнішнього вигляду та функціональності. Наступне із доступних рішень – побудова системи на основі CMS. Система керування вмістом, або Content Management System, є програмним забезпеченням, яке використовують для надання додаткового функціоналу, а також організації та структуризації вебсайтів, як у відокремлених комп’ютерних мережах, так і загалом в інтернеті [19]. Існують сотні, а може, навіть й тисячі доступних CMS-систем. Завдяки розширеній функціональності ці системи можна використовувати в різних галузях людської діяльності. Основні функції сучасних CMS такі [20–23]:

- Генерація необхідної вебсторінки за запитом користувача.
- Керування як текстовими, так і графічними блоками контенту на сайті, із наданням користувачеві зручного інтерфейсу для роботи.
- Забезпечення зручними інструментами оприлюднення інформації на сайті, а також подальше її зберігання.
- Автоматизація усіх процесів із розташування інформації в наявних базах даних та її подальше оприлюднення у коді HTML.
- Попередній показ вебсторінки для редагування одному із адміністраторів.
- Забезпечення інструментарієм для видалення чи редагування контенту.

Здійснимо загальне порівняння CMS систем за основними параметрами, які потрібні для ефективного створення системи, а саме: ефективність написання FrontEnd-у, BackEnd-у, трудомісткість, підтримка із боку розробників, ефективність підключення додаткових засобів розроблення, доступність та оптимізація [20–23].

Таблиця 2

Порівняння основних характеристик CMS систем

	WordPress	Joomla!	Drupal	UMI.CMS	NetCad
FrontEnd	Дуже низька	Низька	Низька	Низька	Низька
BackEnd	Дуже низька	Низька	Дуже низька	Дуже низька	Низька
Трудомісткість	Дуже низька	Дуже низька	Низька	Середня	Середня
Додатки	Відсутні	Відсутні	Відсутні	Відсутні	Відсутні
Підтримка	Висока	Середня	Середня	Низька	Низька
Доступність	Висока	Середня	Середня	Дуже низька	Низька
Оптимізація	Дуже висока	Висока	Середня	Середня	Середня

Детально проаналізувавши ефективність застосування CMS систем і порівнявши їх із можливістю реалізувати систему на зв'язці HTML+CSS+JS із частиною бекенду на Python чи PHP, ми вирішили, що жоден із варіантів не підходить, оскільки хоч CMS і економлять час на побудову системи, однак не надають достатньої свободи дій для побудови системи з нуля. Окрім цього, потрібен додатковий час для їх освоєння, постійного пошуку шаблонів, які будуть максимально відповідати системі, що планується. Варто також зазначити, що всі CMS системи побудовані на мові PHP, яка стрімко втрачає актуальність на ринку розроблення інформаційних систем, зважаючи на її погану оптимізацію, низьку швидкодію та малу функціональність. Хоча CMS системи можуть ефективно використовувати користувачі, в яких немає навичок розроблення, однак вони не мають необхідної гнучкості для розробки, можуть потребувати додаткових грошових затрат та не ефективні для створення комбінованих систем [24]. Альтернативою цим рішенням є можливість використання для розроблення фреймворків. Вони містять елементи, які дають змогу значно швидше та ефективніше розробляти сайти, а також набагато гнучкіші для роботи із системою.

Здійснимо загальне порівняння фреймворків за основними параметрами [25–29], які потрібні для ефективного створення системи, а саме: ефективність написання FrontEnd-у, BackEnd-у, трудомісткість, підтримка розробниками, ефективність підключення додаткових засобів розроблення, доступність, оптимізація, взаємодія з Node. Виконавши детальний аналіз фреймворків та здійснивши їх порівняння із попередніми CMS засобами побудови інформаційних систем, ми виявили їх високу ефективність та відносну простоту в застосуванні. Для подальшої роботи із фронтендом системи вибрано фреймворк ReactJS, а для основної роботи, пов'язаної з бекендом, – ExpressJS із підключеним Material UI пакетом даних та додатковими засобами ExpressRouting. Ці додаткові засоби допоможуть реалізувати сучасний інтерфейс, відповідно до сучасних вимог дизайну, а також побудувати ефективний роутинг для простішої навігації у системі.

Таблиця 3

Порівняння основних характеристик фреймворків

	ExpressJS	ReactJS	Ruby	Angular
FrontEnd	Низька	Дуже висока	Середня	Середня
BackEnd	Дуже висока	Середня	Середня	Середня
Трудомісткість	Середня	Низька	Середня	Висока
Взаємодія з Node	Висока	Висока	Низька	Низька
Додатки	Наявні	Наявні	Відсутні	Відсутні
Підтримка	Дуже висока	Дуже висока	Висока	Середня
Доступність	Висока	Висока	Середня	Висока
Оптимізація	Висока	Висока	Дуже низька	Низька

Основні зв'язуючі програмні засоби розроблення інформаційної системи – ReactJS і ExpressJS. Оскільки головна суть системи полягає у контролі якості на основі засобів штучного інтелекту, головну увагу варто приділити ExpressJS як засобу роботи із бекендом. ExpressJS – найпопулярніший серед вебфреймворк для Node. вебфреймворків Node є базовою бібліотекою для інших популярних і використовується як основний засіб для роботи із бекендом системи. Головними мовами програмування, що застосовуються для роботи в цьому фреймворку, є JavaScript і TypeScript [30]. Хоч сам Express доволі мінімалістичний, розробники створили сумісні пакети проміжного програмного забезпечення для вирішення практично будь-якої проблеми з веброботкою. Існують бібліотеки для роботи з cookie-файлами, сеансами, входами користувачів, параметрами URL, даними POST, заголовками безпеки і багатьма іншими. Express не є обмежувальним фреймворком, завдяки чому надаються широкі можливості вставляти в ланцюжок опрацювання (middleware) запитів

практично будь-які сумісні проміжні компоненти. Також можна структурувати додаток в одному файлі або в декількох, використовуючи будь-яку структуру каталогів.

Для того, щоб детальніше оглянути всі технічні характеристики фреймворку, виконаємо його детальний аналіз. У традиційних динамічних вебсайтах вебдодаток очікує HTTP-запит від веббраузера (або іншого клієнта). Коли запит отримано, додаток визначає, яку дію необхідно виконати на основі URL шаблону і, можливо, пов'язаної інформації, що міститься в даних POST або GET. Залежно від того, що потрібно, Express може потім читати чи записувати дані з бази даних, а також виконувати інші завдання, відповідно до отриманого запиту. Потім додаток повертає відповідь у веббраузер, найчастіше динамічно створюючи HTML сторінку для відображення браузером, вставляючи витягнуті дані в наповнювачі HTML шаблону. Express надає методи, що дають змогу вказати, яка функція викликається для конкретного HTTP запиту (GET, POST, SET, etc.), і URL шаблон ("Route"), а також методи, що дають можливість вказати, який механізм шаблону ("view") використовується, де містяться шаблони файлів і який шаблон використовувати для виведення відповіді. Можна використовувати Express middleware для додавання підтримки файлів cookies, сеансів, і користувачів, отримання POST/GET параметрів тощо. Окрім цього, надається змога використовувати будь-який механізм бази даних, підтримуваний Node (Express не визначає поведінку, пов'язану із базою даних). У наступних підпунктах пояснимо деякі загальні моменти, які детальніше описують роботу з кодом у Express і Node.

Модуль – це бібліотека або файл JavaScript, який можна імпортувати в інший код за допомогою функції require() Node. Express сам по собі є модулем, як і проміжне програмне забезпечення та бібліотеки баз даних, використовувані в додатках Express. Наведений нижче код показує, як здійснюють базовий імпорт модуля за іменем, використовуючи як приклад платформу Express. Спочатку викликають функцію require(), визначаючи ім'я модуля. У вигляді рядка ("express") викликається повернутий об'єкт для створення додатка Express. Після цього надається можливість отримати доступ до властивостей і функцій об'єкта додатка.

```
var express = require('express'); var app = express();
```

Окрім цього, також можливо створювати власні модулі, які можна імпортувати так само. Щоб зробити об'єкти доступними поза модулем, просто потрібно призначити їх об'єкту експорту. В фреймворку Express JS код JavaScript часто використовує асинхронні, а не синхронні API для операцій, виконання яких може бути доволі тривалим. Синхронний API – це той, в якому кожна із операцій має завершитися до початку наступної. Наприклад, такі функції журналу є синхронними і виводять текст на консоль у послідовності (перший, другий).

```
console.log('First'); console.log('Second');
```

На відміну від цього, асинхронний API – це той, в якому API почне операцію і відразу ж повернеться (до завершення операції). Після завершення операції API буде використовувати деякий механізм для виконання додаткових операцій.

Використання неблокуючих асинхронних API-інтерфейсів ще важливіше в Node, ніж в браузері, оскільки Node – це однопотоківі виконання, керовані подіями. "Однопотоківі" означає, що всі запити до сервера виконуються в одному потоці (а не породжуються в окремих процесах). Ця модель надзвичайно ефективна з погляду швидкості й ресурсів сервера, але це означає, що якщо будь-яка з функцій викликає синхронні методи, виконання яких займає багато часу, вони блокуватимуть не тільки поточний запит, але і будь-який інший запит, що опрацьовується вебзастосунком.

Є кілька способів, якими асинхронний API повідомляє застосунок про те, що опрацювання завершено. Найпоширеніший спосіб – зареєструвати callback-функцію під час виклику асинхронного API, який буде викликатися після завершення операції.

Об'єкт застосунок Express також надає методи для визначення оброблювачів маршрутів для всіх інших HTTP-дієслів, які зазвичай використовуються однаково: post(), put(), delete

`()`, `options ()`, `trace ()`, `copy ()`, `lock ()`, `mkcol ()`, `move ()`, `purge ()`, `propfind ()`, `proppatch ()`, `unlock ()`, `report ()`, `mkactivity ()`, `checkout ()`, `merge ()`, `m-search ()`, `notify ()`, `subscribe ()`, `unsubscribe ()`, `patch ()`, `search ()` і `connect ()`.

Маршрути дають змогу зіставляти певні шаблони символів у URL-адресі, видобувати деякі значення із URL-адреси і передавати їх як параметри оброблювачу маршруту (як атрибути об'єкта запиту, переданого як параметр). Для прикладу, використання функції маршрутизації:

```
var wiki = require('./wiki.js'); app.use('/wiki', wiki);
```

Щоб використовувати маршрутизатор, надається модуль `route (wiki.js)`, із викликом `use ()` в застосунку Express, щоб додати маршрутизатор у шлях опрацювання проміжного програмного забезпечення.

Проміжне програмне забезпечення широко використовують у застосунках Express для завдань – від обслуговування статичних файлів до опрацювання помилок і стиснення HTTP-відповідей. Оскільки функції маршруту закінчують цикл запиту-відповіді HTTP, повертаючи деяку відповідь клієнту HTTP, функції проміжного програмного забезпечення зазвичай виконують деяку операцію над запитом або відповіддю і потім викликають наступну функцію у стеку”, яка може бути великою кількістю проміжного програмного забезпечення або маршруту обробника. Послідовність виклику проміжного програмного забезпечення залежить від розробника програми.

Більшість додатків використовують стороннє проміжне програмне забезпечення для спрощення загальних завдань веброзробки, таких як робота з файлами cookie, сесіями, автентифікацією користувача, доступом до даних запитів POST і JSON, ведення журналу тощо.

Для використання стороннього проміжного програмного забезпечення спочатку необхідно встановити його в свій застосунок за допомогою NPM. Після цього надається можливість викликати `use ()` для об'єкта застосунку Express, щоб додати проміжне програмне забезпечення у стек:

```
var express = require('express'); var logger = require('morgan'); var app = express(); app.use(logger('dev'));
```

Окрім всього вищесказаного, можна додати функції проміжного програмного забезпечення в ланцюжок опрацювання за допомогою `app.use ()` або `app.add ()`, залежно від того, чи є необхідність застосувати проміжне програмне забезпечення до всіх відповідей або до відповідей з певним дієсловом HTTP (GET, POST тощо). Маршрути задають однаково в обох випадках, хоча маршрут необов'язковий у разі виклику `app.use ()`.

Express надає користувачу змогу використовувати проміжне програмне забезпечення `express.static` для обслуговування статичних файлів, зокрема зображення, CSS і JavaScript (`static` – єдина функція проміжного програмного забезпечення, яка фактично є частиною Express).

Наприклад, постає необхідність використовувати рядок нижче для обслуговування зображень, файлів CSS і файлів JavaScript із каталогу з ім'ям `public` на тому ж рівні, де викликається вузол:

```
app.use(express.static('public'));
```

Можна викликати `static ()` кілька разів для обслуговування декількох каталогів. Якщо файл неможливо знайти однією функцією проміжного програмного забезпечення, він буде просто переданий подальшому проміжному програмному забезпеченню:

```
app.use(express.static('public')); app.use(express.static('media'));
```

Також можна створити віртуальний префікс для статичних URL-адрес, замість додавання файлів до базової URL-адреси. Наприклад, вказуємо шлях монтування, щоб файли завантажувалися з префіксом `"/media"`:

```
app.use('/media', express.static('public'));
```

Після цього надається можливість завантажувати файли, що містяться в публічному каталозі, з префіксу/media path. Помилки опрацьовуються однією або декількома спеціальними функціями проміжного програмного забезпечення, які мають чотири аргументи замість звичайних трьох: (err, req, res, next), наприклад:

```
app.use(function(err, req, res, next) { console.error(err.stack);
  res.status(500).send('Something broke!'); });
```

Вони можуть повертати будь-який необхідний контент, але повинні викликатися після всіх інших app.use () і маршрутизувати виклики, щоб вони були як останній проміжний ПО в процесі опрацювання запитів. Express поставляється із вбудованим оброблювачем. Ця проміжна функція опрацювання помилок за замовчуванням додається у кінець стека функцій проміжного програмного забезпечення. Якщо користувач передає помилку в next () і не опрацьовує її в обробнику помилок, її опрацює вбудований оброблювач помилок. Надалі помилка буде записана клієнту з трасуванням стека. Фреймворк Express може використовувати будь-який механізм бази даних, підтримуваний Node. Є багато варіантів, зокрема PostgreSQL, MySQL, Redis, SQLite, MongoDB тощо. Щоб використовувати їх, користувач повинен спочатку встановити драйвер бази даних, застосувавши NPM. Наприклад, щоб встановити драйвер для популярної NoSQL MongoDB, він повинен використовувати команду: \$ npm install mongodb. Сама база даних може бути встановлена локально або на хмарному сервері. Іншим популярним підходом є непрямий доступ до бази даних за допомогою Object Relational Mapper (“ORM”). За такого підходу свої дані визначають як “об’єкти” або “моделі”, і ORM відображає їх у базовий формат бази даних. Перевага цього підходу в тому, що і розробник, і користувач може продовжувати думати із погляду об’єктів JavaScript, а не семантики бази даних, і що це очевидне місце для виконання перевірки вхідних даних. Для цієї системи вибрали базу даних PostgreSQL, зважаючи на функції, які дають змогу виконувати деякий код безпосередньо сервером бази даних. Ці функції можуть бути написані на SQL, який має деякі примітивні програмні оператори, такі як розгалуження та цикли. Але гнучкішою буде функція, написана на одній із мов програмування, з якими PostgreSQL може працювати [31]. До мов програмування у PostgreSQL належать:

- Вбудована мова, подібна до процедурної мови PL/SQL компанії Oracle.
- Мови розроблення сценаріїв: PL/Perl, PL/Python, PL/Tcl, PL/Ruby, PL/sh.
- Класичні мови програмування C, C++, Java (за допомогою PL/Java).

Функції можуть виконуватись із привілеями користувача, який їх викликав, або із привілеями користувача, який їх написав.

Механізми шаблонів у Express дають змогу вказувати структуру вихідного документа в шаблоні, використовуючи наповнювачі для даних, які будуть заповнюватися під час створення сторінки. Шаблони часто використовують для створення HTML, але можуть також створювати інші типи документів. В Express є підтримка певних шаблонних движків, і тут корисне порівняння популярніших движків: Jade, Mustache, Dust і More. У коді налаштувань програми надається можливість задати механізми шаблонів для використання і місце, де Express повинен шукати шаблони, використовуючи налаштування “views” і “engine”, як показано нижче:

```
var express = require('express'); var app = express();
app.set('views', path.join(__dirname, 'views')); app.set('view engine',
'some_template_engine');
```

Зовнішній вигляд шаблону залежатиме від того, який движок використано. Припустимо, що є файл шаблону з ім’ям “index”.

<Template_extension> », який містить наповнювачі для змінних даних з іменами “title” і “message”, користувач повинен викликати Response.render () у функції обробника маршруту для створення і надсилання відповіді HTML:

```
app.get('/', function(req, res) { res.render('index', {title: 'About dogs',
  message: 'Dogs rock!' }); });
```

Express не робить ніяких припущень щодо структури або компонентів, які використовує користувач. Маршрути, уявлення, статичні дані або інша логіка конкретного застосунку можуть перебувати в будь-якій кількості файлів з будь-якою структурою каталогів. Хоча цілком можливо мати всі програми Express в одному файлі, зазвичай доцільно розділити застосунок з файлами на основі функції (наприклад, управління обліковими записами, блоги, дошки обговорень) і проблемної області архітектури (наприклад, модель, уявлення або контролер). Цей аспект файлової структури надає додаткову гнучкість розробнику та підвищує ефективність його роботи у декілька разів. Зазвичай для початківців наявна можливість використання Express Application Generator, що створює модульний каркас застосунку, який можна легко розширити для створення вебзастосунків.

Розроблена інформаційна система контролю якості поліетилену доволі складна за структурою та наповненням. Для того, щоб зрозуміти принцип її функціонування, необхідно виконати детальний аналіз логічної структури, функціональних можливостей, а також комунікації елементів системи, зв'язаних між собою. Окрім цього, з огляду на те, що система має потенціал для розширення і розвитку, постає необхідність опису створеного програмного засобу, для подальшого подання інформації новим розробникам, службі технічної підтримки, тестувальникам чи іншим зацікавленим особам. Функціональне призначення інформаційної системи полягає у контролі штучним інтелектом якості продукції в реальному часі, із виявленням дефектної продукції на ранніх етапах, та встановлення відповідності показників продукції і процесів вимогам нормативно-технічної документації, зразкам, еталонам, і запобігання цим випуску неякісної продукції. Окрім цього, користувачам інформаційної системи необхідно надати повну інформацію про перебіг виробничого процесу та підтримання його стабільності, мінімізуючи необхідність втручання користувача у виробничий процес, завдяки чому буде зменшено вплив помилок, спричинених людським фактором, і збільшено ефективність інформаційної системи. Необхідно забезпечити можливості комунікації в середині інформаційної системи про перебіг виробничого процесу. Важливим є повноваження штучного інтелекту і користувача. Хоча користувач обмежений в своїх діях і його вплив на інформаційну систему стає мінімальним після виставлення параметрів для продукції, однак у нього залишається можливість призупинити виробничий процес або достроково завершити його. Логічна структура кожної інформаційної системи може бути подана на основі чотирьох основних моделей: лінійної моделі, моделі “ґрати”, деревоподібної моделі та “павутини”. Варто зазначити, що існують додаткові комбінації на базі основних моделей, які надають змогу реалізувати будь-яку логічну структуру сайту. Оскільки, як зазначено вище, інформаційна система доволі складна, а розроблення такого ресурсу – клопітка робота, наявність оптимальної структури зведе до мінімуму можливість технічних помилок. Це стосується як помилок у коді, так і наявності або відсутності окремих сторінок і матеріалів на них. З огляду на ці фактори, вибрано логічну деревоподібну структуру. Логічна деревоподібна структура – це модель організації системи, що є найпопулярнішою. Така структура дає користувачам системи змогу за бажанням керувати глибиною відвідування системи. Користувачі мають змогу входити тільки на сторінки найвищих рівнів або ж “спускатись” до нижчих рівнів. Наявні можливості вибору залежать від “ширини дерева”. Якщо користувачам для досягнення поставленої кінцевої цілі необхідно виконати занадто багато клацань миші, то структура ієрархії системи може виявитися у цьому випадку занадто вузькою. Користувачів дратуватиме нескінченне “клацання”, оскільки їхні дії не даватимуть очікуваних результатів. Водночас дуже широке “дерево”, яке ґрунтується на доволі великій кількості варіантів вибору, змусить користувачів системи витратити багато часу на вивчення пропонованих варіантів. Це також не дасть позитивних результатів. Беручи до уваги ці фактори логічної структури системи, ми прагнули до забезпечення оптимального рівня глибини і ширини. Графічне зображення логічної деревоподібної структури системи подано на рис. 12.



Рис. 12. Логічна структура системи

Також варто зазначити, що важливу роль у швидкодії та функціонуванні системи відіграє файлова структура, яка у цьому випадку також деревоподібна. Основою файлової структури є головна папка із назвою системи, в якій містяться такі основні директорії і файли:

- `./app` – директорія, що містить файли, які є основою фронтенду системи.
- `./dist` – директорія із транспільованим вихідним кодом бекенду (із TS в JS).
- `./node_modules` – містить бібліотеки платформи NodeJS.
- `./public` – скомпільована збірка фронтенду (build), відкрита для публічного доступу користувачам, а також статичні файли системи.

- `./src` – директорія, яка містить вихідний код бекенду.
- `./typings` – директорія, в якій міститься опис об'єктів та типів всієї аплікації.
- `./env` – файл, що містить ключові параметри запуску аплікації.
- `./knh.ts` – файл, який є вхідною точкою для запуску бекенду.
- `./package.json` – файл, що містить опис необхідних для аплікації бібліотек.
- `./tsconfig.json` – файл конфігурації налаштувань компілятора TypeScript.
- `./webpack.config.js` – файл конфігурації налаштувань бандлера Webpack.

Важливим пунктом для розуміння інформаційної системи є особливість програмних засобів розв'язання задач, які описані в третьому розділі. Завдяки можливостям, які вони надають, вибрано додаткові засоби, які дають змогу реалізувати певні можливості із розроблення інформаційної системи, збільшуючи її надійність. Список усіх підключених засобів розробки системи міститься у файлі `package.json`. Щоб повніше відобразити можливості системи, опишемо всі особливості системи із підключеними засобами:

- FrontEnd і BackEnd реалізовані мовою програмування TypeScript, яка є суперпозицією мови JavaScript.
- BackEnd створений на основі фреймворку ExpressJS. Детальний опис цього фреймворку наведено в третьому розділі.
- Сервер системи працює на платформі NodeJS та має доступ до всіх даних підприємства, для реалізації їх контролю, з метою виготовлення продукції високої якості.
- Інтерфейс користувача розроблений на основі вебаплікації, тобто є програмним забезпеченням, яке працює у браузері й завдяки якому користувачі системи не прив'язані до робочого місця, а можуть відкривати систему із будь-якого переносного пристрою (телефону, планшета, ноутбука).
- Web-аплікація – SPA (Single Page Application) застосунок, написаний на фреймворку ReactJS. SPA є вебзастосунком чи вебсайтом, який вміщується на одній сторінці, щоб забезпечити користувачу досвід, близький до користування настільною програмою.
- Під час реалізації системи використано бандл, тобто програму, яка комбінує сукупність програмних файлів і даних в один та міститься у файлі `bundle.js`.

- Для швидшої та простішої веброзробки поверх фреймворку ReactJS було використано бібліотеку Material-UI, що містить низку графічних та динамічних елементів, які роблять якіснішою візуалізацію системи та збільшують її презентабельність.
- Оскільки різні браузері підтримують різний набір інструкцій мови JavaScript, було вирішено використовувати транспайлер Babel. Загалом транспайлер – це програма, що дає змогу змінювати вихідний код однієї програми на еквівалентний вихідний код іншою мовою. У випадку з Babel він переписує сучасний Javascript на старий, для відкриття системи на старих версіях браузерів.
- Автентифікація запрограмована стандартом JWT. JSON Web Token – це стандарт токена доступу на основі JSON, стандартизованого в RFC 7519. Використовується для верифікації, шифрування і передавання даних авторизованих користувачів між постачальником ідентифікації та постачальником послуг.
- Базою даних слугує реляційна БД PostgreSQL. Сервер PostgreSQL написаний мовою С. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом.
- Для зручного користування базою даних використано ORM (Objectrelational mapping) технологію. Ця технологія зв'язує бази даних із концепціями об'єктно-орієнтованих мов програмування, надаючи “віртуальну об'єктну базу даних” та створюючи запити до бази об'єктно-орієнтованими мовами програмування.

Для того, щоб система почала функціонувати в мережі інтернет для користувачів у звичайному режимі, адміністратору системи необхідно виконати певні важливі дії із завантаженням системи і її розгортанням. Завершивши розроблення системи та перевірку її функціоналу на локальному хостингу, необхідно розгорнути систему для широкого доступу. Передусім необхідно скопіювати як FrontEnd, так і BackEnd. Командою “npm run build” здійснюємо транспіляцію вихідного коду бекенду із TypeScript в JavaScript, який платформа NodeJS вже зможе інтерпретувати, після чого розпочнеться робота бандлера Webpack. Окрім трансформації TS в JS, він перетворить код ECMAScript 2015+ на зворотну сумісну версію JavaScript для старих браузерів чи середовищ. Основний файл бекенду, до якого задіюється транспіляція, це knh.ts:

```
import Express from 'express' import { api } from './src/routers'
import { requestHandler } from './src/middleware' const { PORT } =
process.env new Express()
  .use(requestHandler)
  .use(api.routes())
  .listen(PORT, () => console.log('≡ Launching @KNH'))
```

Подальша робота бекенду запускається командою “node dist/knh.js”. У випадку “живого” сервера використали програму Process Manager 2 (PM2), яка є відкритим вихідним кодом, розробленим Node.js та менеджером процесів, що допомагає розробникам управляти Node.js. Порівняно з іншими менеджерами процесів, такими як Supervisor, Forever, Systemd, ключовими особливостями PM2 є автоматичне збалансування навантаження програми, декларативна конфігурація програми, система розгортання та моніторинг. Скрипт із виклику knh.js, а також додаткові відомості про конфігурацію системи описані у файлі ecosystem.config.js:

```
module.exports = {
  apps: [{
    name: 'knh',
    script: './dist/knh.js',
    instances: 1,
    max_memory_restart: '128M',
    log_date_format: 'MM/DD HH:mm',
    env: {
      PORT: 8007,
      HASH_KEY: '7+frf9pyVhQL0yw5YYd0MAso08c9hd01',
      POSTGRES: 'postgres://username:password@localhost/knh'
    }
  }
}
```

Виходячи із цього файла, можна побачити основний ключ, а також інформацію про те, що бекенд обслуговується на внутрішньому порті 8007, із використанням ngix як зворотного проксі (для доступу до системи через зовнішні порти 443/80). Під час функціонування інформаційна система

використовує вхідні дані, для виконання покладених на неї функцій контролю якості поліетилену. До вхідних даних інформаційної системи належать: час роботи виробничого процесу; каталізатор; фракція; розчинник; метанол; добавки; метиловий спирт; технічні засоби.

1. Для належного їх функціонування необхідно реалізувати валідацію даних, яка міститься у файлі `validators.ts`:

```
export const validateSessionCreation: IMiddleware = async (ctx, next) =>{
```

Важливий крок – деструктуризація у цьому файлі вхідних даних із тіла запиту на локальні змінні:

```
const { time, catalizator, fraction, disolver, methanol, additive, methyl,
gatherer, granulator, centrifugeFirst, centrifugeSecond, centrifugeThird
} = ctx.request.body as TSessionCreation
```

Після цього виконується перевірка наявності змінних (якщо змінні відсутні, інформаційна система надсилає помилку із кодом 400 і відповідне повідомлення):

```
if (time === void 0 || catalizator === void 0 || fraction === void 0 ||
disolver === void 0 || methanol === void 0 || additive === void 0 || methyl
=== void 0 || gatherer === void 0 || granulator === void 0 || centrifugeFirst
=== void 0 || centrifugeSecond === void 0 || centrifugeThird === void 0)
throw new ApiError(400, 'Missing required fields')
```

Наступним кроком є перевірка відповідності типів змінних (якщо відповідність не підтверджена, інформаційна система надсилає помилку із кодом 400 і характерне повідомлення):

```
if (typeof time !== 'number' || typeof catalizator !== 'string' || typeof
fraction !== 'string' || typeof disolver !== 'number' || typeof methanol !==
'number' || typeof additive !== 'number' || typeof methyl !== 'boolean' ||
typeof gatherer !== 'boolean' || typeof granulator !== 'boolean' || typeof
centrifugeFirst !== 'boolean' || typeof centrifugeSecond !== 'boolean' ||
typeof centrifugeThird !== 'boolean') throw new ApiError(400, 'Type
mismatch')
```

Останнім елементом файла є здійснення перевірки технічних засобів, а також меж величини вхідних даних та допустимості їх заповнення:

```
if (time < 1 || time > 240) throw new ApiError(400, 'Bad time value') if
(+catalizator < 1 || +catalizator > 1000) throw new ApiError(400, 'Bad
catalizator value')
if (+fraction < 1 || +fraction > 1000) throw new ApiError(400, 'Bad fraction
value')
if (disolver < 1 || disolver > 100) throw new ApiError(400, 'Bad disolver
value')
if (methanol < 1 || methanol > 100) throw new ApiError(400, 'Bad methanol
value')
if (additive < 1 || additive > 100) throw new ApiError(400, 'Bad additive
value')
if ((centrifugeThird && (!centrifugeSecond || !centrifugeFirst)) ||
(centrifugeSecond && !centrifugeFirst)) throw new ApiError(400, 'Bad
centrifuge configuration') return next() }
```

2. Після виконання валідації даних стає доцільною подальша робота із вхідними даними, яка описана у файлі `services.ts`:

```
export const onCreateSession: TAuthorizedMiddleware = async (ctx) => {
```

Під час виконання експорту здійснюється деструктуризація вхідних даних із тіла запиту на локальні змінні:

```
const { time, catalizator, fraction, disolver, methanol, additive, methyl,
gatherer, granulator, centri fugeFirst, centri fugeSecond, centri fugeThird } =
ctx.request.body as TSessionCreation const { id } = ctx.state
```

Отримуємо об'єкт користувача або повертаємо помилку, якщо такого не існує:

```
const user = await User. findByPk(id)
if (!user) throw new ApiError(404, 'User not found')
```

Виконуємо перевірку та визначаємо стан виробничого процесу. Якщо процес активний – отримуємо характерне повідомлення, якщо ж сеанс неактивний, створюємо нову сесію:

```
if (ctx.state.session) throw new ApiError(409, 'An active or paused session
already exists')
```

```
const session = await user.createSession({ disolver, methanol, additive,
methyl, gatherer, granulator, centri fugeFirst, centri fugeSecond,
centri fugeThird,
time: time * 60, timeLeft: time * 60, catalizator: +catalizator, fraction:
+fraction, status: 'active'})
```

Здійснюємо запуск виробництва, із видаванням результатів через кожні дві секунди. Після цього повертаємо користувача до сесії:

```
createSimulation(session, 2000) ctx.state.data = { session: session.toJSON()
} }
```

Основою для будь-якої звітності та аналізу виробничого процесу є вихідні дані. В цьому випадку вони зображені у вигляді рухомого рядка, під час перегляду якого клієнт користувача кожні дві секунди запитує сервер про нові результати. Після завершення сесії, навіть якщо це екстрене завершення, всі вихідні дані записуються у відповідну базу даних інформаційної системи, яку згодом можна переглянути, виконати відповідний аналіз та надати звітність. Робота із вихідними даними описана у файлі `services.ts`. Її структура така: здійснюємо експорт та констатуємо сесію. Параметр `skip` вказує на кількість результатів, які необхідно пропустити від початку списку рядка:

```
export const onGetResults: TAuthorizedMiddleware = async (ctx) => { const {
session } = ctx.state
```

```
const skip = ctx.query.skip as string | undefined
```

Перевіряємо наявність роботи сесії, після цього виконуємо запит до бази даних для отримання результатів (із можливим “пропуском” перших результатів):

```
if (!session) throw new ApiError(404, 'No active session') const results =
await session.getResults({ offset: skip ? +skip : 0, raw: true })
```

Останній крок – повернення результатів користувачу системи:

```
ctx.state.data = { results, session: session.toJSON() } }
```

БД створена для зберігання інформації про користувачів системи та про вихідні дані виробничого процесу, який прив'язаний до користувача. Як зазначено вище, запити до БД виконуються за допомогою ORM технології, а сама база створена за допомогою PostgreSQL. Графічно ER-діаграму бази даних зображено на рис. 16. БД містить три таблиці, а саме:

- *Users*. У цій таблиці містяться відомості про користувача, такі як: id-користувача, email-адрес, ім'я, прізвище, дата створення.
- *Session*. Містить id-сесії, id-користувача, статус сесії, відомості про сесію, відомості про вхідні дані, а також про технічні засоби.
- *Result*. Містить id-результату, id-сесії та вихідні дані системи.

Відношення між *Users* та *Session* як один до багатьох, *Session* до *Result* також один до багатьох. Графічно ER-діаграму бази даних зображено на рис. 13.

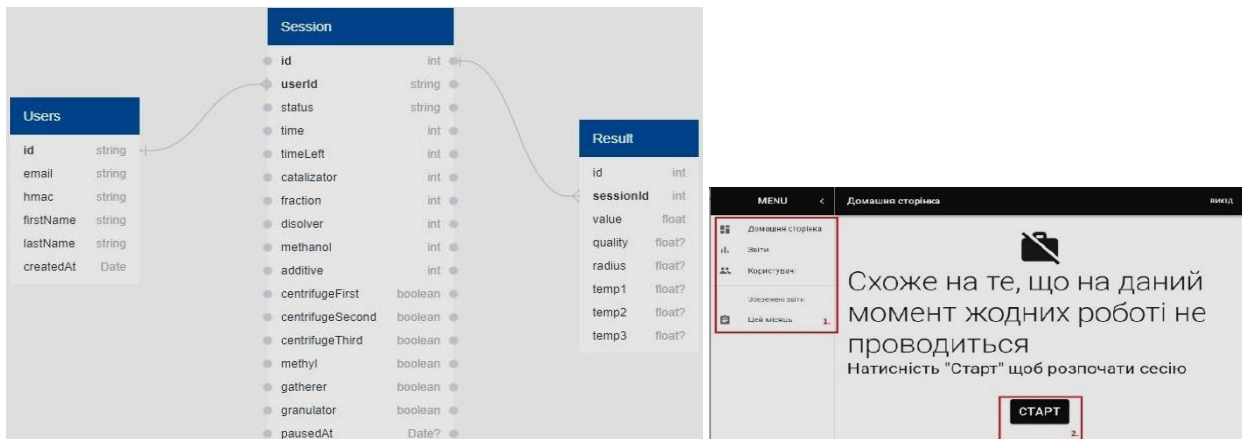


Рис. 13. ER-діаграма бази даних інформаційної системи (рис. 14). Навігація головного меню

Кожна інформаційна система, незалежно від рівня складності, має певний набір правил та вимог до користування нею. Логічно, що чим більша система, тим пересічному користувачу важче з нею працювати, особливо на початкових етапах, які характеризуються ознайомленням із системою.

Для системи контролю якості виробництва поліетилену інструкція користувача необхідна, оскільки допомагає користувачам ефективно запускати та контролювати виробництво. Оскільки розроблена інформаційна система надає повний доступ до процесу виробництва, неналежне користування нею хоч і не може спричинити фатальних наслідків для технічних засобів виробництва (з огляду на контроль процесу штучним інтелектом), однак помилкові дії користувача можуть призвести до браку виготовленої продукції. З огляду на зручність користування інформаційною системою, вся необхідна інструкція для користувача міститься безпосередньо у самій системі, на спеціально відведеній сторінці, перейти до якої можна з головного меню, натиснувши відповідне посилання (детальне зображення зв'язків між сторінками наведено на рис. 15). Таке рішення дасть змогу користувачеві звертатися до інструкції незалежно від місця його перебування на виробництві та за короткий проміжок часу приймати ефективні рішення. Навігацію по головному меню системи зображено на рис. 14 зліва (виділено червоним і пронумеровано одиницею), початок роботи із системою – посередині із кнопкою “СТАРТ” (пронумеровано двійкою). Інформаційна система містить три класи вирішуваних завдань. Відповідно до вимог оформлення інструкції користувача, опишемо детально кожне із завдань та чіткі способи їх виконання.

1. *Контроль якості поліетилену.* Цей процес вимагає від користувача належного введення параметрів для виготовлення продукції належної якості відповідно до отриманої документації із вказівками щодо необхідної продукції. Для виконання цієї операції необхідно дотримуватись таких кроків: отримати документ, в якому чітко вказано параметри продукції, яка повинна бути виготовлена; відповідно до документа заповнити кожне поле необхідним параметром; увімкнути всі технічні засоби виробництва; запустити виробництво; здійснювати аналіз виробництва. У разі отримання відповідних вказівок призупинити або повністю зупинити виробництво. Дочекались повного завершення виробництва і закінчити роботу із сесією.

2. *Перевірка кожного із етапів виробництва.* Цей процес є надзвичайно важливим, оскільки дає змогу здійснити початкову діагностику всього виробничого процесу, виявити проміжок у виготовленні продукції, на якому система перестає правильно працювати. Завдяки цьому процесу можна без значних затрат часу виявити неробочий пристрій, налаштувати новий пристрій, здійснити тренування стосовно роботи із системою, не запускаючи повний цикл виготовлення продукції. Для виконання цієї операції необхідно дотримуватись таких кроків:

- Отримати документ із дозволом частково запустити виробничий процес.

- Відповідно до документа заповнити необхідні параметри (у цьому випадку не обов'язково заповнювати усі параметри).

- Увімкнути тільки ті технічні засоби, які застосовують на необхідних користувачеві етапах виробництва.

- Запустити виробництво та вести моніторинг системи.

- Після отримання необхідних даних завершити роботу сесії.

3. *Аналіз та звітність.* Процес, який необхідний для кожного виробництва, і застосовується для визначення ефективності роботи інформаційної системи, користувача, кількості сеансів, які він здійснив. На основі отриманих результатів визначається заробітна плата користувачів інформаційної системи, нові вимоги до показників якості продукції, оцінка ефективності виробництва, підраховуються витрати на виготовлення та реалізацію. Для здійснення операції необхідно виконати такі кроки:

- Після запуску системи моніторити її роботу на основі отриманих параметрів.

- У разі необхідності можливо достроково завершити роботу, однак необхідно все одно виконати подальші дії.

- Після завершення виробництва перейти до бази даних, в якій записаний весь робочий процес та його показники якості.

- Виконати сортування за необхідними параметрами та зберегти їх окремим файлом.

- Надіслати дані на відповідну електронну пошту, зазначену в документі із вказівками запуску.

- Завершити роботу із інформаційною системою.

Строго дотримуючись кожної із вказівок для трьох завдань, інформаційною системою може користуватися навіть низькокваліфікований працівник. Завдяки штучному інтелекту інформаційна система буде повністю ізольована від його неналежних дій, можливих у разі користування аналогічними системами. Як кожна програма, що передбачає складні обчислення, ця інформаційна система має певні обмеження на застосування. Здебільшого ці обмеження введено спеціально, щоб захистити виробництво від неналежних дій користувачів.

Основним обмеженням інформаційної системи є захист виробничого процесу від паралельного запуску виробництва. Тобто якщо в систему, в якій користувач запустив виробництво, увійшло декілька нових користувачів, вони не зможуть здійснити жодних змін у виробничому процесі та вплинути на роботу системи доти, доки перший користувач не завершить свою роботу. Це обмеження дає змогу створити умови, за яких мінімізується вплив сторонніх користувачів. Другим важливим обмеженням є введення значення параметрів для виготовлення продукції відповідно до відсоткових пропорцій. Для виготовлення поліетилену немає чітких вказівок на параметри, вони можуть міститись у відповідних проміжках, залежно від виду продукції, її товщини, міцності, еластичності тощо, однак наявні певні обмеження із виготовлення. Ці обмеження характеризуються такими показниками, згідно із технологією виготовлення поліетилену:

- Величини каталізатора і фракції повинні бути рівні, тобто становити по 50 % кожний.

- Відсоток розчинника повинен перевищувати 30 %.

- Метанол повинен становити менше ніж 40 %.

- Відсоток добавки повинен бути меншим, ніж 8 %.

- Під час запуску виготовлення всі технічні елементи повинні бути увімкнені.

- Під час запуску діагностики інформаційної системи повинні бути заповнені всі параметри, увімкнений як мінімум перший змішувач та поданий метиловий спирт.

Варто зазначити, що для користувачів є відповідні вказівки на обмеження (рис. 15), це можуть бути як заборони (вказані червоним кольором, 1), так і попередження (жовтим, 2).

Важливо також забезпечити розуміння, на якому етапі роботи перебуває виробничий процес. Відповідь можна отримати, переглянувши дорожню карту цього процесу (рис. 16), в якій пройдені етапи, а також етап, на якому перебуває система, підсвічені чорним, а неактивні етапи відповідно сірим кольором.

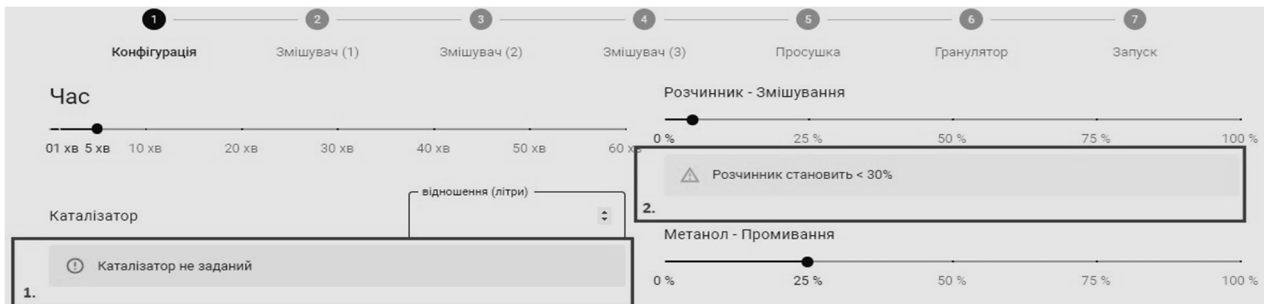


Рис. 15. Вказівки на обмеження та дотримання вимог системи

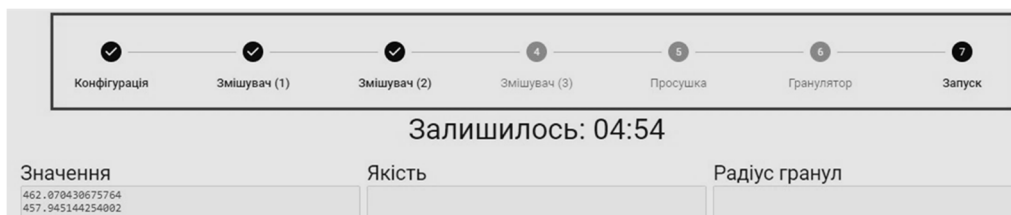


Рис. 16. Дорожня карта виробництва

Ураховуючи ці вимоги, які основані на інструкції користувача та є одним із найважливіших розділів цієї інструкції, можна переходити до безпосереднього аналізу контрольного прикладу системи. Виконаємо аналіз контрольного прикладу. Для цього здійснимо перевірку роботи у двох режимах, а саме звичайному режимі та режимі діагностики системи. Після цього на підставі отриманих результатів згрупуємо дані на основі однієї із сесій системи. Як зазначено в інструкції щодо користування інформаційною системою, для звичайного режиму контролю якості продукції необхідно задати всі параметри, зберігаючи необхідні пропорції та дотримуючись вказівок системи. Заповнені параметри відображено на рис. 17. Після належного заповнення всіх параметрів відбувається безпосередній контроль якості технологічних процесів штучним інтелектом. Користувач в цей час має змогу спостерігати за перебігом подій, виконувати відповідний аналіз даних, а також за необхідності завершувати виробництво достроково. Оскільки штучний інтелект бере на себе весь контроль за виробництвом, користувач інформаційної системи може виконувати обхід по території, на якій відбувається виробничий процес, і оцінювати його.

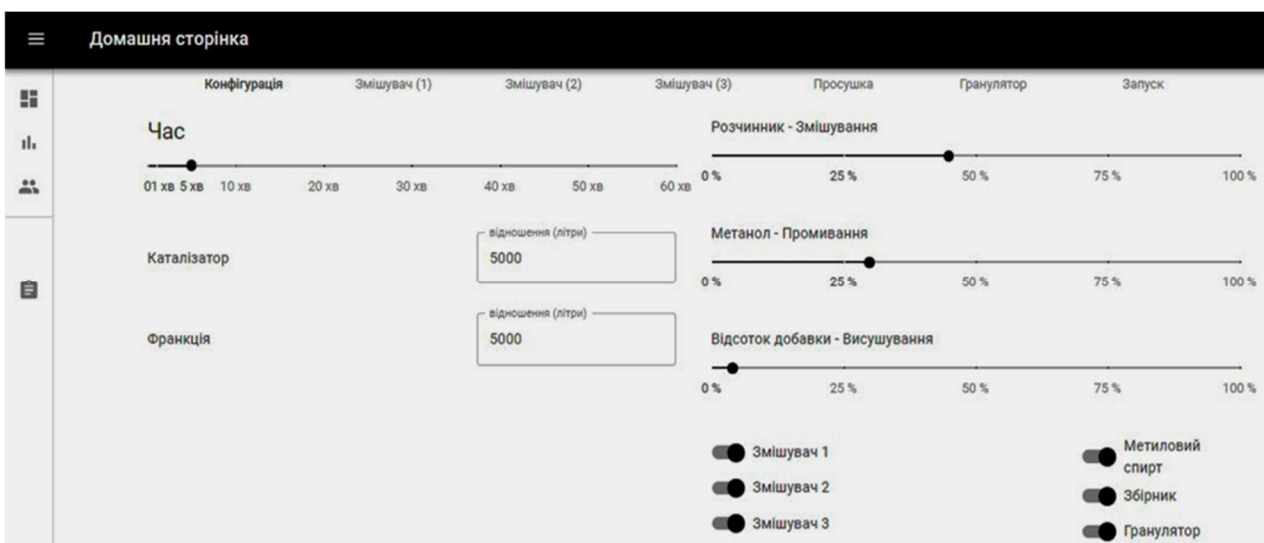


Рис. 17. Заповнення даними для звичайного режиму

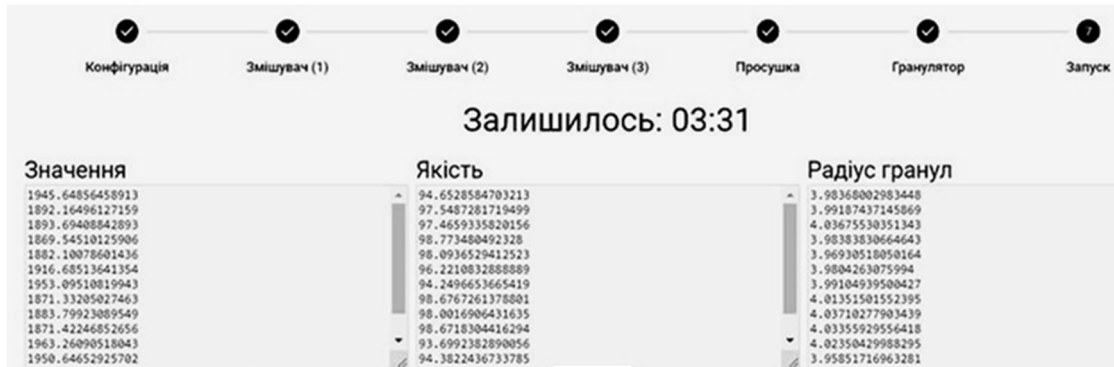


Рис. 18. Виведення показників якості

Для режиму діагностики інформаційної системи виконуємо схожі дії із заповненням параметрами, однак вмикаємо тільки ті технічні засоби, які нам потрібно діагностувати. В цьому режимі допускається введення параметрів, які не відповідають вимогам, щодо виготовлення поліетилену. Цей фактор зумовлений тим, що під час діагностики не виготовляється кінцевий продукт, а отже, визначення і контроль якості у цьому випадку недоцільні. Заповнення параметрів для діагностики відображено на рис. 19. Після того як всі параметри заповнені, а відповідні технічні засоби увімкнені, натискаємо кнопку запуску виробництва та спостерігаємо розраховані значення опрацьованої сировини (рис. 20). Останнім етапом є виведення детальної інформації про завершену сесію. Для цього відкриваємо базу даних і шукаємо потрібну нам сесію або ж відповідним запитом здійснюємо виведення потрібної нам сесії. Виводимо параметри для третьої сесії, написавши такий запит: `select * from results where "sessionId" = 3.`

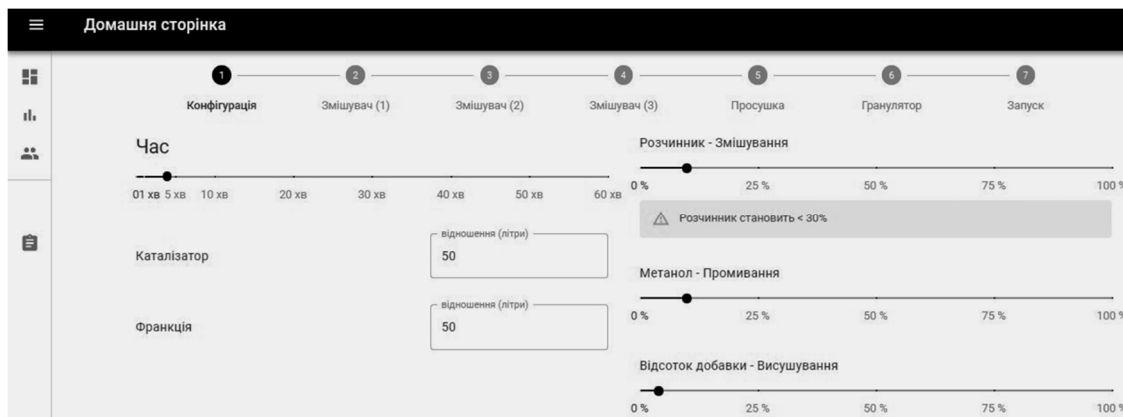


Рис. 19. Заповнення даними для режиму діагностики

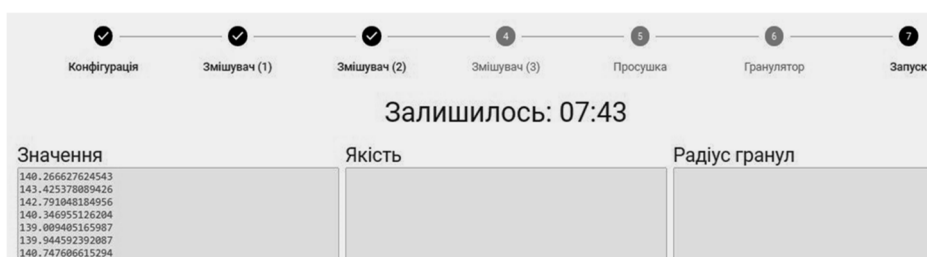


Рис. 20. Виведення показників опрацьованої сировини

На основі цього запиту отримуємо відповідні результати:

id	value	quality	radius	temp1	temp2	temp3	createdAt	sessionId
21	39.9550776954334	97.5152257112132	3.98855917648323	51.1463108629378	103.019227524064	73.6476469027823	2021-03-06 12:22:37.945+00	3
22	41.0065213235054	94.8182728052085	3.95434375928644	50.4500052889284	100.430711102371	70.2375690641921	2021-03-06 12:22:39.944+00	3
23	41.6015257614215	93.2920864219538	4.03594070267298	50.37412668849	104.394565231303	74.739047067402	2021-03-06 12:22:41.944+00	3
24	39.943797890262	97.5441586198478	4.02891020663115	50.223216422223	104.407376082843	71.5841013844989	2021-03-06 12:22:43.944+00	3
25	40.7874181933317	95.3802723341041	4.04479326153905	51.1246191079625	104.184242849001	74.8134557698383	2021-03-06 12:22:45.944+00	3
26	39.735557360269	98.07829537091	3.97020602955835	50.8552791876997	101.815575358276	72.0178396197633	2021-03-06 12:22:47.944+00	3
27	40.3601379052054	96.4762462731482	4.04274852603671	50.6855127392177	104.640361340783	75.3090263318143	2021-03-06 12:22:49.944+00	3
28	40.172862847858	96.9566067952442	3.97345087660016	50.9657075346338	102.147008751541	71.1700948617442	2021-03-06 12:22:51.944+00	3
29	40.5271455060874	96.0478717768858	4.04122997377257	51.4312042984719	101.794328852707	71.2395968565669	2021-03-06 12:22:53.944+00	3
30	40.306132361429	96.6147704929347	3.97461146889697	50.5251890626474	104.566616749937	74.6299867370787	2021-03-06 12:22:55.945+00	3
31	41.0977190266314	94.5843506966904	4.04562665013886	50.8638652500321	100.19356395651	74.7384317214525	2021-03-06 12:22:57.946+00	3
32	40.2876311462256	96.6622261099315	3.99105366073324	50.9416441384734	102.124336747016	70.0689540547853	2021-03-06 12:22:59.946+00	3
33	40.672375753718	97.2044506191714	4.03462856057163	50.5451448361222	100.046288091557	71.038654906007	2021-03-06 12:23:01.947+00	3
34	39.0679338873304	99.7907495789974	4.00508657721883	50.4490316105095	101.739637015183	73.3038703512827	2021-03-06 12:23:03.947+00	3
35	39.596440507744	98.4351300976366	4.02701599335634	50.1139264586728	102.202103526576	71.7435767379913	2021-03-06 12:23:05.947+00	3
36	39.2702076799125	99.2719173010244	3.95632276314834	50.7782605688779	102.483481717883	71.4231071579776	2021-03-06 12:23:07.948+00	3
37	41.0780441864343	94.634816661796	3.9571620393258	50.4593307245735	102.007893461234	71.7644405633339	2021-03-06 12:23:09.956+00	3
38	41.2465138453157	94.2026919867653	4.01190236549625	50.686986371358	104.29035257763	74.347840104384	2021-03-06 12:23:11.948+00	3
39	40.0937179926089	97.1596133489581	4.01625230437202	50.1370978679864	104.400007947638	75.3387884196779	2021-03-06 12:23:13.949+00	3
40	39.6976435574602	98.1755442751145	3.97866760380224	50.8593054889857	102.285837938077	72.614392656545	2021-03-06 12:23:15.95+00	3
41	41.5232455799578	93.4928748309083	4.01084615947817	50.9575221240574	104.98941289228	72.7459093970018	2021-03-06 12:23:17.949+00	3
42	41.088607756735	94.6077211039747	4.03723511071568	50.947965037789	101.371324790356	70.5812295846617	2021-03-06 12:23:19.949+00	3
43	41.0249761331033	94.77093621859	3.99863573670003	51.0138631044161	102.470968177173	74.9380030034222	2021-03-06 12:23:21.949+00	3
44	39.1502016968149	99.5797326476699	4.0026408180025	51.3683701232656	104.199153617271	70.4621034774534	2021-03-06 12:23:23.95+00	3
45	39.7407333498254	98.0650189576978	4.01454824204589	51.4863677832662	103.116141425216	75.2170643651634	2021-03-06 12:23:25.951+00	3
46	41.5199413114353	93.5013505361685	3.99043056191327	50.3768340571408	100.702866921604	74.7706256941229	2021-03-06 12:23:27.952+00	3
47	40.9281701345223	95.0192436049504	3.95587834098219	50.4007965903485	102.699849459606	70.6029295775282	2021-03-06 12:23:29.952+00	3

Рис. 21. Виведення показників якості технологічних процесів для третьої сесії

Щоб краще відобразити повний функціонал інформаційної системи контролю якості виробництва поліетилену, її обмеження, вхідні, вихідні дані, особливості та детальну структуру, ми навели детальний її опис та висвітлили основні технічні характеристики в інструкції. В описі було подано ієрархію файлової структури інформаційної системи, із деталізацією головних директорій та файлів, проаналізовано програмні засоби, використані для реалізації інформаційної системи, її тестування, розгортання та подальшої підтримки. Окрім цього, описано та графічно зображено логічну структуру сторінок інформаційної системи, яка є деревоподібною структурою та допомагає користувачам краще зрозуміти навігацію по системі. Таку структуру вибрано з огляду на необхідність початкової верифікації користувачів та надалі вибору необхідного функціоналу. Також в інструкції детально описано головні файли, що відповідають за операції із вхідними та вихідними даними, викликом та завантаженням інформаційної системи. Детально описано структуру бази даних із відповідною ER-діаграмою. Важливим пунктом інформаційної системи є набір правил та рекомендацій для її користування, саме для цього було створено інструкцію користувача, яка містить детальні, покрокові дії щодо користування нею, зазначено основні функції, проілюстровано її роботу на контрольному прикладі, а також основні особливості інтерфейсу. Для перевірки відповідності інформаційної системи описаним вимогам, а також аналізу ефективності роботи здійснено її апробацію. Виконуючи ці дії, реалізували кожен із трьох основних функціоналів системи, а саме: контроль якості поліетилену, перевірку кожного з етапів виробництва, аналіз звітності. Виконавши детальний аналіз, ми підтвердили відповідність системи заданим вимогам. Під час апробації інформаційної системи жодних помилок у роботі не виявлено.

Висновки

Аналіз етапів проектування та програмної реалізації інформаційної системи для здійснення контролю якості виробництва поліетилену засвідчив, що до основного її функціоналу входять такі основні можливості, як засоби зв'язку між операторами та вертикально підприємства, формування звітності роботи користувача із інформаційною системою, що містить всі основні параметри виробничого процесу, а також можливість самостійного контролю виробничого процесу. Що стосується методів візуалізації даних, варто наголосити, що в цій інформаційній системі є можливість графічного перегляду даних температури у вигляді графіків, а також створена дорожня карта етапів виробництва. Доцільність створення цієї інформаційної системи полягає у тому, що її використання

дає змогу зменшити кількість персоналу для обслуговування виробничого процесу, а також, разом із цим, зменшити затрати сировини у зв'язку із контролем якості продукції. Саме завдяки використанню штучного інтелекту інформаційна система сприяє зменшенню необхідності втручання оператора системи, поліпшується якість технологічних процесів виготовлення кінцевого продукту. В інформаційній системі використано три основні методи контролю якості виробництва, а також на їх основі кілька комбінованих методів. Створений в цьому програмному забезпеченні алгоритм надає нові можливості для економнішого, а головне якіснішого виготовлення поліетилену із залученням меншої кількості працівників, що сприяє значному збільшенню прибутку. Оскільки розроблювана система ґрунтується на сучасних методах контролю якості технологічних процесів, має можливість масштабування відповідно до величини підприємства та містить весь наявний функціонал для ефективної роботи, на неї буде великий попит на ринку і в малих підприємств хімічних галузей як на цілісне програмне забезпечення, і у великих, як на модульну частину. Ринок програмного забезпечення дедалі більше розширюється і з'являються нові види інтерфейсів, засоби опрацювання даних, а також засоби контролю апаратних засобів. Щоб залишатися конкурентоспроможним, необхідно мати вагомі переваги, які полягають у розширенні функціоналу програмного продукту, зростанні швидкості обчислення, поліпшенні якості обчислень. Інформаційна система дає змогу використовувати основні методи контролю технологічних процесів, зокрема такі як: вимірювальний – на основі технічних засобів вимірювання і контролю; ним визначають, наприклад, масу виробу, швидкість; реєстраційний – на основі спостереження і підрахунку кількості певних подій, предметів чи витрат; розрахунковий – на основі використання теоретичних або емпіричних залежностей показників якості продукції від її параметрів (для визначення маси товару, продуктивності, потужності, міцності); експертний – на основі рішення експертів; соціологічний – на основі збирання та аналізування думок фактичних і можливих споживачів продукції.

Для зменшення витрат людських ресурсів спроектовано та реалізовано алгоритм для цього програмного забезпечення, який на основі штучного інтелекту самостійно контролює основні аспекти виготовлення поліетилену, зменшуючи кількість необхідних кваліфікованих працівників і їхню участь у технологічному процесі. Цей підхід передбачає введення оператором початкових даних, які сприяють поліпшенню якості виготовлення поліетилену відповідно до передбачених вимог, а штучний інтелект коригує параметри для досягнення максимально високих показників якості. Це дає змогу мінімізувати вплив людського фактора на виробничий процес, а правильний розподіл сировини – забезпечити її значну економію. Із урахуванням зазначеного можна зробити висновок, що створення цієї інформаційної системи є доцільним вибором і її впровадження надасть можливість підприємцям забезпечити економію сировини, відповідно збільшуючи дохід від виготовлення, а кінцевим користувачам – отримати якіснішу продукцію. Саме ці фактори визначають високий попит на це програмне забезпечення та дозволять використовувати його як для безпосереднього виготовлення поліетилену, так і для подальшої модернізації виробництва з метою виготовлення полібутилену та поліпропілену.

Для досягнення поставленої мети насамперед виконано детальний аналіз систем контролю якості виробництва. Особливу увагу приділено безпосередньо системам контролю якості у хімічній промисловості. Під час проектування системи контролю якості поліетилену необхідно враховувати безліч чинників. На основі аналізу робіт в міжнародному і національному сегментах визначено необхідність якісного моніторингу та контролю процесу виробництва, мінімізації затрат на сировину, фінансових затрат, а також зменшення складності навчання роботи із цими системами. Виявлено необхідність розроблення системи, яка не прив'язана до певного пристрою чи панелі, а побудована за принципом мультиплатформності, що забезпечує мобільність роботи оператора системи. Порівняння схожих систем контролю якості свідчить, що з безлічі сучасних систем неможливо виділити одну, яка була б універсальною і найкращою з усіх аспектів, однак їхні переваги та недоліки були враховані під час розроблення власної інформаційної системи контролю якості поліетилену. Варто зауважити, що під час розроблення системи було неможливо обійтися без нехтування певними

позитивними факторами, щоб забезпечити більший фізичний та програмний функціонал, а також безпеку системи. Маючи цілісне уявлення про вимоги до системи, її функціонал та особливості, ми здійснили системний аналіз об'єкта дослідження, в якому чітко сформуваємо ієрархічні рівні, конкретизували функціонування системи та створили ієрархію задач за допомогою IDEF0 (функціональної діаграми).

З огляду на складність системи, здійснено трирівневу декомпозицію. Таке рішення допомогло чітко зрозуміти проблеми, які могли б виникнути під час розроблення, та завчасно виправити їх. Виконавши детальний аналіз та подальше порівняння програмних засобів для реалізації системи, ми виявили високу ефективність платформи NodeJS, яка містить такі засоби, як ExpressJS та ReactJS. За допомогою цих засобів система отримала сучасні модулі, надійну базу даних, красиву візуалізацію, можливість підключення проміжного програмного забезпечення, високоякісне адміністрування та оптимізацію, завдяки якій цей програмний продукт може працювати на слабких пристроях, а також у старих версіях браузерів. Створена система спрямована на покращення контролю якості, завдяки чому досягається значна економія сировини, необхідної на виготовлення продукції. Окрім економії, за допомогою засобів штучного інтелекту система забезпечує істотне поліпшення якості порівняно з аналогічними методами контролю якості.

З огляду на необхідність захисту виробничого процесу створено низку засобів, які забезпечують контроль від несанкціонованого входу в систему сторонніх користувачів чи помилкових дій користувачів. До них належить заборона входу новим користувачам під час запуску виробництва, здійснення верифікації, запобігання системи введенню критичних параметрів, реалізації інструкції для користувачів. Розроблена система контролю якості може бути використана як автоматизоване робоче місце користувача, а також бути переносною, залежно від того, на якому пристрої її відкрито (персональний комп'ютер, ноутбук, планшет, телефон). Для максимальної ефективності системи рекомендовано використовувати її на планшеті, цей пристрій дасть змогу користувачу пересуватися по підприємству і виконувати додатковий нагляд за його роботою. На відміну від телефону, планшет має більшу діагональ екрана, що дасть змогу безпомилково виставляти необхідні параметри та економити час під час їх задавання. Програмна перевірка контрольного прикладу та подальший аналіз отриманих результатів підтвердили відповідність інформаційної системи вимогам та відповідно, досягнення поставленої мети. З огляду на можливості платформи NodeJS та вибраних програмних засобів реалізації, система має великий потенціал та всі необхідні умови для подальшого розширення, із додаванням контролю якості суміжних кінцевих продуктів у цій галузі, таких як полібутилен, поліпропілен.

Список літератури

1. Kohl H. (2020). Standards for management systems. Management for Professionals. DOI: 10.1007/978-3-030-35832-7
2. Fitriani S., Sofyan Y. (2020). Simulator Human Machine Interface (HMI) using visual basic on the SCADA system. In *IOP Conference Series: Materials Science and Engineering*, Vol. 830, No. 3, p. 032016. IOP Publishing. DOI: 10.1088/1757-899X/830/3/032016
3. Faccia A., Petratos, P. (2021). Blockchain, enterprise resource planning (ERP) and accounting information systems (AIS): Research on e-procurement and system integration. *Applied Sciences*, 11(15), 6792. DOI: 10.3390/app11156792.
4. Sarvaiya H. B., Keswani R. A., Pannase D., Patil S., Turankar P., Janbandhu A., Patle V. (2022). Recent Developments in SCADA System for Remote Industry and It's Experimental Implementation. *International Journal of Engineering Technology and Management Sciences*. Website: ijetms. in, 6(3). DOI: 10.46647/ijetms.2022.v06i03.005
5. Upadhyay D., Sampalli S. (2020). SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations. *Computers & Security*, 89, 101666. DOI: 10.1016/j.cose.2019.101666
6. Tien N. H. (2019). International economics, business and management strategy. Dehli: Academic Publications. URL: https://www.researchgate.net/profile/Nguyen-Tien-32/publication/338570555_International_Economics_Business_and_Management_Strategy/links/63031e46ceb9764f7214b825/International-Economics-Business-and-Management-Strategy.pdf

7. Ceko E. (2021). On relations between creativity and quality management culture. *Creativity studies*, 14(1), 251–270. DOI: 10.3846/cs.2021.12154
8. World Health Organization. (2021). WHO global air quality guidelines: particulate matter (PM_{2.5} and PM₁₀), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide: executive summary. URL: <https://apps.who.int/iris/bitstream/handle/10665/345334/9789240034433-eng.pdf>
9. Baran M., Kuzmin O., Bublyk M., Panasyuk V., Lishchynska K. (2021). Information System for Quality Control of Polyethylene Production in a Circular Economy. *CEUR Workshop Proceedings*, Vol. 2917, 465–502. URL: <https://ceur-ws.org/Vol-2917/paper34.pdf>
10. Mukherjee P., Acharyya A., Dash N., Alam A., Barik K. C., Behera S., Dash R. N. (2022). Linear Bottle filling system using Simatic S7-200 and S7-1200 PLC with HMI control. In *International Conference on Industrial Electronics: Developments & Applications*, 78–82. DOI: 10.1109/ICIDeA53933.2022.9970172
11. Yamazaki M., Nakagawa, W. (2021). How open systems support end users: End users have been vocal about wanting open systems, and automation vendors are responding by delivering systems compliant with international standards. *Control Engineering*, 68(4), 30–34.
12. Schneider, M. J. (2017). Anti-inflammatory effects of herbal preparations in cytokine-challenged normal human colon cells: doctoral dissertation. DOI: 10.25358/openscience-843
13. Gozhyj A., Vysotska V., Yevseyeva I., Kalinina I., Gozhyj V. (2019). Web Resources Management Method Based on Intelligent Technologies. In: Shakhovska, N., Medykovskyy, M. (eds) *Advances in Intelligent Systems and Computing III. CSIT 2018. Advances in Intelligent Systems and Computing*, Vol. 871. Springer, Cham. DOI: 10.1007/978-3-030-01069-0_15
14. Lytvyn V., Pukach P., Bobyk I., Vysotska, V. (2016). The method of formation of the status of personality understanding based on the content analysis. *Eastern-European Journal of Enterprise Technologies*, (5 (2)), 4–12. DOI: 10.15587/1729-4061.2016.77174
15. Vysotska V., Chyrun L., Chyrun L. (2016). Information technology of processing information resources in electronic content commerce systems. In 2016 XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 2016, 212–222, DOI: 10.1109/STC-CSIT.2016.7589909.
16. Luis P., Van der Bruggen B. (2014). Exergy analysis of energy-intensive production processes: advancing towards a sustainable chemical industry. *Journal of Chemical Technology & Biotechnology*, 89(9), 1288–1303. DOI: 10.1002/jctb.4422
17. Michler G. H., Balta-Calleja F. J. (2016). Mechanical properties of polymers based on nanostructure and morphology. CRC Press. DOI: 10.1201/9781420027136
18. Han Y., Dai L. (2019). Conducting polymers for flexible supercapacitors. *Macromolecular chemistry and physics*, 220(3), 1800355. DOI: 10.1002/macp.201800355
19. Gozhyj A., Kalinina I., Vysotska V., Gozhyj V. (2018). The method of web-resources management under conditions of uncertainty based on fuzzy logic. In 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, 343–346. IEEE. DOI: 10.1109/STC-CSIT.2018.8526761
20. Berko A., Vysotska V., Chyrun L. (2014). Features of information resources processing in electronic content commerce. *Applied Computer Science*, 10(2). URL: <https://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-ad40e0fb-e999-4602-a6ab-91a778d003dc/c/berko-vysotska.pdf>
21. Vysotska V., Chyrun L., Kozlov P. (2016). Analysis of business processes in electronic content-commerce systems. *ECONTECHMOD: an international quarterly journal on economics of technology and modelling processes*, 5(1), 111–125. URL: <https://bibliotekanauki.pl/articles/411167.pdf>
22. Chyrun L., Vysotska V., Laba, R. (2016). Information Resources Analysis in Electronic Content Commerce Systems. *Applied Computer Science*, 12(1), 48–66. URL: <https://www.infona.pl/resource/bwmeta1.element.baztech-3acd5d28-f592-435e-a72b-367b6af05cc2>
23. Vysotska V., Berko A., Lytvyn V., Kravets P., Dzyubyk L., Bardachov Y., Vyshemyrska S. (2021). Information Resource Management Technology Based on Fuzzy Logic. *Advances in Intelligent Systems and Computing*, Vol. 1246. Springer, Cham. DOI: 10.1007/978-3-030-54215-3_11
24. Berko A., Vysotska V., Lytvyn V., Naum O. (2018). Planning the activities of intellectual agents in the electronic commerce systems. *Radio electronics, informatics, management*, 4 (47), 143–158. DOI: 10.15588/1607-3274-2018-4-14
25. NodeJS. URL: <https://nodejs.org/docs/latest-v9.x/api/>.
26. Chyrun L., Kowalska-Styczen A., Burov Y., Berko A., Vasevych A., Pelekh I., Ryshkovets, Y. (2019). Heterogeneous Data with Agreed Content Aggregation System Development. In *MoMLeT*, 35–54.

27. Bächle M., Kirchberg P. (2007). Ruby on rails. *IEEE Softw.*, 24(6), 105–108.
28. ReactJS. URL: <https://e.lanbook.com/book/97339>
29. AngularJS. URL: <http://www.webshac.com/Angular/AngularJS%20Essentials.pdf>
30. Node.js. URL: <https://nodejs.org/docs/latest-v9.x/api/>.
31. PostgreSQL. URL: <https://momjian.us/main/writings/pgsql/other/bookfigs.pdf>.

INFORMATION SYSTEM OF POLYETHYLENE QUALITY CONTROL IN THE CIRCULAR ECONOMY

Mykola Baran¹, Myroslava Bublyk¹, Lyubomyr Chyrun², Maryna Shevchenko³, Sofia Chyrun¹

¹ Lviv Polytechnic National University,
Information Systems and Networks Department, Lviv, Ukraine

² Ivan Franko National University of Lviv,
Applied Mathematics Department, Lviv, Ukraine

³ Osnabrück University, International Economic Policy Chair, Osnabrück, Germany

E-mail: mykola.baran.kn.2017@lpnu.ua, ORCID: 0000-0002-8199-0530

E-mail: Myroslava.I.Bublyk@lpnu.ua, ORCID: 0000-0003-2403-0784

E-mail: Lyubomyr.Chyrun@lnu.edu.ua, ORCID: 0000-0002-9448-1751

E-mail: mshevchenko@uni-osnabrueck.de, ORCID: 0000-0003-2165-9907

E-mail: sofia.chyrun.sa.2022@lpnu.ua, ORCID: 0000-0002-2829-0164

© Baran M., Bublyk M., Chyrun L., Shevchenko M., Chyrun S., 2023

Polyethylene quality control system is a set of methods and means of control and regulation of components that determine the level of product quality, as well as technical control at all stages of the production process. The first experience of polymerization of polyethylene at the end of the XIX century was received by the scientist Gustavson. For many years, polyethylene was created in small quantities, but in 1938 the process of industrial production was mastered by the British. At that time, the polymerization method was not yet perfect and required additional means of production control. Nowadays, each company uses different methods to control production, but given the complex chemical and technological processes, the need for a more efficient and integrated system that will control the quality of final products and at the same time save raw materials is still relevant. Goal of research: creation of a quality control system for the tasks of notification, control, management of automation systems using remote access and artificial intelligence systems. Study object: technical features of implementation of quality control system for polyethylene production enterprise. Scope of research: quality control system for polyethylene production with elements of artificial intelligence technology. The article deals with the creation of an information system for quality control of polyethylene, which will provide effective and safe management of the production process, high functionality and cross-platform with the help of web technologies and artificial intelligence. The work consists of an introduction, five chapters, conclusions and a list of sources used. In the first section the theoretical and methodological bases of research of quality control system are considered. The second section provides a systematic analysis of the object of study. The third section compares and selects the means of implementing systems. The fourth section describes the created software. In the fifth section, the economic characteristics of the software product and the calculation of the cost of the software product. According to the study, an information system for quality control of polyethylene production was created, the work of which is to establish compliance of products and processes with the requirements of regulatory and technical documentation, standard samples, provide information on the production process and maintain its stability, protect enterprise data, identify defective products in the early stages, preventing the production of substandard products.

Key words: quality control information system; control panel; web technology; means of artificial intelligence; polyethylene; chemical industry.