

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ПРОЄКТУВАЛЬНИКІВ ІНТЕРФЕЙСІВ
SCADA-CИСТЕМ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
INTELLIGENT SUPPORT OF DESIGNERS OF SCADA SYSTEMS
INTERFACES

Виконав: студент(ка) 2 курсу, групи КНТ-119м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмногозабезпечення

Корсунов К.А.

(прізвище та ініціали)

Керівник Сердюк С.М.

(прізвище та ініціали)

Рецензент Казурова А.Є.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2020 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Корсунова Костянтина Андрійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів інтелектуальної підтримки проєктувальників інтерфейсів SCADA-систем. Research and software implementation of methods for intelligent support of designers of SCADA-systems interfaces

керівник проєкту (роботи) Сердюк Сергій Микитович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “03” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) 16 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація, тестування та експериментальне дослідження програми 5. Економіко-організаційна частина. 6. Охорона праці та безпека в надзвичайних ситуаціях. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Сердюк С.М., доцент		
Економіко-організаційна частина	Пожуєва Т.О. проф.		
Охорона праці та безпека в надзвичайних ситуаціях	Коробко О. В., ст., викладач		
Нормоконтроль	Калініна М.В., асист.		

7. Дата видачі завдання “ 29 ” лютого 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2-3 тижні	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	4-5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділи 2, 3
5	Розробка програми.	7-8 тижні	Розділ 3
6	Тестування та експериментальне дослідження програмного забезпечення.	9 тиждень	Розділ 4
7	Розробка економіко-організаційної частини	10 тиждень	Розділ 5
8	Розробка частини охорони праці та безпеки в надзвичайних ситуаціях.	11 тиждень	Розділ 6
9	Оформлення пояснювальної записки та документів до неї.	12-13 тижні	
10	Нормоконтроль та рецензування.	14-15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка)

(підпис)

Корсунов К.А.

(прізвище та ініціали)

Керівник проєкту (роботи)

Сердюк С.М.

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
126 с., 21 табл., 46 рис., 2 дод., 37 джерел.

ЛЮДИНО-МАШИННИЙ ІНТЕРФЕЙС, ІНФОРМАЦІЙНА МОДЕЛЬ,
ЕРГОНОМІЧНЕ ЗАБЕЗПЕЧЕННЯ ПРОЄКТУВАННЯ, СИСТЕМА
ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ, C#, ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ.

Об'єкт дослідження – ергономічне забезпечення проєктування (ЕЗП) інтерфейсів SCADA-систем.

Предмет дослідження – інформаційна технологія (ІТ) інтелектуальної підтримки процесу ергономічного проєктування людино-машинних інтерфейсів (ЛМІ) SCADA-систем.

Мета роботи – підвищення рівня автоматизації та ефективності роботи проєктувальників інтерфейсів SCADA-систем шляхом розроблення і застосування програмного забезпечення на основі методів і моделей штучного інтелекту.

Матеріали, методи та технічні засоби: методи штучного інтелекту; структурне та об'єктно-орієнтоване програмування; ергономічне забезпечення проєктування; середовище розробки Microsoft Visual Studio 2017; мова програмування C#; операційна система Windows 7 або вище.

Результати. Дістала подальшого розвитку відома ІТ інтелектуальної підтримки процесу ергономічного проєктування ЛМІ шляхом введення в неї засобів та правил кодування технологічної інформації слуховою модальністю.

Висновки. Використання розробленої ІТ забезпечує підвищення ефективності діяльності проєктувальників інтерфейсів SCADA-систем.

Галузь використання – проєктування інтерфейсів SCADA-систем.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 126 pages, 21 tables, 46 figures, 2 appendixes, 37 sources.

HUMAN-MACHINE INTERFACE, INFORMATION MODEL, ERGONOMIC SUPPORT OF DESIGN, INTELLECTUAL SUPPORT SYSTEM, C #, INFORMATION TECHNOLOGY.

Object of study is ergonomic support of designers (ESD) of SCADA-system interfaces.

Subject of study are information technology (IT) for intelligent support of the process of SCADA-systems human-machine interfaces (HMI) ergonomic design.

The purpose of the work is to increase the level of automation and efficiency of SCADA system interface designers by developing and applying software based on methods and models of artificial intelligence.

Materials, methods, and tools: artificial intelligence methods; structural and object-oriented programming; ergonomic design support: Microsoft Visual Studio 2017 development environment; C # programming language; operating system Windows 7 or higher.

Results. The well-known IT for intelligent support of the ergonomic design of LMI process has been further developed by introducing into it means and rules for coding technological information of the auditory modality.

The implementation of the developed IT provides increase of efficiency of designers of SCADA-systems interfaces activity.

The field of use is a SCADA-systems interfaces design.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	12
1.1 SCADA-системи: загальні поняття і структура	12
1.2 Аналіз технологій побудови SCADA-систем.....	16
1.3 Постановка задач дослідження.....	22
1.4 Висновки до першого розділу.....	23
2 Матеріали і методи	24
2.1 Аналіз стану досліджень у області проєктування людино-машинних інтерфейсів SCADA-систем	24
2.2 Системний аналіз діяльності проєктувальника інтерфейсів.....	27
2.3 Розробка інформаційної технології інтелектуальної підтримки процесу ергономічного проєктування людино-машинних інтерфейсів.....	34
2.4 Розробка стратегії кодування слуховою модальністю	38
2.5 Висновки до другого розділу.....	41
3 Розробка архітектури програми.....	42
3.1 Моделювання програмного забезпечення	42
3.2 Розробка системи інтелектуальної підтримки	45
3.3 Висновки до третього розділу	59
4 Експлуатація, тестування та експериментальне дослідження програми.....	60
4.1 Приклад проєктування людино-машинного інтерфейсу «Система охолодження»	60
4.2 Висновки до четвертого розділу.....	65
5 Економіко-організаційна частина.....	67
5.1 Планування розробки програмного продукту.....	67
5.2 Визначення витрат на розробку програми	69
5.3 Розрахунок економічної ефективності програмного продукту	77
6 Охорона праці та безпека в надзвичайних ситуаціях.....	81

6.1 Аналіз потенційних небезпек	81
6.2 Заходи із забезпечення безпеки	82
6.3 Заходи з виробничої санітарії та гігієни праці	86
6.4 Заходи безпеки у надзвичайних ситуаціях	92
Висновки	97
Перелік джерел посилання	98
Додаток А Текст програми	102
Додаток Б Копії слайдів презентації	113

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

SCADA	– Supervisory Control And Data Acquisition;
БЗД	– база знань і даних;
ЕЗП	– ергономічне забезпечення проектування;
ІТ	– інформаційна технологія;
ЗВІ	– засоби відображення інформації;
ЛМІ	– людино-машинний інтерфейс;
ОКК	– об'єкти контролю й керування;
ОПР	– особа, що приймає рішення;
ПдО	– предметна область;
ПЗ	– програмне забезпечення;
ПК	– персональний комп'ютер;
СІП	– система інтелектуальної підтримки;
СЛТС	– система "людина-техніка-середовище";
ТП	– технологічний процес.

ВСТУП

В даний час при побудові систем автоматизованого управління технологічними процесами (ТП) інтерфейс користувача, з системою управління, реалізується за допомогою обчислювальних систем. Такий підхід обумовлений декількома причинами: компактністю (у фізичному та енергетичному сенсі) сучасної обчислювальної техніки, розвиненістю способів відображення інформації, великий функціональності і мінливості систем управління.

Застосування комп'ютерної техніки в АСУТП взагалі, і на робочих місцях операторів зокрема, привело до зародження класу програмного забезпечення (ПЗ), відомого як SCADA (Supervisory control and data acquisition - диспетчерське управління и збір даних). Отже, найважливішим завданням ПЗ SCADA є надання інтерфейсу взаємодії між оператором і системою управління ТП. Часто на SCADA покладають і такі завдання як: формування сигналізації про відхилення в ТП, ведення архівів параметрів ТП і протоколів подій.

Тому програмне забезпечення SCADA зручно розглядати як сукупність підсистем: бази даних параметрів ТП і зв'язку з системами управління ТП (контролерами), формування сигналізації про відхилення ведення ТП, архівування, протоколювання, візуалізації оперативних і архівних даних.

У доповнення, до перерахованих вище завдань можна віднести поділ прав доступу на читання-зміну тих чи інших параметрів ТП, реалізоване в підсистемі безпеки.

Таким чином сучасні SCADA-системи являють собою досить складні програмні комплекси.

Для розробки SCADA-систем використовують спеціалізований інструментарій у вигляді програмних систем для автоматизації технологічних процесів, таких як InTouch, Citect, TRACE MODE і ін. Системи даного класу використовують широкий спектр бібліотек спеціальних компонентів і

графічних редакторів. Однак вони не містять модулів або надбудов, що забезпечують інтелектуальну підтримку ергономічного проектування ЛМІ.

Виходячи з цього актуальним є розробка, дослідження та програмна реалізація ІТ інтелектуальної підтримки проєктувальників інтерфейсів SCADA-систем.

Значну роль у формуванні та розвитку методів та систем інтелектуальної підтримки прийняття рішень ергономічного проєктування інтерфейсів відіграють роботи зарубіжних і вітчизняних авторів Stuart J. Russell, Peter Norvig, В.Ф. Венди, А.І. Губінського, В.М. Муніпова, Д.О. Поспєлова, Е. В. Попова та ін. [1]-[6].

Відома також ІТ інтелектуальної підтримки процесу ергономічного проєктування ЛМІ, розроблених за допомогою SCADA-систем [7]. Однак в ній передбачена можливість використання тільки зорової модальності і відсутній формальний опис кодування технологічної інформації слуховою модальністю.

Об'єкт дослідження даної випускної кваліфікаційної роботи магістра – ергономічне забезпечення проєктування інтерфейсів SCADA-систем.

Предмет дослідження – ІТ інтелектуальної підтримки процесу ергономічного проєктування ЛМІ, що розробляються за допомогою SCADA-систем.

Мета роботи – підвищення рівня автоматизації та ефективності роботи проєктувальників інтерфейсів SCADA-систем шляхом розроблення і застосування програмного забезпечення на основі методів і моделей штучного інтелекту.

Для досягнення поставленої мети у роботі необхідно вирішити такі завдання:

- розглянути призначення та структуру SCADA-систем;
- дослідити існуючі технології побудови SCADA-систем;
- проаналізувати стан досліджень у області проєктування ЛМІ SCADA-систем;

- провести системний аналіз діяльності проєктувальника інтерфейсів та виявити номенклатуру проблемних ситуацій, у яких опиняється проєктувальник інтерфейсів SCADA-систем;
- розробити ІТ інтелектуальної підтримки проєктувальників ЛМІ SCADA-систем для двох типів модальностей – зорової та слухової;
- розробити стратегію кодування технологічної інформації слуховою модальністю;
- розробити моделі програмного забезпечення для автоматизації діяльності проєктувальників ЛМІ SCADA-систем;
- розробити програмне та інформаційне забезпечення компонентів системи, що інтелектуально підтримує проєктувальника;
- перевірка розробленого програмного забезпечення на прикладі проєктування фрагментів ЛМІ SCADA-системи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 SCADA-системи: загальні поняття і структура

Аналіз багатьох проєктів автоматизованих систем контролю і управління для великого спектру областей застосування дозволив виділити узагальнену схему їх реалізації, представлену на рисунку 1.1.

Як правило, це дворівневі системи, оскільки саме на цих рівнях реалізується безпосереднє управління технологічними процесами [8]. Специфіка кожної конкретної системи управління визначається використаною на кожному рівні програмно-апаратної платформою.

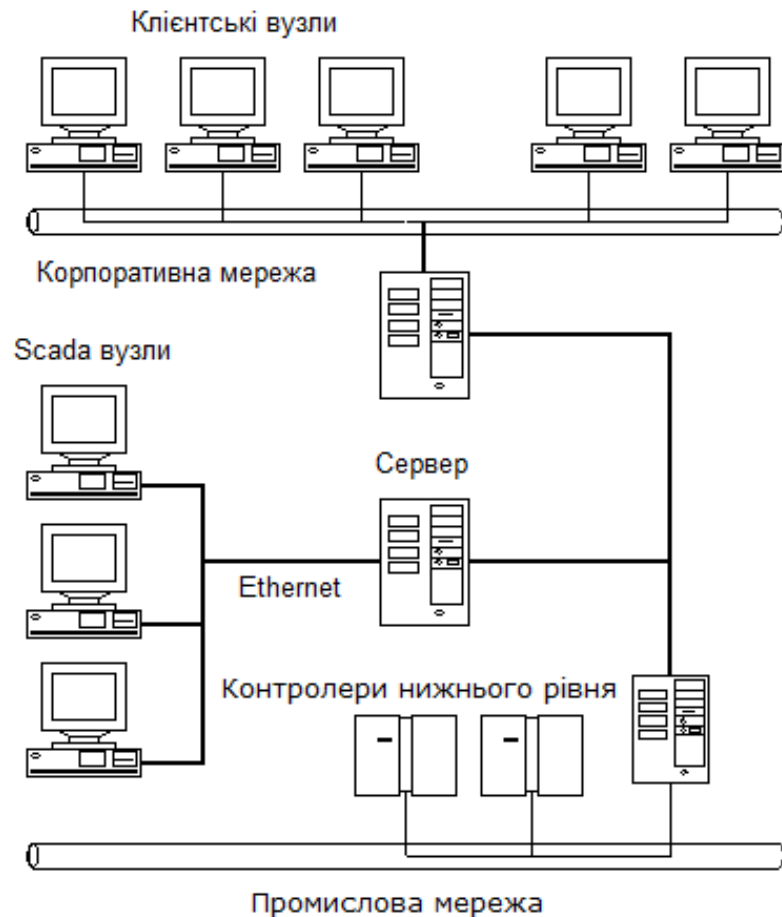


Рисунок 1.1 – Узагальнена схема реалізації сучасних АСУТП

Нижній рівень - рівень об'єкта (контролеру) - включає різні датчики для збору інформації про хід технологічного процесу, електроприводи і виконавчі

механізми для реалізації регулюючих і керуючих впливів. Датчики постачають інформацію локальним програмованим логічним контролерам (PLC - Programming Logical Controoller), які можуть виконувати такі функції:

- збір та обробка інформації про параметри технологічного процесу;
- управління електроприводами та іншими виконавчими механізмами;
- рішення завдань автоматичного логічного управління та ін.

Так як інформація в контролерах попередньо обробляється і частково використовується на місці, істотно знижуються вимоги до пропускну здатності каналів зв'язку.

До апаратно-програмних засобів нижнього рівня управління висуваються жорсткі вимоги щодо надійності, часу реакції на виконавчі пристрої, датчики і т.д. Програмовані логічні контролери повинні гарантовано відгукуватися на зовнішні події за час, визначений для кожної події. Для критичних з цієї точки зору об'єктів рекомендується використовувати контролери з операційними системами реального часу.

Розробка, налагодження і виконання програм управління локальними контролерами здійснюється за допомогою спеціалізованого програмного забезпечення, широко представленого на ринку. До цього класу інструментального ПЗ відносяться пакети типу ISaGRAF (CJ International France), InConrol (Wonderware, USA), Paradym 31 (Intellution, USA), що мають відкриту архітектуру.

Інформація з локальних контролерів може направлятися в мережу диспетчерського пункту безпосередньо, а також через контролери верхнього рівня. Залежно від поставленого завдання контролери верхнього рівня (концентратори, інтелектуальні або комунікаційні контролери) реалізують різні функції. Деякі з них перераховані нижче:

- збір даних з локальних контролерів;
- обробка даних, включаючи масштабування;
- підтримка єдиного часу в системі;
- синхронізація роботи підсистем;

- організація архівів за обраними параметрами;
- обмін інформацією між локальними контролерами і верхнім рівнем;
- робота в автономному режимі при порушеннях зв'язку з верхнім рівнем;
- резервування каналів передачі даних та ін.

Верхній рівень (диспетчерський пункт) – включає, перш за все, одну чи кілька станцій управління, що представляють собою автоматизоване робоче місце оператора. Тут же може бути розміщений сервер бази даних, робочі місця (комп'ютери) для фахівців і т. п. Часто в якості робочих станцій використовуються ПЕОМ типу IBM PC різних конфігурацій. Станції управління призначені для відображення ходу технологічного процесу та оперативного управління. Ці завдання і покликані вирішувати SCADA - системи.

Основна мета SCADA-систем - зробити інформацію про виробництво, технологічні процеси більш доступною для всіх підрозділів підприємства, забезпечити можливість роботи з цією інформацією в будь-який час і в будь-якому місці. Область застосування SCADA-систем може бути дуже широкою. Ось лише кілька прикладів ефективно розв'язуваних завдань [9]:

- контроль виробництва в реальному часі, доступ до технологічної інформації по необхідності з будь-якого відділу підприємства або віддаленого офісу без очікування звітів, можливість оперативного прийняття рішень не тільки на технологічному рівні, а й на рівні управління підприємством;
- інтеграція даних АСУТП з програмними системами управління підприємством, можливість створення "цифрової нервової системи" підприємства;
- віддалений моніторинг, необхідний в задачах диспетчеризації підприємств, систем життєзабезпечення будівель;
- дистанційна діагностика обладнання, оперативне оповіщення персоналу про збої та аваріях;

- поліпшення сервісу при обслуговуванні клієнтів шляхом надання інформації в реальному часі про проходження замовлення.

АСУТП в більшості випадків є системами організаційно-технічними, а це означає наявність функцій, необхідних для виконання людиною (оператором). Функції, які покладаються на будь-яку SCADA-систему, незалежно від того, є вона багатофункціональною або створена для конкретних потреб, наступні:

- прийом інформації від контролерів нижніх рівнів і датчиків;
- збереження прийнятої інформації в архівах для подальшого використання;
- вторинна обробка прийнятої інформації для формування зведених даних;
- графічне представлення ходу технологічного процесу, а також прийнятої і архівної інформації в зручній для сприйняття формі;
- прийом команд оператора і передача їх на адресу контролерів нижніх рівнів і виконавчих механізмів;
- оповіщення експлуатаційного та обслуговуючого персоналу про виявлені аварійні події, пов'язані з технологічним процесом і функціонуванням програмно-апаратних засобів АСУТП з реєстрацією дій персоналу в аварійних ситуаціях;
- формування зведень та інших звітних документів на основі архівної інформації;
- обмін інформацією з автоматизованою системою управління підприємством або з так званою, комплексною автоматизованою системою;
- безпосереднє автоматичне управління технологічним процесом у відповідності з заданими алгоритмами.

1.2 Аналіз технологій побудови SCADA-систем

1.2.1 Web технології в SCADA-системах

Важливою характеристикою SCADA-систем є незалежність від апаратної платформи і середовища передачі даних. Це може бути реалізоване використанням web технологій для візуалізації і оперативного впровадження технологічних схем, сигналізацій, трендів, звітів кінцевим користувачам. Внаслідок цього спрощується обробка та інтегрування інформації з різних джерел в рамках єдиної графічної складової з головними експлуатаційними показниками, забезпечується безпечний спільний доступ до інформації в географічно віддалених точках. Це дозволяє операторам контролювати умови експлуатації та функціонування усього обладнання за допомогою одного вікна web браузера (рис. 1.2) [10].



Рисунок 1.2 – Реалізація SCADA-системи за допомогою web технології

Переваги та потенційні скорочення витрат можливі завдяки самій структурі SCADA на мережевій основі. Вони можуть бути досить істотними, адже реальна мережева архітектура на основі сервера і клієнта слабо пов'язана з операційною системою і забезпечує зручний віддалений доступ і підвищену безпеку при запуску графічних і прикладних служб в мережі. Наприклад, для підтримки корпоративної політики безпеки, SCADA-системи та їх додатки можуть мати додатковий захист від неавторизованого використання та доступу. Прозорість, незалежність і гнучкість web технологій дозволяють легко адаптувати систему до найсучасніших технологій IT-безпеки.

Більше того, платформна незалежність подібного класу SCADA-систем є важливим фактором забезпечення достатньої гнучкості для роботи на комп'ютерах з різними ОС. Досягти цього можна, використовуючи платформу .NET або Java як основну програмну технологію, яка перетворює SCADA-клієнт в середу, доступну як незалежний додаток в web браузері на будь-якому мікропроцесорному пристрої.

Загальні характеристики SCADA-системи на мережевій основі:

- система проектується і створюється для роботи повністю в мережевому оточенні, що дозволяє мати доступ до всіх функціональних можливостей за наявності на рівні робочого місця лише засобів перегляду (браузера);

- система забезпечує більш високий рівень інтерактивності завдяки обміну малими обсягами даних з сервером “за кадром”. При цьому здійснюється обмін лише тією інформацією, яка є актуальною в даний момент часу для кожного користувача. Це досягається використанням технології Ajax (Asynchronous Javascript and XML) - підходу до побудови інтерактивних користувацьких інтерфейсів веб-додатків, що полягає в фоновому обміні даними браузера з веб-сервером. В результаті, при оновленні даних веб-сторінка не перезавантажується повністю, і веб-додатки стають швидше і зручніше.

Ще одним фактором зниження вартості при використанні Web-технологій є установка і обслуговування середовища візуалізації SCADA на віддалених клієнтах за структурою обміну даними системи. Така функціональна можливість називається “zero deployment”. Ця функція призначена для спрощення та удосконалення поширення додатків серед користувачів (клієнтів). Тобто користувачам не потрібно завантажувати додаток, оскільки він буде автоматично встановлений зі SCADA-сервера при першому звертанні до нього. При наступному звертанні клієнт отримує тільки оновлення даних і постійно зберігає актуальний стан у разі будь-яких змін додатків або конфігурації.

Запуски клієнтських додатків не створюють істотного додаткового навантаження на сервер, оскільки виконуються локально, з використанням обчислювальної потужності клієнтського ПК.

Основні переваги такого підходу:

- авторизовані користувачі можуть автоматично отримувати клієнтські програми за допомогою будь-якого web браузерa;
- користувач завжди отримує найновіші версії додатка;
- немає необхідності управління ліцензіями і установкою ПЗ з клієнтської сторони;
- відсутність файлів для копіювання і необхідності установки конфігурування підключення;
- більш ефективний розподіл навантаження між сервером і клієнтами;
- можливість використання затверджених заходів IT-безпеки та безпечних мережевих підключень.

Вбудовані засоби управління політикою безпеки повинні враховувати роль кожного користувача, характеристики пристрою і стан мережі для визначення того, до яких програм та даних їм дозволено доступ. Це має такі переваги:

- безпечне отримання додатків;
- вища продуктивність IT;

- відсутність неавторизованого доступу;
- якісне шифрування даних.

Таким чином, технології візуалізації на мережевій основі надають кінцевому користувачу економічно ефективну експлуатацію і цілісність інформації всередині всього підприємства, не піддаючи ризику безпеку і технологічну ефективність системи.

1.2.2 Інструментальні засоби опису інформаційних моделей

Автоматизація технологічних процесів привела до того, що основним джерелом інформації про об'єкти контролю й керування (ОКК) для оператора стали використовуватися засоби відображення інформації (ЗВІ). ЗВІ (монітори персональних комп'ютерів (ПК), прилади, екрани, мнемосхеми, табло й т.ін.) призначені для пред'явлення даних оператору, що характеризують ОКК, хід технологічного процесу, енергетичні ресурси, стан каналів зв'язку та ін. Ці дані пред'являються оператору в кількісній і якісній формі, причому він взаємодіє не з реальними ОКК, а з їхніми інформаційними моделями [11].

Згідно [3] "ІМ є організована за певними правилами сукупність інформації про стан і функціонування об'єкта керування й зовнішнього середовища. Вона є для оператора своєрідним імітатором істотно важливих для керування властивостей, реальних об'єктів, тобто тим джерелом інформації, на основі якого він формує образ реальної обстановки, робить аналіз і оцінку ситуації, що сформувалась, планує керуючі впливи, приймає рішення, що забезпечують ефективну роботу системи, а також оцінює результати їхньої реалізації".

Для сучасних АСУ ТП характерно те, що оператор часто віддалений від об'єктів контролю і керування і здійснює управління дистанційно за допомогою ІМ. Існує багато типів ІМ. Окремим випадком ІМ є людино-машинний інтерфейс оператора сучасних АСУ ТП, що проєктується засобами спеціалізованого ПЗ SCADA-систем.

Проектування ІМ, тобто ЛМІ, є одною з найважливіших задач ергономіки.

Різні інструментальні SCADA системи, розроблені крупними фірмами, такими як Intellution, WonderWare, FactoryLink, AdAstra та ін. мають власні засоби створення і відображення інформаційних моделей. При цьому формати їх зберігання, як правило, не описані і є закритими. Інформаційна модель, створена в одному середовищі розробки, не може бути використана в інший. Частково така ситуація зумовлена значним рівнем конкуренції на ринку SCADA, а частково відсутністю відповідних стандартів.

Необхідність чіткої специфікації опису інформаційних моделей встала в зв'язку з розширенням технологічної бази розробки SCADA-систем в область Інтернет. Інформаційні моделі, розроблені в середовищі спеціалізованого графічного редактора, слід було відобразити за допомогою нових засобів. Для цього необхідно розробити специфікацію представлення інформаційних моделей, зрозумілу різним застосункам, написаним на різних мовах програмування і в різних операційних системах.

Опис мнемосхеми може містити графічне зображення, звукові сигнали та сценарій поведінки в разі тих чи інших подій. Події можуть виникати або в результаті інтерактивних дій користувача, або в результаті отримання оперативних даних. Графічна складова інформаційних моделей може містити растрові, векторні зображення та елементи інтерфейсу (кнопки, графіки, текстові поля тощо). При цьому кожен елемент має велику кількість параметрів, що потребують зберігання для подальшої візуалізації.

Одним з можливих варіантів опису інформаційних моделей є використання XML (Extensible Markup Language) - це мова розмітки, що описує ієрархію об'єктів даних. XML дозволяє здійснювати контроль за коректністю даних, що зберігаються в документах, робити перевірки ієрархічних співвідношень усередині документа і встановлювати єдиний стандарт на структуру документів, вмістом яких можуть бути самі різні дані. Це означає, що його можна використовувати при побудові складних інформаційних систем, у

яких дуже важливим є питання обміну інформацією між різними додатками, що працюють в одній системі. XML є W3C стандартом і набув широкої популярності в області передачі та зберігання структурованої інформації.

На базі XML реалізована велика кількість форматів представлення графічних даних. Одним з них є стандарт для двовимірних векторних зображень SVG. Стандарт SVG активно розвивається за участю провідних розробників програмного забезпечення (Microsoft, Adobe, Corel та ін.). Наочність структури XML, вбудовані Java-скрипти, правила SSL є значним кроком вперед у порівнянні з графічними мета файлами, однак проблема прив'язки оперативних даних з конфігураційної бази даних за допомогою стандартних засобів редагування залишається не вирішеною.

У SVG обробка подій покладена на JavaScript, події орієнтовані на інтерактивні дії користувача і реалізовані за аналогією з оброблювачами подій в HTML. Цей шлях має два суттєвих недоліки в плані застосовності його для опису мнемосхем. Перше - скрипти повільна інтерпретована мова. Друге - від розробника інформаційних моделей необхідне вміння програмувати на мові високого рівня.

Найближчими конкурентами SVG можна назвати формати VML і FLASH. Розроблений в Microsoft формат VML (Vector Markup Language) також заснований на XML і близький до SVG по концепції, можливостям і навіть за назвами тегів. Цей формат в 1998 році був представлений консорціуму W3C для стандартизації, але був відкинутий на користь SVG. Формат FLASH переважно використовується для створення невеликих анімованих роликів. Це пояснюється орієнтацією формату FLASH на публікацію в Інтернеті анімацій і наявністю розвинених інструментальних засобів створення FLASH фільмів.

Отже, на даний момент не існує стандартного формату опису інформаційних моделей. Кожен з розробників SCADA-систем використовує власні розробки або вже існуючі стандарти передачі структурованих даних, найпоширенішим з яких є XML

1.3 Постановка задач дослідження

Аналіз сучасних SCADA-систем виявив, що жодна з них не має в своєму складі модулів інтелектуальної підтримки процесу ергономічного проєктування ЛМІ. Це призводить до створення неякісних, з точки зору ергономіки, інформаційних моделей - без урахування психофізіологічних характеристик людини-оператора, завдань, що стоять перед ним, характеристик середовища тощо. Про актуальність даної проблеми свідчить постійне збільшення впливу людського фактору (з 20% до 80% за останні пів століття) на аварійність технологічного процесу.

Для досягнення мети роботи необхідно вирішити певні задачі дослідження, що були вже наведені у вступній частині (с. 10 та с.11). Розкриємо більш детально перелік задач, що необхідно вирішити при розробці програмного та інформаційного забезпечення компонентів системи, що інтелектуально підтримує проєктувальника:

- формалізація знань предметної області “Ергономічне проєктування інформаційних моделей” (розробка керуючих стратегій і правил);
- реалізація інтерфейсу для проведення опитування експерта в конкретній предметній області “Технологічний процес і його об’єкти контролю та управління”;
- інтелектуальна підтримка проєктувальника ЛМІ (аналіз даних, отриманих в результаті опитування експерта і формування рекомендацій на всіх етапах проєктування ЛМІ (інформаційної моделі);
- для забезпечення в реальному часі збору, обробки, відображення та архівування інформації про об’єкти контролю та управління необхідно розробити модуль візуалізації. Найбільш перспективним при цьому є використання технології web-додатків, що забезпечить зручний віддалений доступ та підвищену безпеку при запуску графічних і прикладних служб в мережі.

1.4 Висновки до першого розділу

Аналіз проєктів автоматизованих систем контролю і управління показав, що управління технологічними процесами здійснюється, як правило, із застосуванням дворівневих систем. На кожному рівні може бути людина-оператор.

Проведений аналіз технологій побудови SCADA-систем показав, що найбільш ефективною (з точки зору безпеки та технологічної ефективності системи) є реалізація на основі web технологій.

У складі відомих інструментальних засобів розробки ЛМІ SCADA-систем відсутні модулі або підсистеми інтелектуальної підтримки ергономічного проєктування. Це є показником актуальності розробки СІП для проєктувальника ЛМІ SCADA-систем.

Для досягнення поставленої мети роботи було сформульовано постановку певних задач дослідження.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Аналіз стану досліджень у області проєктування людино-машинних інтерфейсів SCADA-систем

Об'єктом дослідження в даній роботі є ергономічне забезпечення проєктування інтерфейсів SCADA-систем. Для подальшого розгляду будемо виходити з того, що технологічна інформація повинна відображатися оператору засобами відображення інформації індивідуального користування (тобто відображаються на моніторах ПК).

Предметом дослідження є ІТ інтелектуальної підтримки процесу ергономічного проєктування ЛМІ, що розробляються за допомогою SCADA-систем.

Вимоги до інформаційного, технічного та програмного забезпечення SCADA-систем постійно змінюються під впливом ринку. З'явився новий напрямок функціонального розвитку SCADA-систем, заснований на впровадженні інтелектуальних інформаційних технологій, що дозволяє істотно розширити область використання цих систем.

Для виявлення актуальності задач досліджень було використано [11] метод [12]. Суть методу полягає в наступному: на основі робіт провідних спеціалістів виділяються об'єкти й предмети дослідження в області "Ергономіка"; потім відомі наукові результати (публікації) співвідносяться з "ергономічними нішами" і виявляються актуальні завдання, що вимагають розробки. В [12] наведене зведення досліджень в області ергономіки та суміжних областях (містить 50 авторів). Предмети досліджень наукового напрямку "ергономіка" автори класифікують у двох аспектах: з позиції ергономічних вимог (А), загальна номенклатура й основні характеристики яких установлені ГОСТ 20.39.108-85 і частково в [13], і з позицій етапів ергономічного забезпечення (В). В аспекті В розрізняють наступні класи:

- методологія (шифр 1);
- встановлення ергономічних вимог (шифр 2);

- оцінювання й оптимізація параметрів (шифр 3);
- ергономічна експертиза (шифр 4);
- моделювання (шифр 5);
- автоматизація (шифр 6);
- ефект ергономічного забезпечення (економіка) (шифр 7).

Предмет дослідження даної роботи належить (з позиції А) до класу "Організація діяльності оператора" (має в [12] шифр 2), підкласу "Інформаційна модель" (шифр 2.3). З позиції В – до класу "Автоматизація" (шифр 6).

Згідно з [12] предмет досліджень по напрямку із шифром 2.3 є присутнім у трьох авторів: Венда В.Ф. [2]; Галактіонов А.І. [14] і Рібаков Ф.І. [15]. Однак з позиції В дані роботи належать: Венда В.Ф. - до класів із шифрами 1, 3 і 5; Галактіонов А.І. - до класів із шифрами 1-5, а Рібаков Ф.І. - класу із шифром 3.

Відомо застосування інтелектуальних систем як компонент SCADA-систем, або спільно з ними, наприклад [16]. Така інтеграція забезпечує інтелектуальну підтримку оператора в процесі аналізу даних, діагностики стану технологічного обладнання, прогнозування розвитку ситуацій і прийняття відповідних рішень в штатних і позаштатних ситуаціях.

Незважаючи на наявність в SCADA-системах великого арсеналу бібліотек спеціалізованих елементів і графічних редакторів, в їх складі відсутні засоби інтелектуальної підтримки ергономічного проєктування ЛМІ. Іншими словами, SCADA-системи надають проєктувальнику тільки конструктор, з "деталей" якого він створює кінцевий продукт праці у вигляді ЛМІ, але інформаційно не підтримують методологію ЕЗП інтерфейсу.

Існує ряд відомих методик ЕЗП інформаційних моделей операторів АСУ ТП, наприклад [4]. Їх недоліком є слабка формалізація етапів проєктування. Проведений аналіз [11], а також подальші дослідження показали наступне [7]:

- ведуться розробки програмного забезпечення для відображення в Інтернет оперативної диспетчерської інформації, а також архівних даних [17]. Була розроблена мова опису мнемосхем на основі стандарту XML. Файл опису мнемосхеми містить інформацію про зовнішній вигляд мнемосхеми і способах

реакції на різні події або інтерактивні дії користувача. Недоліком даного підходу є відмова від представлення мнемосхем у форматі SVG, в результаті чого втрачається можливість малювати мнемосхеми за допомогою графічних редакторів, розроблених такими фірмами як COREL DRAW, VISIO тощо;

- були сформульовані та досліджені [16] основні вимоги до організації ЛМІ операторів-технологів АЕС з реакторами ВВЕР-1000 та визначено обмежений, достатній для організації контролю, управління та діагностики енергоблоку набір елементів ЛМІ;

- ведуться розробки програмного забезпечення системи підвищення якості інтерфейсу комп'ютерних тренажерних систем [18]. На основі аналізу факторів, що найбільш впливають на якість НМІ комп'ютерних тренажерних систем на мнемосхемах, був розроблений новий метод, що автоматизує процес оцінки якості НМІ;

- запропоновано інформаційну технологію інтелектуальної підтримки процесу ергономічного проектування ЛМІ, що розробляються за допомогою SCADA-систем . Проведено аналіз структури ЛМІ та виявлено властивості його типових елементів, які можуть бути представлені у вигляді відомих категорій кодування інформаційних моделей. Визначено номенклатура класів MSDN, необхідних для програмної реалізації кодів. Отримані результати є теоретичною основою для розробки експертної системи, що здійснює інтелектуальну підтримку проектувальника ЛМІ [7];

- в даний час є вітчизняні стандарти, що регламентують вимоги до ергономічного проектування таких типів ІМ як мнемосхеми [19]. Найбільш близькими до проблеми, що розглядається є міжнародний стандарт ANSI/ISA-101.01-2015. Цей стандарт стосується ЛМІ для обладнання і автоматизованих процесів. Якщо дотримуються стандартні, рекомендуємі практики та методологія, результат повинен дозволити користувачам бути більш ефективними, забезпечуючи підвищену безпеку, якість, продуктивність і надійність. У стандартах ISO-9241, ISO-10075, ISO-13407, ISO-14915 надано рекомендації щодо ЕЗП взаємодії «людина-система», ментальної

завантаженості оператора та розробка орієнтованого на користувача ЛМІ відповідно.

На підставі вищевикладеного, можна зробити висновок, що найбільш близьким за призначенням аналогом є ІТ інтелектуальної підтримки процесу ергономічного проєктування ЛМІ [7]. Недоліком даної технології є те, що в ній не передбачена можливість застосування слухової модальності для кодування технологічної інформації в ЛМІ.

Слід також зазначити, що завдання проєктування і налаштування ЛМІ можуть значно збільшити вартість і складність проєкту АСУ ТП. Згідно [20], проєктувальники витрачають 10 000 доларів США або більше на розробку кожної сторінки графіки ЛМІ для своїх виробничих операцій. Тому, актуальним завданням є розробка методу інтелектуальної підтримки діяльності проєктувальників ЛМІ АСУТП.

Для виявлення номенклатури проблемних ситуацій *PS*, у яких опиняється проєктувальник інтерфейсів проведемо системний аналіз його діяльності.

2.2 Системний аналіз діяльності проєктувальника інтерфейсів

Системний аналіз діяльності проєктувальника ЛМІ складається з компонентного, морфологічного й еволюційного аналізів [11]. Метою компонентного аналізу є виявлення всіх сутностей (об'єктів, функцій, станів), що складають вміст даної системи "людина-техніка-середовище" СЛТС(*N*). Для цього побудований комплекс наступних компонентних моделей: компонентно-системних *KSS(N)*; компонентно-елементних *KES(N)* і компонентно-функціональних *KFS(N)*.

KSS(N) являє собою семантичний граф у вигляді ієрархічного дерева, вершинами якого є імена систем різного масштабу, а дугами - відношення «ціле (*G*) - частина (*P*)» R_{gp} і відношення «впливати» R_F . Дана модель уявляє собою знання про навколишнє середовище *EES(N)* і про внутрішній (підсистемний)

склад системи, що досліджується – $ETS(N)$. Знання $EES(N)$ включають: ім'я метасистеми ($EES-0$), у яку частиною входить система, що досліджується; множина систем, що збурюють зовнішнє середовище ($EES-1$), які організаційно не пов'язані з $ETS(N)$, але впливають на якість її функціонування. ЦентRALЬНОЮ вершиною графа (коренем дерева) є метасистема $EES-0$. Кінцеві вершини графа – локальні підсистеми ($LETS-N$), у яких діє тільки одна особа, що приймає рішення (ОПР - проєктувальник інтерфейсів). СЛТС, що досліджується зв'язана з метасистемою відношенням R_{GP} , а із збурювальними системами, - відношенням R_F .

У даному випадку СЛТС, що досліджується - $ETS(N)$, є система "проєктувальник ЛМІ – СІП ЕЗП проєктувальника ЛМІ (далі СІП ЕЗП) - середовище", метасистемою $EES-0$ – організація в якій працює проєктувальник, збурювальною системою $EES-1$ – умови й режими праці й відпочинку проєктувальника, його емоційна напруженість і вказівки вищих організацій, а локальною $LETS-N$ – "конкретний n -й проєктувальник ЛМІ і його робоче місце" [11].

Для всіх локальних підсистем $LETS-N$ побудована множина $KES(N)$ за формулою (2.1):

$$KES(N) = \langle KTK, KDK-N \rangle, \quad (2.1)$$

де KTK – знання про типи компонентів будь-якої $LETS-N$;

$KDK-N$ – знання про конкретні компоненти, що утворюють дану локальну підсистему $LETS-N$ за формулою (2.2):

$$KTK = \langle ST, R_{GP}^{ST*ST} \rangle, \quad (2.2)$$

де $ST = \{LHT - N.X.Y, TK\}$ - множина вершин типової структури;

$TK = \{T_i\} = \{TH, TM, TO, TE_1, TE_2, TE_3, TE_4, TE_5, TE_6\}$ - множина типів компонентів внутрішньої структури організації, що наведена на рис. 2.1;

$R_{GP}^{ST \times ST}$ – відношення “ціле-частина”, задане на декартовому добутку $ST \times ST$ відповідно до рис. 2.1.

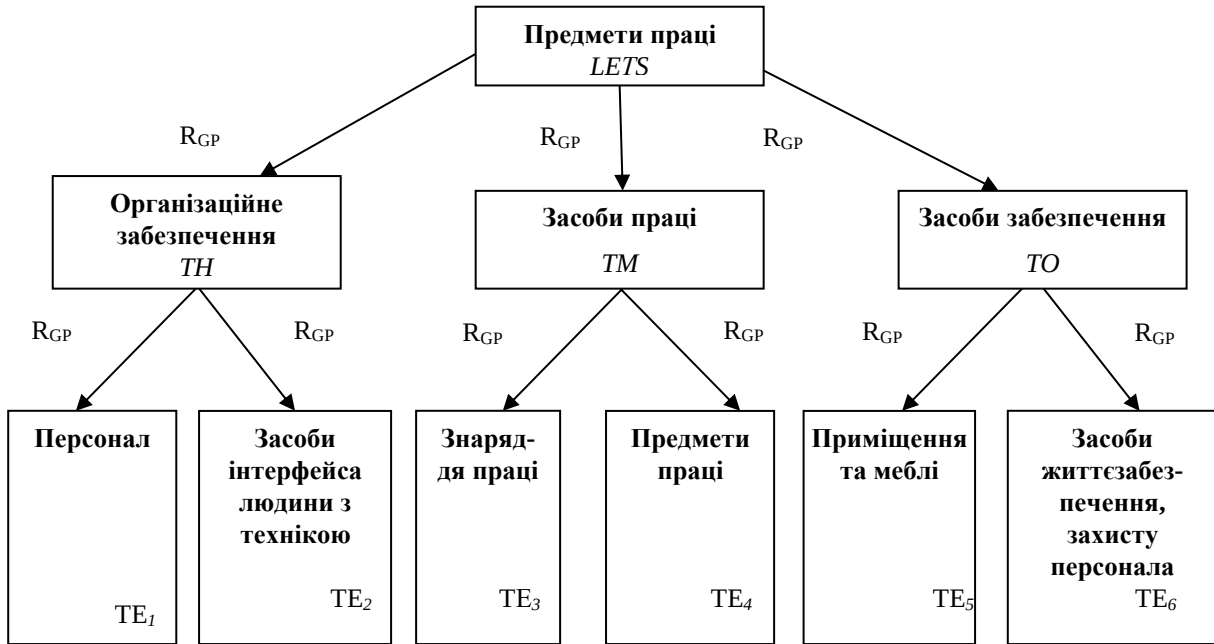


Рисунок 2.1 – Компонентно-елементна структура локальної підсистеми

Елементами $ETS(N)$ є: ергатичні елементи (фахівці, що працюють у складі системи $ETS(N)$) – користувач СІП ЕЗП (проектувальник ЛМІ), і група фахівців, що забезпечує розробку, функціонування, працездатність і розвиток СІП ЕЗП; неергатичний елемент – інформаційно-програмний комплекс у вигляді СІП ЕЗП. Вміст бази знань і даних (БЗД) про конкретні елементи $ETS(N)$, яка представлена у вигляді $KES(N)$, наступний:

- TE_1 : проектувальник ЛМІ, інженер по знаннях, експерти в предметних областях (ПдО) "ЕЗП ЛМІ" і ПдО "Технологічні процеси й ОКК", програміст, інженер-електронник, адміністратор БЗД;
- TE_2 : клавіатура, екран дисплея ПК, принтер;
- TE_3 : інформаційно-програмний комплекс (СІП ЕЗП), папір, ручка;

- TE_4 : предмет праці - база знань і даних СІП ЕЗП, норми, вимоги й принципи розробки СІП, ЛМІ і придбання знань; продукт праці – інформація, що перетворена у вигляді рекомендацій проєктувальнику, локальні БЗД;

- TE_5 : стіл письмовий, робоче крісло;

- TE_6 : освітлення, вентиляція (кондиціювання).

Структура $KFS(N)$ являє собою семантичний граф у вигляді ієрархічного дерева, вершинами якого є імена функціональних одиниць (F_i) різного масштабу, а дуги відображають відношення $R_{F_i F_j}$ - “для реалізації функціональної одиниці F_i (мети, задачі) необхідно виконати функціональну одиницю F_j (задачу, процедуру, функцію)”. Корінь дерева – поняття “Процес функціонування $ETS(N)$ ”. Перший рівень вершин дерева – назва цілей - T_{Ri} , для яких створена система N , що досліджується. Другий рівень – назва додаткових цілей T_{DRij} , які треба вирішити для досягнення цілей T_{Ri} . Третій рівень – назва задач T_{Sijk} , які треба вирішити для досягнення додаткових цілей. Четвертий рівень – назва процедур T_{PRijkl} , які необхідно виконати для рішення задачі T_{Sijk} .

Компонентно-функціональна структура для $ETS(N)$ представлена в табл. 2.1 у вигляді чотирирівневого списку, еквівалентного згаданому вище графу. Дана структура розроблена з врахуванням того, що з точки зору архітектури в СІП ЕЗП доцільно представляти наступні знання [6]:

- інтерпретовні (знання про представлення знань у внутрішньому форматі, предметні знання й керуючі знання);

- неінтерпретовні (допоміжні знання, що підтримні знання).

Морфологічні моделі СЛТС відображають різного роду взаємозв'язки (організаційні, технологічні, часові й ін.), що існують між системами й компонентами, виділеними в результаті компонентного аналізу. Їхня номенклатура визначається цілями конкретного дослідження. Практично при всяких видах досліджень виникає необхідність мати функціонально-часові структури (FTS), а в багатьох випадках – суб'єктно-організаційні (SOS) і функціонально-організаційні (FOS) [11].

Таблиця 2.1 – Компонентно-функціональна структура *ETS(N)*

Ціль	Додат- кові цілі	Зада- чі	Проце- дури	Назва
1	2	3	4	5
1				Забезпечення інтелектуальної підтримки проєктувальника ЛМІ.
	1.1			Системний аналіз діяльності проєктувальника ЛМІ.
		1.1.1		Компонентний аналіз.
			1.1.1.1	Розробка компонентно-системних моделей.
			1.1.1.2	Розробка компонентно-елементних моделей.
			1.1.1.3	Розробка компонентно-функціональних моделей.
		1.1.2		Морфологічний аналіз.
			1.1.2.1	Розробка функціонально-часових моделей.
			1.1.2.2	Розробка суб'єктно-організаційних моделей.
			1.1.2.3	Розробка функціонально-організаційних моделей.
		1.1.3		Еволюційний аналіз (у випадку зміни в часі компонентних або морфологічних структур).
	1.2			Визначення номенклатури проблемних ситуацій у діяльності проєктувальника ЛМІ.
		1.2.1		Визначення номенклатури інтелектуальних задач.
			1.2.1.1	Аналіз стовпця 3 табл. 1
		1.2.2		Визначення номенклатури інтелектуальних процедур.
			1.2.2.1	Аналіз стовпця 4 табл. 1
		1.2.3		Визначення номенклатури інтелектуальних функцій.
			1.2.3.1	Розробка функцій для кожної процедури та їх аналіз.
	1.3			Придбання знань для СІП ЕЗП.
		1.3.1		Придбання предметних знань.
			1.3.1.1	Видобування знань із експертів у конкретній ПдО, про об'єкти, що будуть відображатися в ЛМІ.

Продовження таблиці 2.1

1	2	3	4	5
			1.3.1.2	Видобування знань із експертів у ПдО "ЕЗП ЛМІ" для кожної проблемної ситуації, що була виявлена в п.1.2.
			1.3.1.3	Формалізація експертних знань і уведення їх у базу знань і даних СІП ЕЗП.
		1.3.2		Розробка знань про представлення.
			1.3.2.1	Розробка організації знань про представлення.
			1.3.2.2	Розробка вмісту знань про представлення.
		1.3.3		Розробка керуючих знань.
			1.3.3.1	Розробка фокусувальних знань.
			1.3.3.2	Розробка вирішальних знань.
		1.3.4		Розробка допоміжних знань
			1.3.4.1	Розробка знань про мову.
			1.3.4.2	Розробка знань про діалог.
		1.3.5		Розробка підтримуючих знань.
			1.3.5.1	Розробка технологічних знань.
			1.3.5.2	Розробка семантичних знань.
	1.4			Автоматизація ергономічного проектування ЛМІ.
		1.4.1		Розробка загальної композиції ЛМІ.
			1.4.1.1	Визначення загального числа зон.
			1.4.1.2	Вибір місця розташування зони.
			1.4.1.3	Визначення обсягу інформації в зоні.
			1.4.1.4	Виділення зони.
		1.4.2		Кодування об'єктів у зоні (за даними п.1.3.1.1).
			1.4.2.1	Вибір системи й способів кодування об'єктів у ЛМІ: Вибір типу модальності сигналу; Вибір виду алфавіту; Вибір мірності коду; Вибір міри абстрактності коду; Вибір оптимального коду.
			1.4.2.2	Розробка компоновання кодового знака й групи (формуляра).
		1.4.3		Ергономічна оцінка результатів проектування ЛМІ.

Продовження таблиці 2.1

1	2	3	4	5
			1.4.3.1	Розробка правил ергономічної оцінки результатів проектування ЛМІ.
			1.4.3.2	Розробка рекомендацій з оптимізації рішень, прийнятих у п.п. 1.4.1 і 1.4.2
			1.4.3.3	Корегування результатів проектування ЛМІ.

Оскільки структури *SOS* відображають тільки відношення R_0 : «суб'єкт S_i , є керівником суб'єкта S_j », то виходячи із задач даного дослідження, необхідності в їх побудові не було.

Структури *FTS* – функціональні мережі (графи), вершинами яких є функціональні сутності, а дуги відповідають елементам відношення R_r : «функція F_i , безпосередньо передуює функції F_j ».

Структури *FOS* являють собою набір графів, число яких дорівнює числу функцій, що досліджується. Кожний граф складається з вершин трьох типів: перший – ім'я функції F_i ; другий – узагальнене ім'я комплексу компонентів K_i , що реалізують виконання цієї функції; третій – імена елементів, що входять у цей комплекс: активний елемент (людина) K_{i1} , предмети праці K_{i2} , знаряддя праці K_{i3} . Дуги відображають відношення: між F_i і K_{ij} – R_k : «функція F_i виконується комплексом компонентів K »; між K_i і K_j – R_{ij} : «у комплексі K_i виконуються функції j : $j=1$ – активного елемента, $j=2$ – предмета праці, $j=3$ – знаряддя праці». Перераховані вище компоненти беруться із числа компонентів, що були отримані при компонентно-елементному аналізі (імена елементів K_{i1} – зі списків TE_1 або TE_3 , K_{i2} – зі списків TE_4 ; K_{i3} – зі списків TE_2 , TE_5 , TE_6).

Для *ETS(N)* на підставі даних табл. 2.1 було побудовано: *FTS(N)* – алгоритмічну функціональну мережу процедур (вершинами графа є імена процедур) і *FOS(N)* – на підставі знань із *KES(N)*.

Еволюційні структури будуються тільки в тому випадку, коли важливо відзначити можливі зміни компонентних або морфологічних структур, наприклад, відобразити состав фахівців, що працюють у СЛТС у ранкові,

денну й вечірню зміни. Виходячи із задач даного дослідження, необхідності вводити еволюційні структури не було.

Сукупність побудованих компонентних (KSS , KFS , KES) і морфологічних структур (FTS , FOS) створили базу інтегральної моделі формального опису СЛТС.

На підставі даних структури $KFS(N)$ проведена декомпозиція процедур, на ергатичні (що виконуються проєктувальником ЛМІ разом з технікою) і автоматичні, що виконуються без втручання людини. До подальшого аналізу прийняті тільки ергатичні інтелектуальні процедури, що виконуються ОПР у процесі проєктування ЛМІ.

У відповідність із даними табл. 2.1 проєктувальник ЛМІ повинен виконувати процедури 1.4.1.1-1.4.2.2 і 1.4.3.1-1.4.3.3. Для $ETS(N)$ всі процедури, з даного переліку є ергатичними й інтелектуальними.

Таким чином, отримана номенклатура інтелектуальних задач T_{Sijk} і процедур T_{PRijkl} , що характеризують проблемні ситуації PS_i , які можуть виникнути в процесі діяльності проєктувальника ЛМІ.

2.3 Розробка інформаційної технології інтелектуальної підтримки процесу ергономічного проєктування людино-машинних інтерфейсів

Методологія проєктування ЛМІ SCADA-систем повинна базуватися на методах системного підходу. Це означає, що як об'єкт автоматизації "проєктування ЛМІ", так і знаряддя праці проєктувальника "СПП ЕЗП", повинні розглядатися окремо, а також спільно, як єдина людино-машинна система.

Проведений в п.2.2 системний аналіз даної системи дозволив визначити: компоненти системи і відносини між ними; функціональне призначення СПП ЕЗП; перелік проблемних ситуацій в діяльності проєктувальника ЛМІ, що вимагають інтелектуальної підтримки, а також номенклатуру типових елементів інтерфейсу, які він використовує в процесі проєктування.

праці проєктувальника - ЛМІ SCADA-системи ("ЛМІ"). Клас "WorkSpace" є контейнером для наступних класів: MainWindow (головне вікно інструментального засобу розробки ЛМІ); PictureBox (клас з простору імен System.Windows.Forms, що надає елемент управління графічним вікном Windows для відображення рисунка); StatusBar (панель стану, клас з простору імен System.Windows.Forms); ToolBar (клас з простору імен System.Windows.Controls, панель інструментів, що є контейнером для групи команд або елементів управління); TabControl (клас з простору імен System.Windows.Forms, панель вкладок, що управляє пов'язаним набором сторінок вкладок).

Інтелектуальна підтримка проєктувальника здійснюється за допомогою СП. Типова СП складається з наступних основних компонентів [6]: вирішувача (інтерпретатора); робочої пам'яті (бази даних), призначеної для зберігання вихідних і проміжних даних задачі, що розв'язується у поточний момент; бази знань (БЗ); компонентів придбання знань; пояснювального компонента; діалогового компонента.

Для їх відображення в концептуальну модель (рис. 2.2) введено такі класи:

- "Інтерпретатор" - для вирішувача;
- "Знання" - для бази знань і даних;
- "ІнтелектІнтерфейс" (підсистема "Інтелектуальний інтерфейс") - для інших компонент.

В результаті інтерактивної взаємодії проєктувальника з підсистемою "Інтелектуальний інтерфейс" здійснюється його інтелектуальна підтримка у вигляді порад, що генеруються СП ЕЗП.

Процес вироблення поради полягає в наступному. На вхід підсистеми "Інтелектуальний інтерфейс" надходить інформація: про поточний стан проєкту (від "Поточний проєкт"); про обраний проєктувальником типовий елемент інтерфейсу (від "Палітра"), а також результати діалогу з проєктувальником (від "Проєктувальник").

Результати обробки вхідної інформації дана підсистема передає "Інтерпретатору" у вигляді деяких вихідних суджень (передумов), які описують поточну проблемну проєктну задачу TS_i . "Інтерпретатор", використовуючи вихідні дані і знання БЗД СІП ЕЗП ("Знання"), формує послідовність правил (поради проєктувальнику), які призводять до вирішення задачі TS_i .

База знань та даних СІП ЕЗП складається з множини локальних БЗД. На рис. 2.2 представлені чотири локальні БЗД "ЕргВимоги", "Алфавіт" "ЗоровеКод" та "СлуховеКод". Перші три розглянуто в [7]. Локальна база "СлуховеКод" представляє знання про способи кодування (алфавіт) інформації про ОКК технологічного процесу слуховою модальністю: алфавітами "Звукові сигнали" та "Мовні сигнали".

Для графічного оформлення та компоновки елементів ЛМІ у СІП ЕЗП необхідно реалізувати спеціальний графічний редактор, що концептуально базується на моделі класу "Палітра", що є елементом концептуальної моделі предметної області. Ця модель наведена на рис. 2.3.

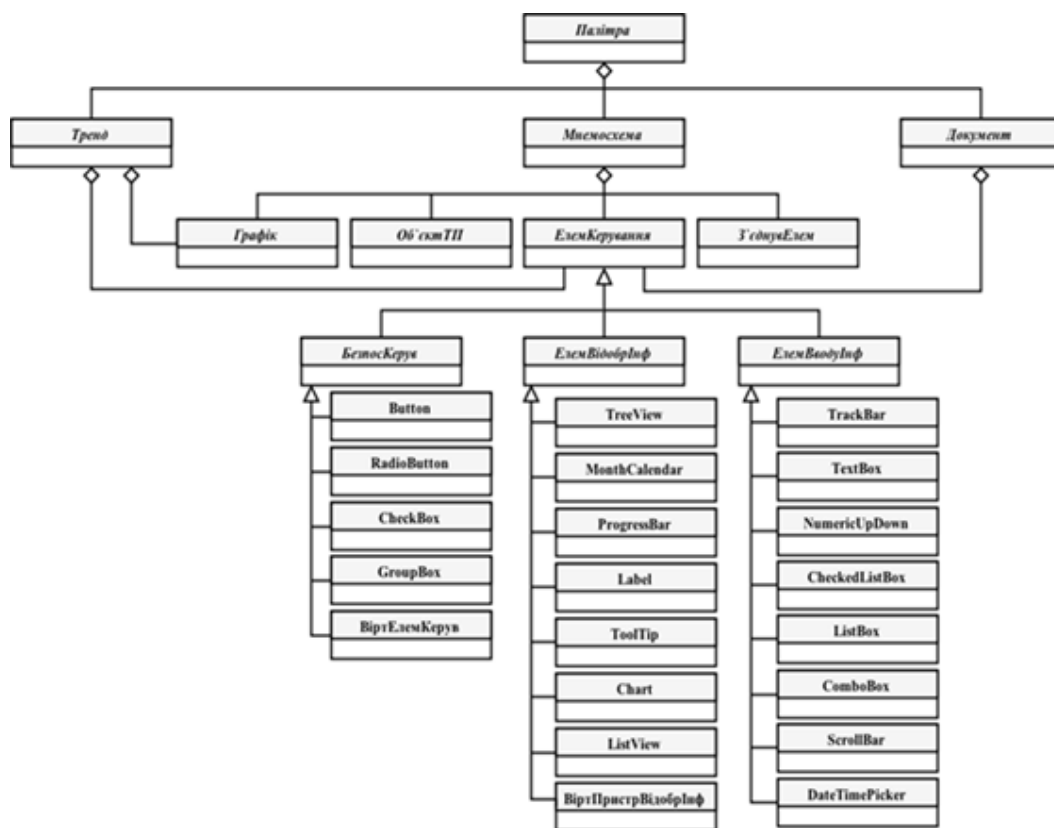


Рисунок 2.3 – Концепткальна модель предметної області "Палітра"

У відповідність з моделлю, наприклад, в зоні формату мнемосхеми можуть розміщуватися наступні елементи: "Графік"; "Об'єкт ТП"; "ЕлемКерування" (елементи керування ЛМІ) та "З'єднувЕлем" (з'єднувальні елементи на мнемосхеми).

Клас "ЕлемКерування" є батьківським для наступних класів: "БезпосКерув" (елементи безпосереднього керування); "ЕлемВідобрІнф" (елементи відображення інформації) та "ЕлемВводуІнф" (елементи вводу інформації). Кожен з цих класів, у свою чергу, має класи-нащадки. Більшість з них представляє собою типові елементи інтерфейсу "Button", "TreeView", "ComboBox" тощо. Інтерфейс оператора може містити також віртуальні елементи, що імітують реальні елементи керування та пристрої. Для їх відображення у модель введені класи "ВіртЕлемКерув" та "ВіртПристрВідобрІнф" відповідно.

2.4 Розробка стратегії кодування слуховою модальністю

Необхідність дублювання графічного подання в ЛМІ слуховою модальністю виникає за умов, що перешкоджають розпізнаванню графічних елементів, або за умов виникнення небезпечної або аварійної ситуації.

Розглянемо приклад вирішення задачі кодування з використанням звукових сигналів.

Наведемо правила та факти, що стосуються цього виду кодування [21].

П80. ЯКЩО вид алфавіту (слухова, звукові сигнали) (Ф95), ТО частота тональних сигналів повинна бути в межах 200-5000 Гц.

П81. ЯКЩО вид алфавіту (слухова, звукові сигнали) (Ф95), ТО допустимий рівень звукового тиску сигналів на робочому місці від 30 до 100 дБ.

П82. ЯКЩО вид алфавіту (слухова, звукові сигнали) (Ф95), ТО тривалість окремих сигналів не менше 0,2 с.

П83. ЯКЩО вид алфавіту (слухова, звукові сигнали) (Ф95), ТО інтервали між сигналами не менше 0,2 с.

П84. ЯКЩО вид алфавіту (слухова, звукові сигнали) (Ф95), ТО тривалість звучання інтенсивних звукових сигналів не більше 10 с.

- Ф85. умова (об'єкт, сигнал аварійний).
- Ф97. умова (звук, зміна тривалості).
- Ф98. умова (частота, високочастотний маскуючий шум).
- Ф99. умова (частота, наявність в приміщенні поста управління акустичних екранів).
- Ф100. умова (частота, зміна частота).
- Ф101. умова (тиск, зміна звукового тиску).
- Ф102. умова (частота, сигнал застережливий).
- Ф103. умова (тип, сигнал що повідомляє).
- П85. ЯКЩО умова (частота, високочастотний маскує шум) (Ф98), ТО частота до 10000Гц, максимально відрізняється від частоти шуму.
- П86. ЯКЩО умова (частота, високочастотний маскує шум) (Ф98), ТО допустимий рівень звукового тиску від 110 до 120 дБ І необхідно забезпечити перевищення порога маскування від 10 до 16 дБ.
- П87. ЯКЩО умова (частота, наявність в приміщенні поста управління акустичних екранів) (Ф99), ТО частота тональних сигналів 200 - 1000 Гц.
- П88. ЯКЩО умова (частота, зміна частоти) (Ф100), ТО крок зміни не менше 3% по відношенню до вихідної частоти.
- П89. ЯКЩО умова (частота, сигнал застережливий) (Ф102), ТО сигнал повинен бути переривчастим.
- П90. ЯКЩО умова (об'єкт, сигнал аварійний) (Ф85), ТО сигнал повинен бути переривчастим.
- П91. ЯКЩО умова (частота, сигнал застережливий) (Ф102) АБО умова (об'єкт, сигнал аварійний) ТО тривалість сигналу 1 - 3 с.
- П92. ЯКЩО умова (частота, сигнал застережливий) (Ф102) ТО тривалість інтервалів між сигналами 1 - 3 с.
- П93. ЯКЩО умова (частота, сигнал застережливий) (Ф102) ТО частота сигналів 200 - 600 Гц.
- П94. ЯКЩО умова (частота, сигнал застережливий) (Ф102) ТО рівень звукового тиску не вище 80 - 90 дБ.
- П95. ЯКЩО умова (об'єкт, сигнал аварійний) (Ф85) ТО тривалість інтервалів 0,2 - 0,8 с.
- П96. ЯКЩО умова (об'єкт, сигнал аварійний) (Ф85) ТО рівень звукового тиску не вище 100 дБ.
- П97. ЯКЩО умова (об'єкт, сигнал аварійний) (Ф85) ТО частота сигналів 800 - 2000 Гц.
- П98. ЯКЩО умова (тип, сигнал що повідомляє) (Ф103) ТО рівень звукового тиску не вище 95 дБ.

П99. ЯКЩО умова (тиск, зміна звукового тиску) (Ф101) ТО крок зміни не менше 3 дБ.

П100. ЯКЩО умова (звук, зміна тривалості) (Ф97) ТО крок зміни повинен бути не менше 25% по відношенню до вихідної тривалості.

П101. ЯКЩО умова (звук, зміна тривалості) (Ф97) ТО найбільш точно розпізнаються градації частоти і числа перерв.

Проаналізувавши правила П80-101 визначимо необхідну номенклатуру параметрів та їх характеристик, характерних для даного виду алфавіту. Таким чином можна виділити цілі та підцілі при кодуванні звуковими сигналами, а також правила та факти, які необхідні для встановлення відповідних параметрів та їх характеристик. Результати аналізу правил для алфавіту звукові сигнали наведено у табл. 2.2.

Таблиця 2.2 – Результати аналізу правил для алфавіту звукові сигнали

Цілі	Факти	Правила
Ц1. Вибір частоти	Ф98, Ф99, Ф85	П80, П85, П87, П97
Ц1.1. Зміна частоти	Ф100	П88
Ц2. Вибір тривалості	Ф102, Ф85	П82, П84, П91
Ц2.1. Зміна тривалості	Ф97	П100, П101
Ц3. Інтервал між сигналами	Ф85	П89, П90
Ц3.1. Тривалість інтервалу	Ф85, Ф95	П83, П92, П95
Ц4. Рівень звукового тиску	Ф85, Ф98	П81, П86, П94, П96, П98
Ц4.1 Зміна рівня звукового тиску	Ф101	П99

Виходячи з отриманих результатів у якості прикладу побудуємо дерево рішень для досягнення цілі Ц1. Дерево рішень наведено у вигляді табл. 2.3.

Таблиця 2.3 – Дерево рішень для визначення частоти звукового сигналу

Ф98	Ф99	Ф85	Правила, що використовуються
0	0	0	П80
0	0	1	П97
0	1	0	П87
0	1	1	П87, П97
1	0	0	П85
1	0	1	П97
1	1	0	П87
1	1	1	П87, П97

Таким чином отримано рішення задачі кодування об'єкту за допомогою алфавіту звукові сигнали.

2.5 Висновки до другого розділу

Аналіз стану досліджень у області проєктування людино-машинних інтерфейсів SCADA-систем дозволив визначити, що базою для подальшої розробки ІТ інтелектуальної підтримки процесу ергономічного проєктування людино-машинних інтерфейсів SCADA-систем є [7].

Проведений системний аналіз діяльності проєктувальника інтерфейсів ЛМІ дозволив виявити номенклатуру інтелектуальних задач та процедур, що вирішує проєктувальник у проблемних ситуаціях.

В результаті введення у концептуальну модель предметної області "Проєктування ЛМІ" та розробки стратегії кодування слуховою модальністю у вигляді дерев рішень, дістала подальшого розвитку ІТ інтелектуальної підтримки процесу ергономічного проєктування ЛМІ [7].

3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

В даному розділі представлено опис розробки програми системи інтелектуальної підтримки для комп'ютерної підтримки процесу ергономічного проектування ЛМІ SCADA-систем. Для розробки застосовувалася мова програмування C# та середовище розробки Microsoft Visual Studio 2017.

3.1 Моделювання програмного забезпечення

Типовий сценарій роботи проєктувальника ЛМІ з програмою може бути описаний за допомогою прецеденту, що наведений нижче.

Прецедент «Розробка ЛМІ»:

- основний виконавець - проєктувальник ЛМІ;
- зацікавлена особа - проєктувальник ЛМІ;
- передумови: програма відкрита;

Післяумова: ЛМІ розроблено.

Основний успішний сценарій наступний.

Крок 1. Проєктувальник проходить початкове опитування.

Крок 1.1 Проєктувальник отримує попередження про необхідність дублювання зорової модальності слуховою;

Крок 2. Проєктувальник додає об'єкти ТП;

Крок 2.1 Проєктувальник кодує стани об'єктів ТП;

Крок 2.2 Проєктувальник застосовує шаблони.

Крок 3. Проєктувальник додає з'єднувальні елементи;

Крок 4. Проєктувальник додає елементи керування;

Крок 4.1 Проєктувальник проходить опитування;

Крок 4.2 Проєктувальник отримує рекомендації щодо кодування;

Крок 4.3 Проєктувальник застосовує кодування.

Use-case діаграму, що ілюструє даний прецедент наведено на рис. 3.1.

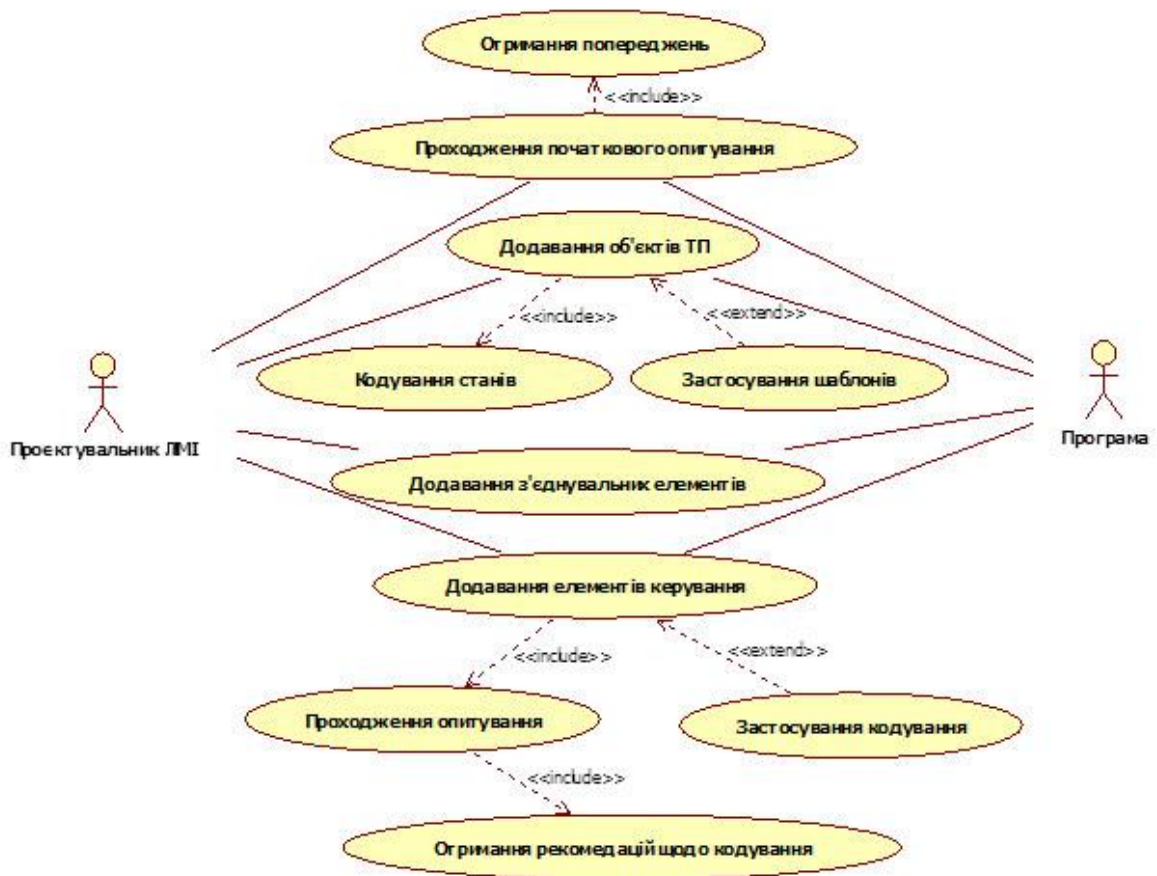


Рисунок 3.1 – Use-case діаграма «Розробка ЛМІ»

Загальна структура програми була розроблена на основі наведеної на рис. 2.2 концептуальної моделі.

Для забезпечення процесу побудови ЛМІ необхідно надати середовище побудови та інструменти кодування ОКК та їх параметрів у двох модальностях: зоровій та слуховій. Таким чином, необхідно передбачити в складі системи підсистему кодування ЛМІ, у якій можна виділити модулі графічного та звукового редакторів.

Завданням даної підсистеми є надання інструментарію для забезпечення можливості кодування об'єктів ЛМІ та їх параметрів у всіх станах, а також планування їх композиції з виділенням зон оперативного простору, що використовують різні формати представлення даних. Модулі графічного та звукового редактора призначені безпосередньо для виконання кодування ОКК

у всіх станах з використанням алфавітів кодування у зоровій та слуховій модальностях.

Збережені у форматі XML дані про представлення ОКК та їх параметрів у різних станах та про компонування елементів ЛМІ є вхідними даними для процесу візуалізації ЛМІ.

Вирішення задач спрощення процесу ергономічного проєктування ЛМІ та візуалізації на робочому місці оператора було покладено на підсистему «Інтелектуальний інтерфейс». Ця підсистема має в своєму складі два модулі: модуль інтелектуальної підтримки та модуль візуалізації. Загальна структура системи інтелектуальної підтримки проєктування ЛМІ SCADA-систем наведена на рис. 3.2.

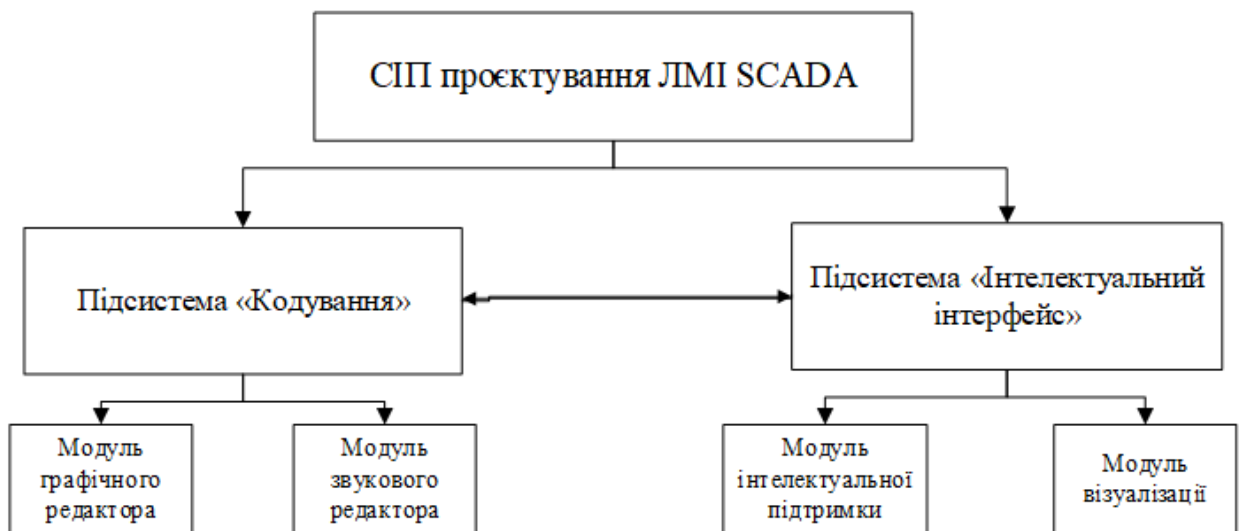


Рисунок 3.2 – Загальна структура системи

Враховуючи складність СІП ЕЗП, в якості діаграми, що відображає статичну модель системи, наведемо діаграму пакетів (рис. 3.3). До модуля інтелектуальної підтримки можна віднести наступні пакети: IntellSystem, що включає в себе TechProcessModules та HmiModules. До модуля візуалізації: WebForms, DataManager, XmlManager, DatabaseManager, Serializer. Пакет TechProcess включає в себе опис структури об'єктів технологічного процесу та використовується як в модулі візуалізації, так і в модулі графічного редактора.

Пакети Graphics та Audio реалізують модулі графічного та звукового редакторів відповідно.

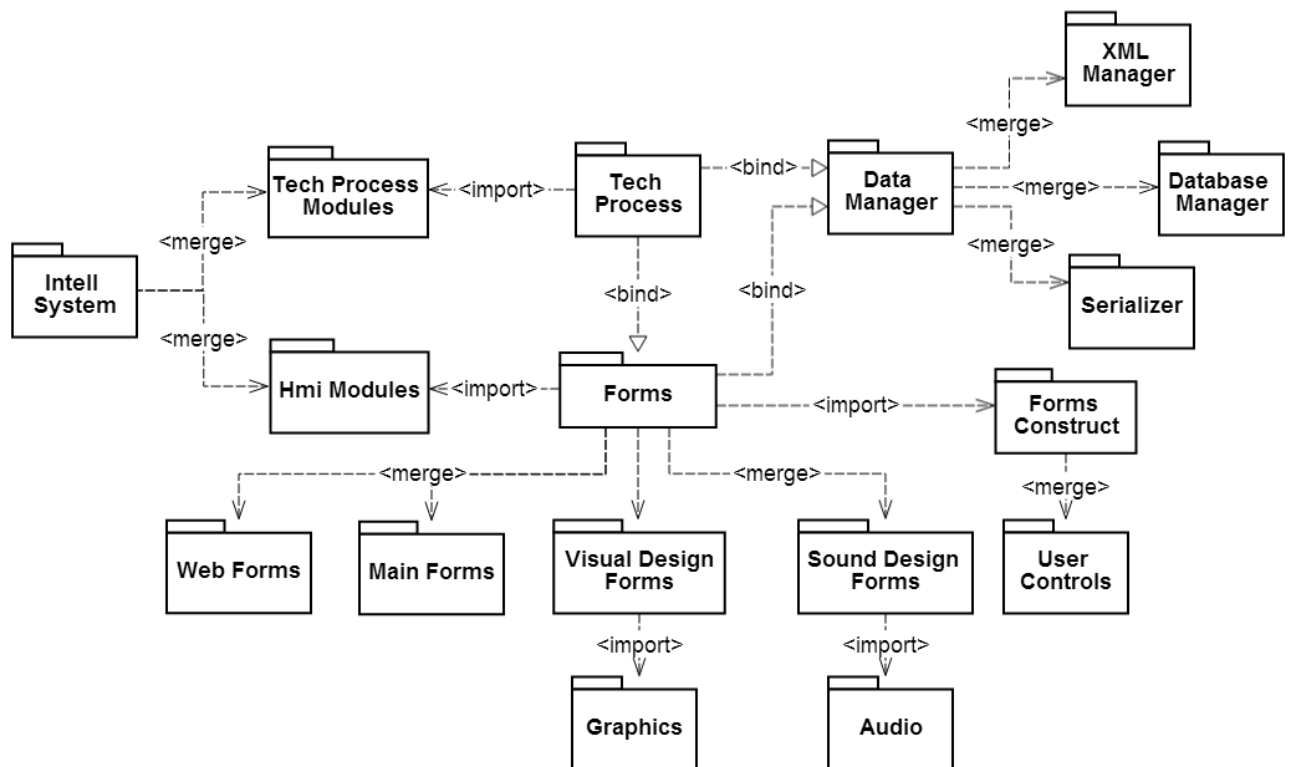


Рисунок 3.3 – Діаграма пакетів системи

3.2 Розробка системи інтелектуальної підтримки

3.2.1 Розробка підсистеми «Кодування»

Для реалізації функцій покладених на підсистему «Кодування», необхідно розробити ієрархію класів, кожний з яких буде направлено на вирішення певних задач.

Розглянемо структуру модуля звукового редактора наведену на рис. 3.4. Головним елементом цього модуля є генератор звукових файлів WaveGenerator, який на базі введених до форми кодування об'єкту FormObjDesign проєктувальником бажаних параметрів звукового сигналу будує файл сигналу, спираючись на описану у класі Chunks структуру звукового файлу у форматі

WAV. Інформація про звукове кодування OKK описується класом SoundEncoder, що належить до пакету TechProcess.

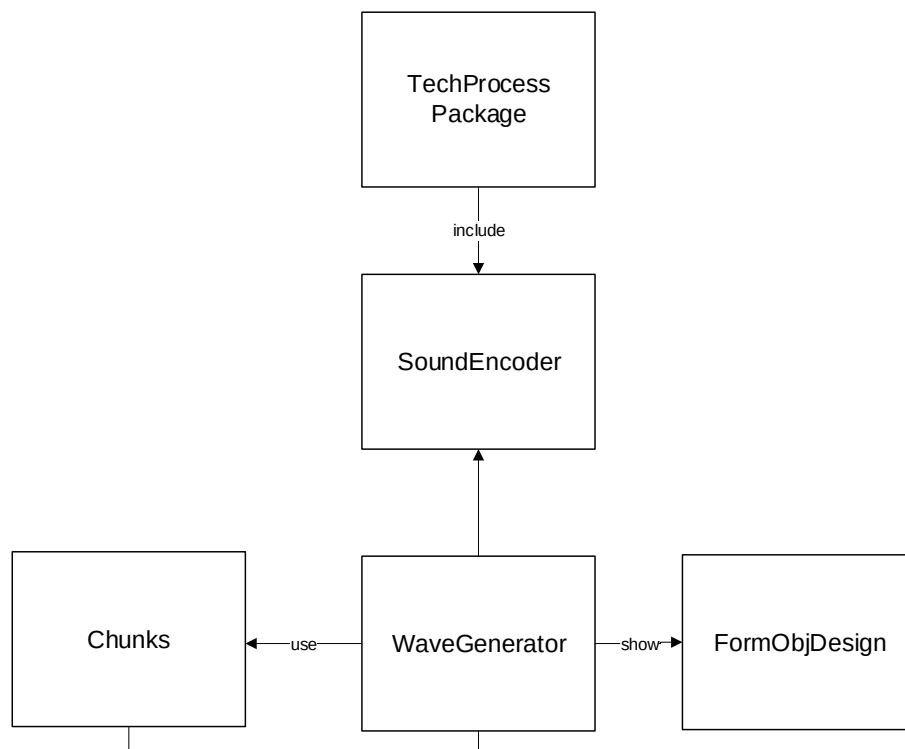


Рисунок 3.4 – Структура класів модуля звукового редактора

Основні дані та методи класу WaveGenerator наведені у табл. 3.1. Для простоти методи доступу до даних get/set, конструктори та деструктори не описані.

Структура модулю графічного редактора є схожою на структуру модуля звукового редактора, однак містить значно більшу кількість класів, що описують інструменти графічного кодування, та форм майстра графічного кодування. Це пояснюється більшою кількістю алфавітів графічного кодування та загальною складністю процесу побудови графічного зображення. Структуру класів модулю графічного редактора наведено на рис. 3.5.

Таблиця 3.1 – Основні дані та методи класу WaveGenertor

Атрибут	Тип/Значення,що повертається	Назва	Параметри	Опис
private	double	length		тривалість сигналу
private	int	volume		гучність сигналу
private	double	frequency		частота сигналу
private	int	pauseNum		кількість пауз переривчастого сигналу
private	double	pauseLength		тривалість пауз
private	int	volumeChange		зміна гучності після паузи
private	int	frequencyChange		зміна частоти після паузи
private	int	lengthChange		зміна тривалості після паузи
private	string	filepath		шлях до файлу
public	void	generate	length volume frequency	генерує дані звукового сигналу
public	void	save	filepath	записує згенерований файл

За побудову графічного зображення об'єкта відповідає клас DrawGraphics, який отримує дані з форм майстра графічного кодування, що входять до складу пакету VisualDesignForms, та використовує класи з пакету GeoFigures у випадку вільного редагування зображення. Також для побудови анімаційних зображень використовується додаткова бібліотека GifComponents. Аналогічно до звукового кодування, інформацію про графічне кодування об'єктів описує клас VisualEncoder з пакету TechProcess.

Класи, що входять до складу пакету GeoFigures, наведено у табл. 3.2.

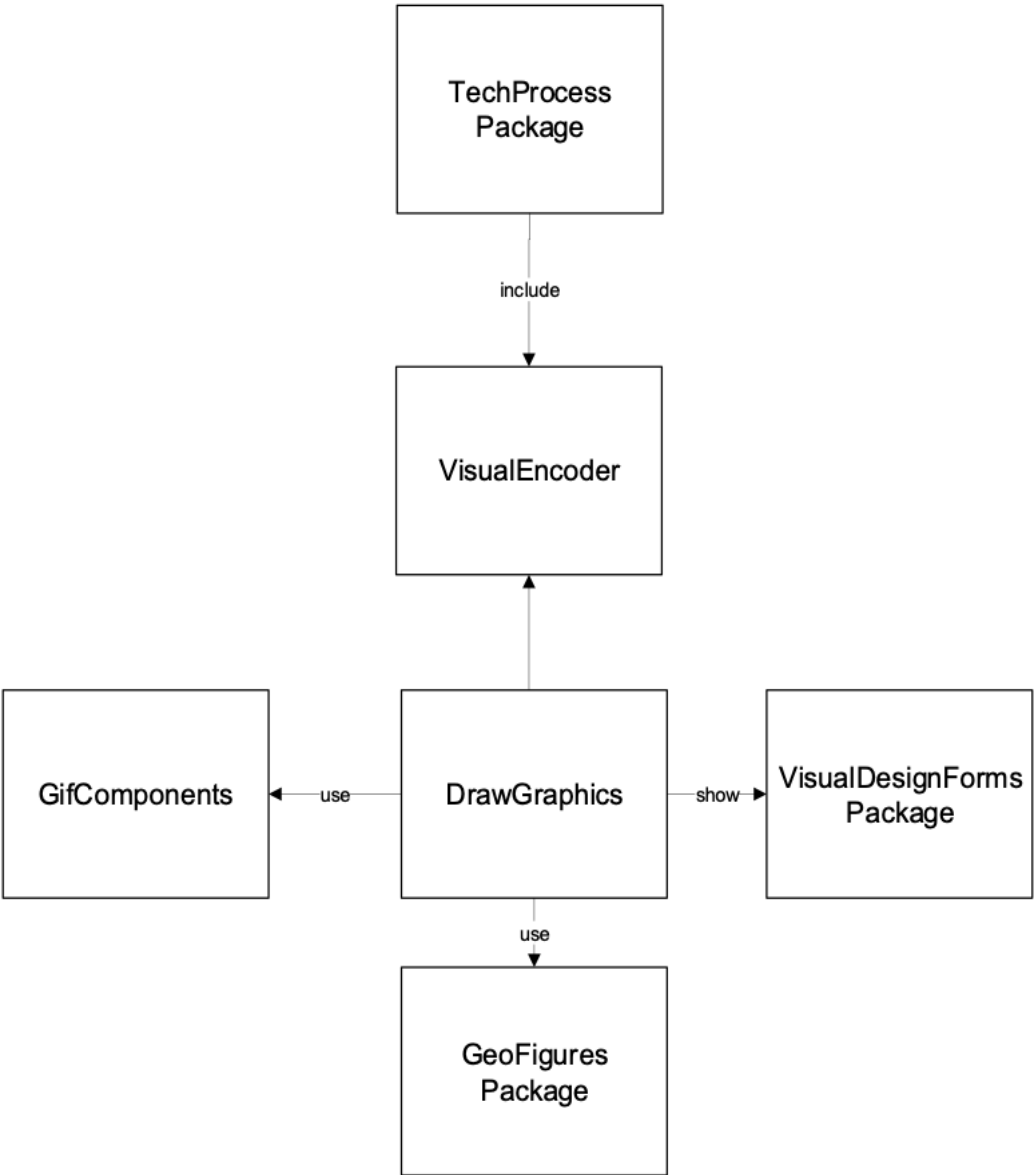


Рисунок 3.5 – Структура класів модуля графічного редактора

Структуру класів модуля графічного редактора наведено на рис. 3.5.

Таблиця 3.2 – Опис класів пакету GeoFigures

Назва	Призначення
GeoEllipse	Описує інструмент малювання еліпсів
GeoFigure	Абстрактний клас для інструментів малювання
GeoLine	Описує інструмент малювання ліній
GeoRectangle	Описує інструмент малювання прямокутників
GeoText	Описує інструмент вставки тексту
GeoTriangle	Описує інструмент малювання трикутників

Класи та форми, що входять до складу пакету VisualDesignForms, наведено у табл. 3.3.

Таблиця 3.3 – Опис класів та форм пакету VisualDesignForms

Назва	Призначення
Designer	Керує процесом графічного кодування у режимі майстра
HatchItem	Описує параметри штриховки, що використовується
LineItem	Описує параметри ліній, що використовуються
SymbolItem	Описує параметри відображення символів
FormShape	Форма кодування алфавітом форма
FormAnimation	Форма кодування анімацією
FormSize	Форма кодування розміру
FormPosition	Форма кодування просторовим розташуванням
FormText	Форма кодування літеро-цифровим алфавітом
FormBrightness	Форма кодування яскравістю
FormColor	Форма кодування кольором

Опис основних методів класу DrawGraphics наведено у табл. 3.4.

Таблиця 3.4 – Основні методи класу DrawGraphics

Атри- бут	Значення, що повертається	Назва	Параметри	Опис
1	2	3	4	5
public	bitmap	createBitmapFromStr	text font colorBG brush	створює зображення з текстом
public	image	scaleImage	baseImage newImageSize	масштабує зображення
public	void	centerDrawImage	target background centerme	центрує зображення
public	void	fillColorImage	image point newColor	заповнює зображення заданим кольором
public	image	createImage	size color	створює нове зображення
public	image	createImage	size color img	створює нове зображення на основі імпортованого

Продовження таблиці 3.4

1	2	3	4	5
public	bitmap	rotateImage	image rotateAtX rotateAtY angle bNoClip	виконує поворот зображення
public	image	createGif	List images name delay	створює файл анімації
public	image	SetBrightnessGammaC ontrast	srcImage brightness gamma contrast	встановлює параметри яскравості гами та контрасту
public	color	ModifyColorBrightness	basecolor factor	змінює яскравість кольору

Клас FormMode описує інструменти роботи зі сценами. Взаємодію даного класу з іншими структурними елементами системи відображено на рис. 3.6.

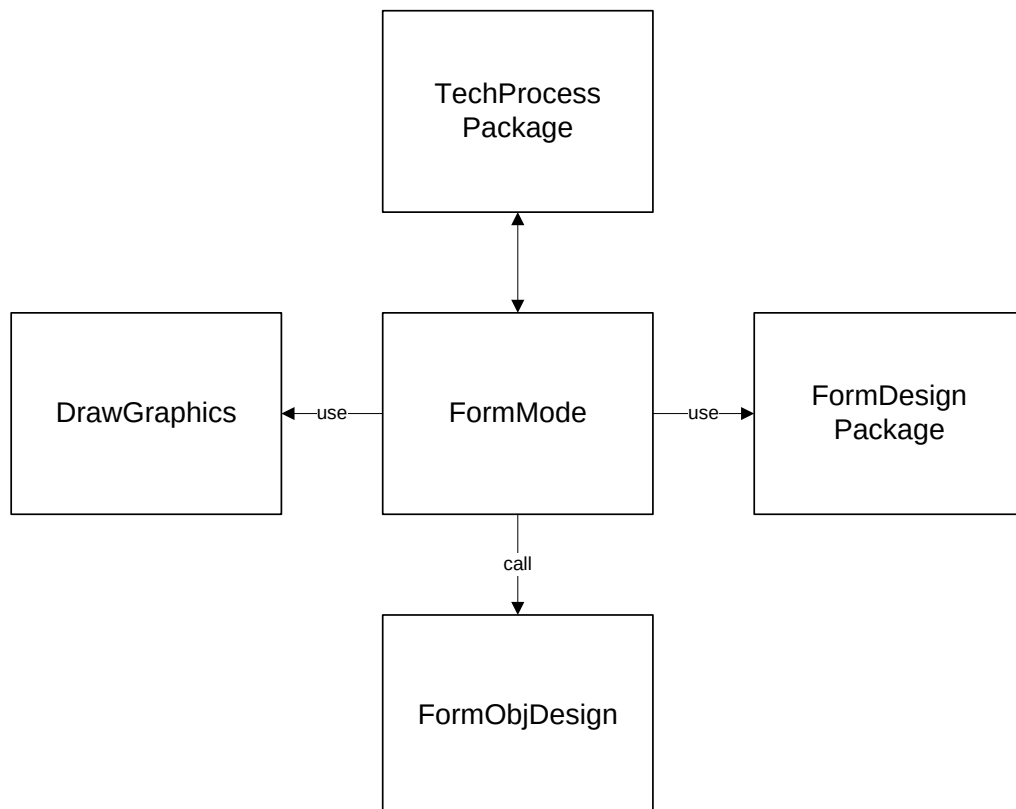


Рисунок 3.6 – Структурні зв'язки редактору сцен

Дана форма надає середовище для побудови сцен з використанням зображень закодованих ОКК та додаткових графічних елементів (комутації, введення/виведення даних, управління). На даній формі передбачені інструменти поділу сцен на зони, що можуть мати відмінні формати представлення даних. Також дана форма містить дерево ОКК обраного етапу ТП та їх параметрів, звідки виконується виклик модулів графічного та звукового редакторів для кодування обраного об'єкта.

Таким чином, за виконуваними функціями дана форма представляє собою редактор сцен. Як можна побачити, редактор сцен використовує деякі методи класу DrawGraphics, що належить до модуля графічного редактора, та класи пакету FormDesign, що містять опис службових елементів форми (ResMovControl, Grid, DialBar) та конструктор форм FormsConstruct, що виконує побудову дерева ТП. З дерева ОКК можливо викликати форму кодування об'єктів, що пов'язана з модулями графічного та звукового редакторів. Також редактор сцен використовує класи пакету TechProcess, які описують ТП (Mode, TechProcessObject, State, Parameter) та параметри кодування ОКК (VisualEncoder).

Опис основних даних та методів класу FormMode наведено у табл. 3.5.

Таблиця 3.5 – Основні дані та методи класу FormMode

Атрибут	Тип/Значення, що повертається	Назва	Параметри	Опис
1	2	3	4	5
private	PropertyGrid	propertyGrid		перелік властивостей об'єкта
public	Mode	mode		поточний режим ТП
private	ResMovControl	activeUIControl		контроль поточного графічного елемента

Продовження таблиці 3.5

1	2	3	4	5
private	void	createElements	list	створює меню додаткових графічних елементів
private	void	refreshImageContainers	parentControl	оновлює зображення елементів
private	void	createNamedLabel	parent location text vencoder	створює текстовий елемент
private	void	setContainerProperties	parent location vencoder	встановлює властивості елемента
private	void	createTplItemContainer	parent location tplItem	створює елемент для кодованого об'єкту
private	void	removeControls		видалити елемент
private	void	createSplit	splitDistance splitOrientation	створює розділення на зони

Опис основних даних та методів класу SoundEncoder наведено у табл. 3.6, а класу VisualEncoder у табл. 3.7.

Таблиця 3.6 – Дані та методи класу SoundEncoder

Атрибут	Тип/Значення, що повертається	Назва	Опис
1	2	3	4
public	string	soundPath	шлях до файлу звукового сигналу
public	int	volume	гучність сигналу
public	double	frequency	частота сигналу
public	double	signalLength	тривалість сигналу
public	double	pauseLength	тривалість паузи
public	int	pauseCount	кількість пауз
public	int	volumeDelta	зміна гучності сигналу
public	int	lengthDelta	зміна тривалості відрізка сигналу
public	int	frequencyDelta	зміна частоти сигналу

Продовження таблиці 3.6

1	2	3	4
public	string	textToSpeech	текст мовного повідомлення
public	int	textSpeechRate	швидкість мовного повідомлення
public	SoundEncoder	SoundEncoder	конструктор

Таблиця 3.7 – Дані та методи класу VisualEncoder

Атрибут	Тип/Значення, що повертається	Назва	Параметри	Опис
public	image	image		графічне зображення
public	size	size		розмір зображення
public	point	screenLocation		координати розташування
public	color	bgColor		колір фону
public	list<SymbolItem>	symbols		список символів, з яких складається зображення
public	ResMovControl	container		контейнер, що містить зображення
public	image	drawSymbols	needAnimate	відображає усі символи зі списку
public	void	drawSymbol	symbol image	відображає обраний символ
public	void	copyTo	outEncoder	копіює параметри графічного кодування

3.2.2 Розробка підсистеми «Інтелектуальний інтерфейс»

Наведемо визначення основних класів модулів інтелектуальної підтримки та візуалізації.

Клас `IntellStateModule` (табл. 3.8) є нащадком абстрактного класу `IntellModule` та реалізує інтелектуальний модуль кодування стану об'єкта. Він зберігає словник фактів, доступ до яких здійснюється з використанням унікального ключа. В ньому передбачені методи для проведення опитування

проектувальника, активації правил, формування рекомендацій та імпорту фактів з іншого модуля.

Клас `SerializedControl` (табл. 3.9) призначений для серіалізації та дисеріалізації структури створеного ЛМІ. Оскільки стандартні контроли `Windows.Form.Control` безпосередньо не можуть бути серіалізовані, єдиним можливим засобом їх збереження є створення власної структури, яка б описувала ці контроли, а при десеріалізації створювала нові екземпляри із заданими параметрами. Екземпляри цього класу використовується в класі `SerializedStructure`, що, в свою чергу, задіяний в singleton-класі `SerializeManager`.

Таблиця 3.8 – Основні дані та методи класу `IntellStateModule`

Атрибути	Тип/Значення, що повертається	Назва	Параметри	Опис
1	2	3	4	5
private	Dictionary <string, Fact>	facts		Словник фактів модуля
public	void	ShowGoal Interview		Проводить опитування розробника за мету стану
public	void	ShowShape Interview		Проводить опитування розробника за форму знаків
public	void	ShowText Interview		Проводить опитування розробника за кодування текстом
public	void	ShowColor Interview		Проводить опитування розробника за кодування кольором
public	void	ShowBrightne ssInterview		Проводить опитування розробника за кодування яскравістю
public	void	ShowSound Interview		Проводить опитування розробника за кодування звуком
public	void	SetFactsFrom StateType	StateType stateType	Встановлює факти відповідно типу стану
public	string	GetShapeRec ommendation		Надає рекомендації за формою

Продовження таблиці 3.8

1	2	3	4	5
public	string	GetSizeRecommendations		Надає рекомендації за розміром
public	string	GetTextRecommendations		Надає рекомендації за текстом
public	string	GetBrightnessRecommendations		Надає рекомендації за яскравістю
public	string	GetSoundRecommendation		Надає рекомендації за звуковими сигналами
public	string	GetVoiceRecommendation		Надає рекомендації за голосовими сигналами
public	void	ImportFacts	Dictionary <string, Fact> inFacts	Імпортує факти з іншого модуля
private	void	ActivateGoalRules		Активізує правила мети стану
private	void	ActivateShapeRules		Активізує правила кодування формою
private	void	ActivateColorRules		Активізує правила кодування кольором

Таблиця 3.9 – Основні дані та методи класу SerializedControl

Атрибути	Тип/Значення, що повертається	Назва	Опис
1	2	3	4
private	string	typeName	Тип контролю
private	Visual Encoder	visual Encoder	Графічне представлення параметра або стану
private	Color	backColor	Колір фону
private	Color	foreColor	Колір шрифту
private	Size	size	Розмір контролю
private	Point	location	Позиція
private	string	text	Текст
private	Font	font	Шрифт
private	Content Alignment	textAlign	Вирівнювання тексту
private	bool	visible	Відображення
private	Image	backImage	Фоновий рисунок

Продовження таблиці 3.9

1	2	3	4
private	Image	image	Рисунок
public	ResMov Control	GetControl	Створює контрол з заданими параметрами
private	void	Modify Control	Модифікує значення параметрів контрола

Клас `TechProcessObject` (табл. 3.10) описує об'єкт технологічного процесу та характеризується набором параметрів та станів. Має методи для зчитування даних з XML файлу. Кожен об'єкт в заданий час знаходиться в поточному стані (`CurrentState`), що однозначно визначається значеннями параметрів при визові методу `SetCurrentState`.

Таблиця 3.10 – Основні дані та методи класу `TechProcessObject`

Атрибути	Тип/Значення, що повертається	Назва	Параметри	Опис
1	2	3	4	5
public static readonly	string	xmlTreePath		Абсолютний шлях до тегу об'єкта в XML файлі
public static readonly	string	xmlTreeTagName		Тег об'єкта в документі опису технологічного процесу
public static readonly	string	xmlObjTagName		Тег об'єкта в документі опису об'єкта в XML файлі
public static readonly	string	xmlAttrUuid		Атрибут uuid об'єкта в XML файлі
public static readonly	string	xmlAttrName		Атрибут назви об'єкта в XML файлі
private	Mode	mode		Посилання на батьківський режим
private	string	name		Назва
private	string	uuid		Ідентифікаційний номер
private	State	currentState		Поточний стан

Продовження таблиці 3.10

1	2	3	4	5
private	List<State>	States		Список станів
private	List <Parameter>	parameters		Список параметрів
private	Object Module	intellModule		Інтелектуальний модуль об'єкта
private	void	ReadStates FromXml	XmlDocu ment doc	Зчитує стани об'єкта з XML файлу
private	void	ReadParam etersFromXm l	XmlDocu ment doc	Зчитує параметри об'єкта з XML файлу
public	void	SetCurrent State		Встановлює поточний стан об'єкта

Абстрактний клас Parameter (табл. 3.11) є батьківським для класів QuantitativeParameter та QualitativeParameter. Має методи для зчитування даних з XML файлу, створення контейнеру заданого типу та його оновлення, перевірки значення на належність до певного стану.

Таблиця 3.11 – Основні дані та методи класу Parameter

Атрибути	Тип/Значення, що повертається	Назва	Параметри	Опис
1	2	3	4	5
public static readonly	string	xmlPath		Абсолютний шлях до тегу стану в XML файлі
public static readonly	string	xmlAttrUuid		Атрибут uuid в XML файлі
public static readonly	string	xmlAttrName		Атрибут назви в XML файлі
public static readonly	string	xmlAttrValue		Атрибут значення в XML файлі
private	TechProcessO bject	techProcess Object		Посилання на батьківський об'єкт

Продовження таблиці 3.11

1	2	3	4	5
private	string	name		Назва
private	string	uuid		Ідентифікаційний номер
private	string	value		Поточне значення
private	bool	isControlled		Параметр потребує управління
private	bool	isObserved		Параметр потребує нагляду
private	Visual Encoder	visualEncoder		Графічне представлення параметра
public	void	Create Container	Type controlType	Створює контейнер для відображення параметра
public	void	Update Container		Оновлює контейнер параметра

Клас State (табл. 3.12) описує стан об'єкту технологічного процесу та характеризується набором параметрів. Має методи для зчитування даних з XML файлу. Для збереження закодованої інформації використовуються об'єкти класів VisualEncoder та SoundEncoder.

Таблиця 3.12 – Основні дані та методи класу State

Атрибути	Тип/Значення, що повертається	Назва	Параметри	Опис
1	2	3	4	5
public static readonly	string	xmlTagName		Тег стану в XML файлі
public static readonly	string	xmlPath		Абсолютний шлях до тегу стану в XML файлі
public static readonly	string	xmlAttrUuid		Атрибут uuid в XML файлі
public static readonly	string	xmlAttrName		Атрибут назви в XML файлі

Продовження таблиці 3.12

1	2	3	4	5
private	TechProcessObject	techProcessObject		Посилання на батьківський об'єкт
private	string	name		Назва
private	string	uuid		Ідентифікаційний номер
private	StateType	stateType		Тип стану
private	Sound Encoder	sound Encoder		Звукове представлення стану
private	Visual Encoder	visual Encoder		Графічне представлення стану
private	List<Parameter>	parameters		Список параметрів
private	StateModule	intellModule		Інтелектуальний модуль стану
private	void	ReadParametersFromXml	XmlDocument doc	Зчитує параметри стану з XML файлу

З іншими класами, їх визначенням та реалізацією можна ознайомитись в додатку А “Текст програми”.

3.3 Висновки до третього розділу

Проведено моделювання програмного забезпечення СІП, що розробляється. Розроблено структуру СІП ЕЗП та дві моделі в нотації мови UML: діаграма прецедентів, що описувала взаємодію користувача з СІП та діаграма пакетів цієї системи.

На основі діаграми прецедентів розроблено структури класів для обох підсистем СІП ЕЗП. Надано опис розробки цих класів та їх основних методів, необхідних для інтелектуальної підтримки проектувальника ЛМІ SCADA-систем.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Приклад проєктування людино-машинного інтерфейсу «Система охолодження»

Розглянемо процес інтелектуальної підтримки проєктування на прикладі створення ЛМІ системи охолодження (Refrigeration System).

На першому етапі проводиться опитування стосовно зовнішніх умов роботи оператора SCADA-системи (рис. 4.1). Це необхідно для з'ясування питання «Чи є необхідним при проєктуванні ЛМІ дублювання зорової модальності слуховою?»

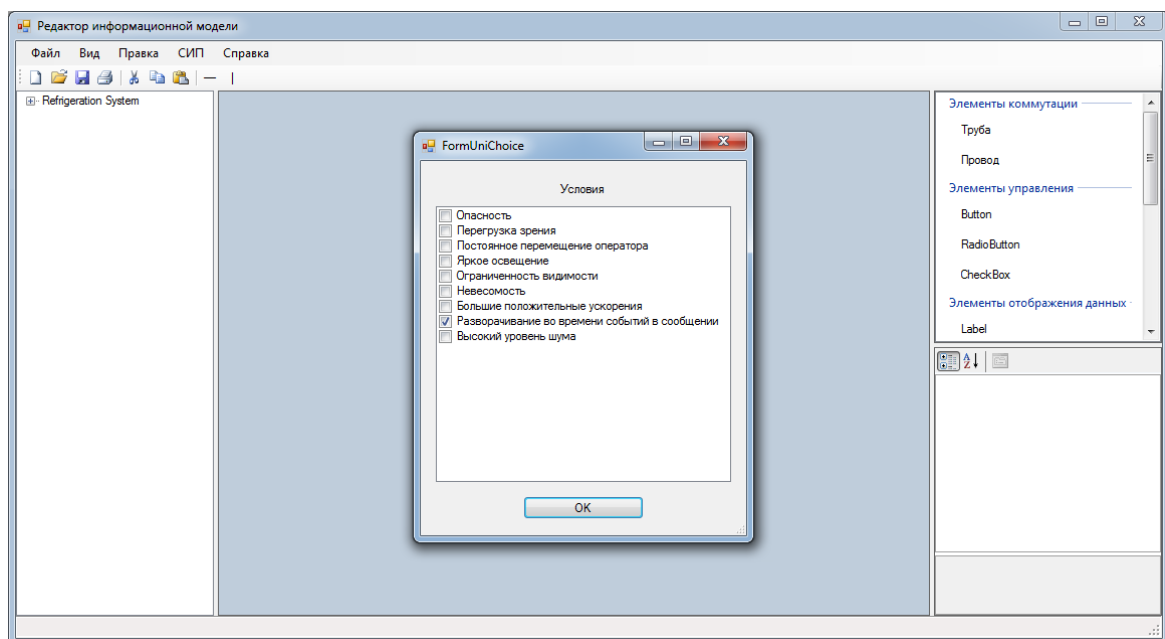


Рисунок 4.1 – Опитування стосовно зовнішніх умов роботи оператора

Далі виконаємо кодування ОКК, що включає в себе візуальне та звукове кодування їх станів та параметрів. При виборі модальності кодування об'єктів буде виведене повідомлення, в якому рекомендується дублювання зорової модальності слуховою (рис. 4.2).

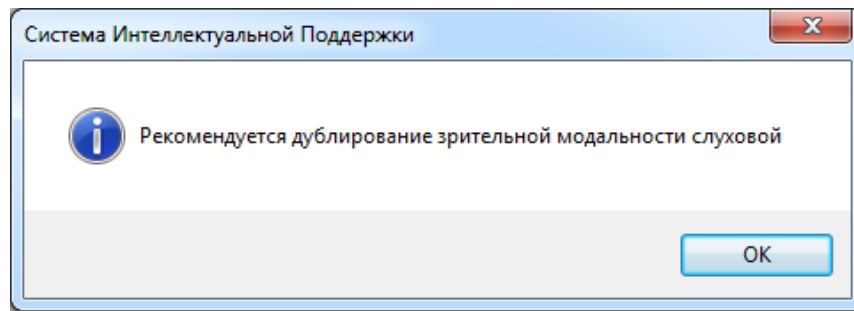


Рисунок 4.2 – Рекомендації по вибору модальності

Для вибору алфавіту кодування скористаємось СИП. В результаті проведення опитування було запропоновано алфавіт, наведений на рис. 4.3, що включає в себе кодування формою, кольором, розміром та ін.

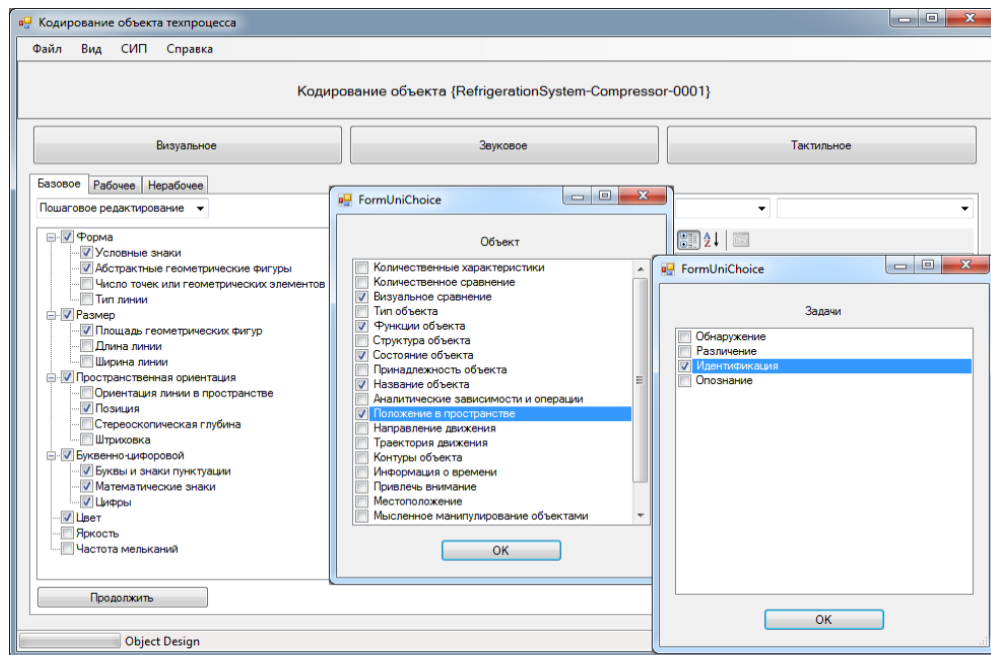


Рисунок 4.3 – Вибір алфавіту кодування

На кожному етапі візуального кодування стану об'єкта розробнику надається набір рекомендацій з питань ергономіки (приклад для кодування кольором наведений на рис. 4.4).

Результат кодування робочого стану об'єкту Compressor1 умовним знаком, геометричною фігурою, розміром та кольором наведений на рис. 4.5.

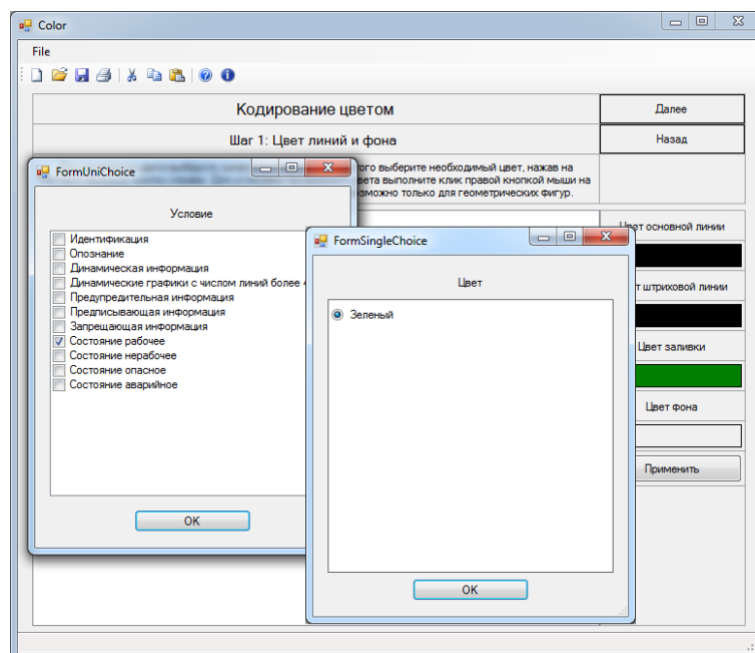


Рисунок 4.4 – Кодування кольором

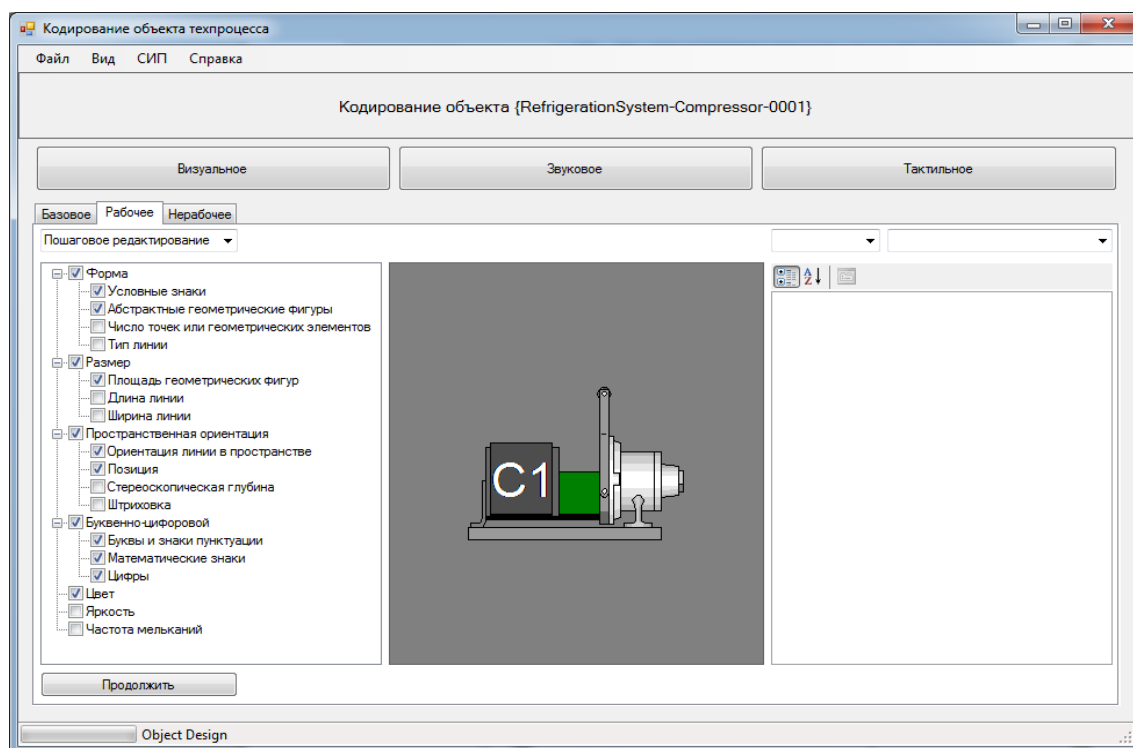


Рисунок 4.5 – Результат кодування робочого стану об'єкта

Після переходу до вікна кодування об'єктів та вибору слухової модальності кодування у даному вікні відкриється вкладка звукового редактору, вигляд якої наведено на рис. 4.6.

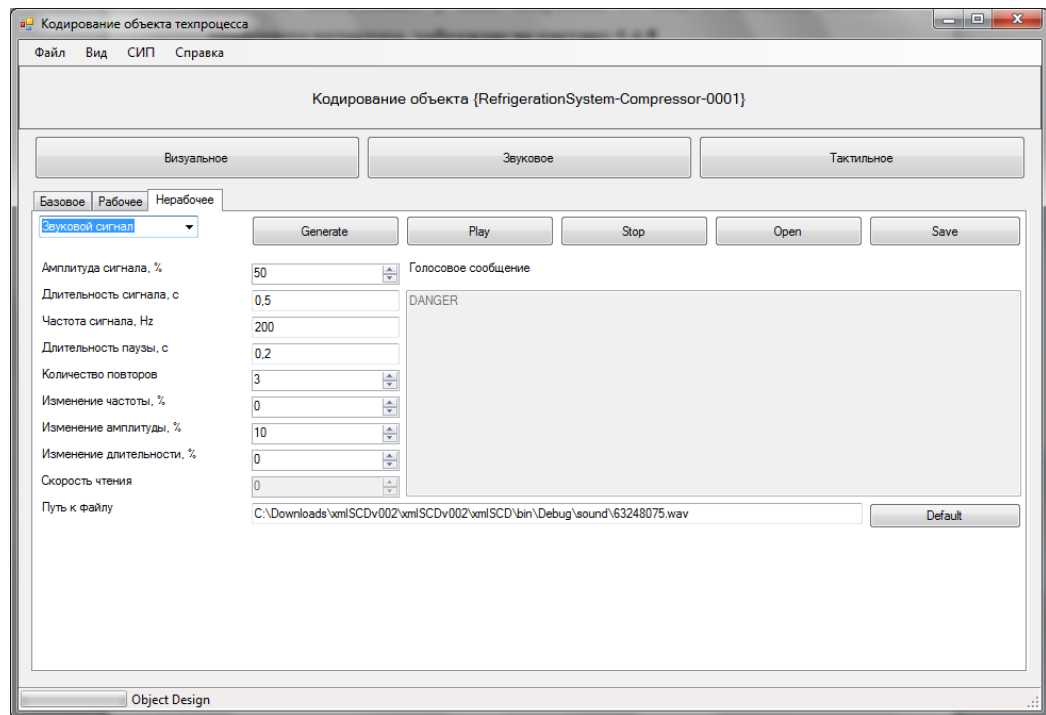


Рисунок 4.6 – Вікно звукового редактору

У верхній частині вікна звукового редактора можна побачити вкладки станів об'єкта, що кодується. Вони визначають для якого стану даного об'єкта буде закодовано звуковий сигнал.

У лівій верхній частині вкладки кодування розташовано випадаюче меню вибору алфавіту кодування – звуковий чи мовний сигнал. У залежності від обраного алфавіту активуються комірки параметрів відповідного сигналу.

У нижній частині вікна розташовано строку адреси для вибору шляху збереження звукового файлу та кнопку встановлення шляху за замовченням.

У верхній частині вікна розташовані кнопки створення звукового файлу, його відтворення та зупинки відтворення, відкриття існуючого файлу та збереження створеного файлу.

Далі закодуємо параметр Pressure об'єкта. Для його відображення рекомендується використовувати графік або стрілочний прилад (рис. 4.7).

Всі інші об'єкти ТП кодуються аналогічним чином. Далі виконаємо поділ екрану на області (фрейми) та позиціонування в них закодованих об'єктів та елементів ЛМІ.

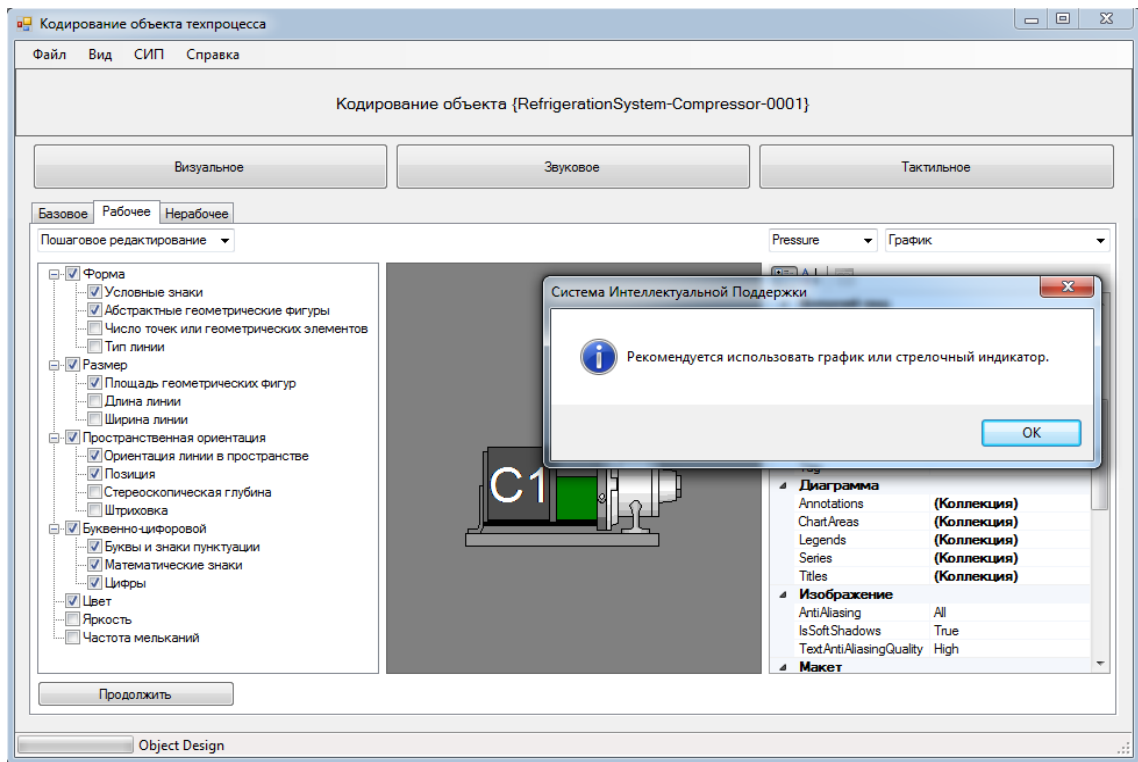


Рисунок 4.7 – Кодування параметру об'єкта

При проєктуванні ЛМІ розробнику надаються рекомендації стосовно основних параметрів кожного елемента (рис. 4.8).

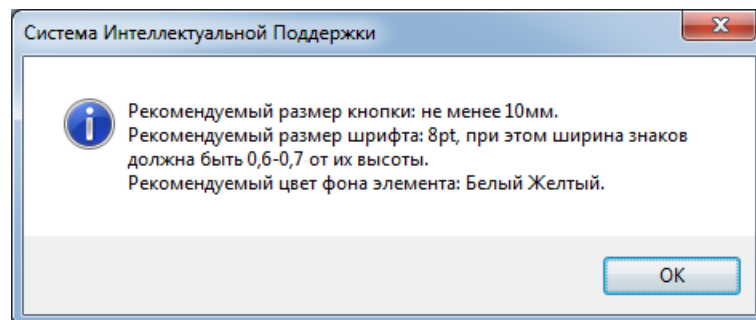


Рисунок 4.8 – Рекомендації по проєктуванню ЛМІ

Результат проєктування ЛМІ системи охолодження наведений на рис. 4.9.

Для звернення до запущеного web застосунку з термінальних станцій достатньо ввести його адресу на сервері у вікні браузера. Приклад роботи web застосунку наведений на рис. 4.10.

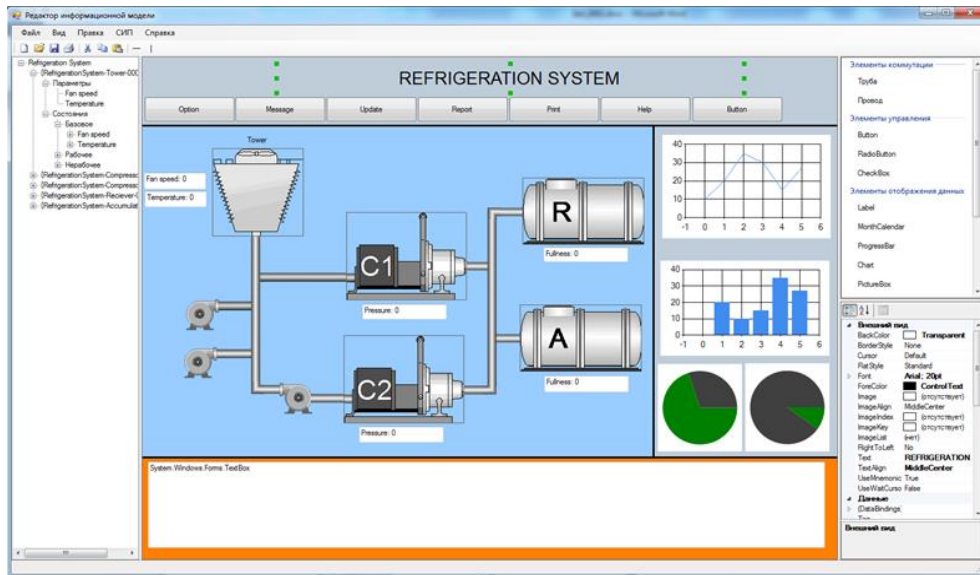


Рисунок 4.9 – Результат проектування ЛМІ

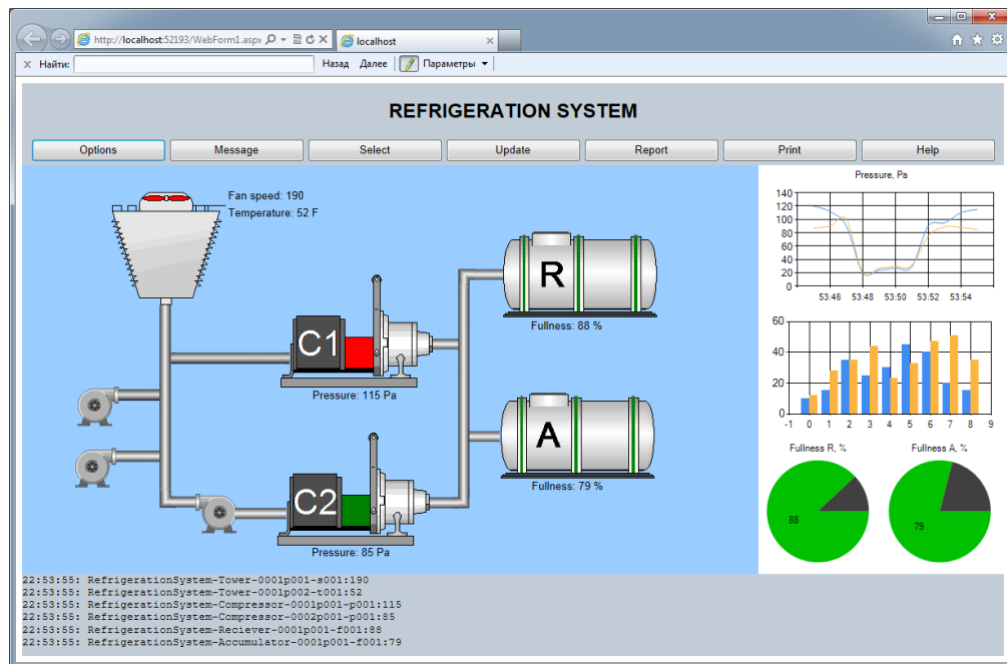


Рисунок 4.10 – Приклад роботи web застосунку

4.3 Висновки до четвертого розділу

У даному розділі представлено процес інтелектуальної підтримки діяльності проєктувальника ЛМІ на прикладі проєктування фрагментів ЛМІ SCADA-системи «Система охолодження».

Методологія ергономічного забезпечення проектування ЛМІ, що надається системою інтелектуальної підтримки базується на інформаційній технології інтелектуальної підтримки процесу ергономічного проектування людино-машинних інтерфейсів SCADA-систем, що була розроблена у попередніх розділах.

Практичне використання розробленої СІП ЕЗП, показало що розроблений продукт здатен задовольняти вимогам щодо встановленого або передбаченого використання та надає підтримку при кодуванні технологічної інформації двома типами модальностей – зоровою та слуховою.

5 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

Проект системи інтелектуальної підтримки, що розробляється у даній дипломній кваліфікаційній роботі магістра, призначений для оптимізації процесу прийняття різних видів рішень при проєктуванні людино-машинних інтерфейсів (ЛМІ) SCADA-систем і покликаний надати підтримку особі, що приймає рішення (ОПР). Програма дозволяє скоротити час і трудомісткість процесу проєктування ЛМІ, на основі введених даних та відповідних правил для розробки. Згенеровані програмою рішення будуть обґрунтованими з точки зору ергономічного забезпечення, що може гарантувати отримання найбільш оптимального результату при взаємодії людини-оператора з інтерфейсом.

В даний час ринок подібного програмного забезпечення представлений дещо схожими програмними продуктами, однак вони спрямовані на інші (не ергономічні) напрямки проєктування інтерфейсів, що не дозволяє використовувати їх в нашому випадку, а отже є необхідність в розробці власного програмного застосунку.

Програмний засіб, що розробляється, враховує специфіку роботи спеціаліста організації/підрозділу з проєктування ЛМІ і дозволяє за рахунок підвищення продуктивності праці (розрахунки виконуються за 450 годин замість 1000, крім того в програму введені додаткові функції) зекономити на заробітній платі, на суму 50 грн в годину протягом 550 годин.

5.1 Планування розробки програмного продукту

Весь комплекс розробки програмного продукту можна розділити на такі важливі етапи. Для кожного етапу обов'язково вказуються трудомісткість, кількість виконавців і тривалість робіт. В розробці беруть участь програміст протягом трьох місяців і консультант з ЛМІ протягом двох місяців. Розробка починається першого вересня і повинна бути виконана до першого грудня 2020 року [22].

Тривалість робіт визначають за формулою (5.1):

$$T_{\text{ц}} = \frac{Q}{R}, \quad (5.1)$$

де $T_{\text{ц}}$ – тривалість циклу, днів;

Q - трудомісткість, людино-днів;

R - кількість виконавців, чол.

Отримана інформація зведена в табл. 5.1.

Таблиця 5.1 – Завдання та обов'язки по розробці програми

Назва роботи	Трудомісткість		Виконавці	Тривалість, днів
	люд.-дні	% до підсумку		
1. Аналіз предметної області (ПО)	10	9,3	програміст консультант	5
2. Визначення вимог до програмного продукту	20	18,5	програміст консультант	10
3. Проектування структури програми	18	16,6	програміст консультант	9
4. Розробка схеми функціонування програми	10	9,3	програміст	10
5. Створення програмного коду	10	9,3	програміст	10
6. Тестування та налагодження програмного продукту	20	18,5	програміст консультант	10
7. Складання програмної документації	20	18,5	програміст консультант	5
Разом	108	100		64

За даними табл. 5.1 складається зведений стрічковий графік планування розробки програмного продукту, який представляє собою таблицю, в першому стовпці якої розміщені в порядку збільшення термінів початку виконання всі види роботи, а навпаки - календарний період їх виконання. Даний графік наведено в табл. 5.2.

Таблиця 5.2 – Зведений стрічковий графік планування розробки програмного продукту

Назва роботи	Календарний період, дні												
	01.09 - 07.09	10.09 - 14.09	17.09 - 21.09	24.09 - 29.09	01.10 - 05.10	08.10 - 12.10	16.10 - 20.10	22.10 - 26.10	29.10 - 02.11	05.11 - 09.11	12.11 - 17.11	19.11 - 23.11	26.11 - 30.11
1. Аналіз предметної області (ПО)	■												
2. Визначення вимог до програмного продукту		■	■										
3. Проектування структури програми				■	■								
4. Розробка схеми функціонування						■	■						
5. Створення програмного коду								■	■				
6. Тестування та налагодження програмного продукту										■	■		
7. Складання програмної документації												■	■

■ - програміст ■ - консультант

5.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складається калькуляція кошторисної вартості робіт, яка включає наступні статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок (ЄСВ);

- витрати на спеціальне обладнання;
- матеріали і комплектуючі вироби;
- накладні витрати;
- податки.

5.2.1 Розрахунок основної заробітної плати

Витрати за цією статтею складаються з планового фонду зарплати всіх категорій працівників, зайнятих в розробці програми. Розрахунок зарплати ведеться на підставі даних про трудомісткості, представлених в табл. 5.3.

Таблиця 5.3 - Розрахунок основної заробітної плати

Посада виконавця	чисельність, чол.	місячний оклад, грн.	Кількість місяців роботи	Сума ЗП, грн.
Програміст	1	7000	3	21000
Консультант	1	4500	2	9000
Разом	2			30000

5.2.2 Розрахунок додаткової заробітної плати

Додаткову заробітну плату приймають рівною 10% від основної заробітної плати працівників і розраховують за формулою (5.2):

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 10\%. \quad (5.2)$$

Підставивши величину основної заробітної плати в дану формулу, отримуємо за формулою (5.3):

$$ЗП_{\text{доп}} = 30000 \cdot 0,1 = 3000 \text{ грн.} \quad (5.3)$$

5.2.3 Відрахування на єдиний соціальний внесок

Ці відрахування становлять 22% і беруться від основної та додаткової заробітної плати за формулою (5.4):

$$OT = (ЗП_{осн} + ЗП_{дод}) \cdot 0,22. \quad (5.4)$$

Розраховуємо:

$$OT = (30000 + 3000) \cdot 0,22 = 7260 \text{ грн.}$$

5.2.4 Визначення витрат на матеріали

Використовується три найменування матеріалів: диск CD-R - 5 грн.; картридж - 150 грн. і папір 50 грн. (1 упаковка).

Витрати на матеріали розраховують за формулою (5.5):

$$M = \sum_{i=1}^n (\text{Ц}_i \cdot N_i \cdot (1 + K_{т.з}) - \text{Ц}_{io} \cdot N_{io}) , \quad (5.5)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн.;

$K_{т.з}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

Ц_i – ціна i -го найменування матеріалу, напівфабрикату і комплектуючого, грн.;

N_i – потреба в i -му матеріалі, напівфабрикаті та комплектуючому;

Ц_{io} – ціна зворотних відходів i -го найменування матеріалу, грн;

N_{io} – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів і комплектуючих.

Отримуємо:

$$Ц_{io} = 0; N_{io} = 0; K_{т.з} = 0;$$

$$M = (1 + 0,05) \cdot (5+50+150) = 215,25 \text{ грн.}$$

Разом, витрати на матеріали становлять 215,25 грн.

5.2.5 Витрати на спеціальне обладнання

У цю статтю входять витрати на придбання, транспортування, монтаж і налагодження нестандартного обладнання.

Практично, в даному випадку, в цій статті враховуються витрати на оплату машинного часу ЕОМ для написання і налагодження даної програми. Для цього необхідно скласти кошторис «витрат на утримання та експлуатацію обладнання» виходячи з якої визначиться вартість однієї машино-години роботи ПК, після множення якої на машинний час, що був витрачений на написання і налагодження програми, отримаємо витрати на оплату машинного часу.

Амортизаційні відрахування визначають за формулою (5.6):

$$A = \Phi_6 \cdot \frac{H_A}{1000}, \quad (5.6)$$

де Φ_6 – балансова вартість обчислювальної техніки, грн.;

H_A – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки, %.

Балансова вартість обчислювальної техніки становить 8500 грн.

Отримуємо:

$$A = 8500 \cdot 0,25 = 2125 \text{ грн.}$$

Статтю «Експлуатація обладнання» розраховують підсумовуванням витрат на електроенергію і допоміжні матеріали.

$$C_3 = N_H \cdot \Phi_{эф} \cdot K_{зв} \cdot K_{зм} \cdot Ц_3, \quad (5.7)$$

де N_H – номінальна потужність ЕОМ, кВт;

$\Phi_{эф}$ – річний ефективний фонд часу роботи ЕОМ, машино–год;

$K_{зв}$ – середній коефіцієнт завантаження за часом;

$K_{зм}$ – коефіцієнт завантаження по потужності;

$Ц_3$ – ціна одного кВт · год електроенергії, грн / (кВт · год).

Номінальна потужність ЕОМ - 0,3 кВт. Річний ефективний фонд часу роботи ЕОМ становить 1800 годин. Середні коефіцієнти завантаження за часом і за потужністю рівні відповідно 0,9 і 0,6. Ціна однієї кіловат-години електроенергії складає 2,11 грн.

Отримуємо:

$$C_e = 0,2 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 2,11 = 615,276 \text{ грн.}$$

Зарплата обслуговуючого персоналу розраховується за формулою (5.8):

$$ЗП_{обсл} = \Phi ЗП_{г} \cdot (1 + K_{відрах.}) \cdot \frac{t_{обсл}}{\Phi_{эф.обсл}} \cdot \frac{S_{рп}}{\Gamma_c} + E_{зп}, \quad (5.8)$$

де $\Phi ЗП_{г}$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн.;

$K_{відрах}$ – коефіцієнт, що враховує відрахування на соціальне страхування і в інші фонди;

$t_{\text{обсл}}$ – час протягом року, необхідне технічне обслуговування ЕОМ, годин / рік;

$\Phi_{\text{эф.оюсл}}$ – річний ефективний фонд часу обслуговуючого персоналу, годин / рік.

Місячна заробітна плата обслуговуючого персоналу становить 3723 грн., а річний фонд заробітної плати відповідно дорівнює 44676 грн. Річний ефективний фонд робочого часу обслуговуючого ПК працівника дорівнює 1750 год/рік. На обслуговування одного ПК витрачається по 1 годині на місяць, що в рік становить 12 годин.

Отримуємо:

$$ЗП_{\text{обсл}} = 44676 \cdot (1+0,22) \cdot 12/1750 = 373,95 \text{ грн.}$$

Стаття «Поточний ремонт обладнання» приймається рівною 3% від балансової вартості обладнання і складає 255 грн. Стаття «Інші витрати» приймається рівною п'яти відсоткам від суми всіх попередніх статей витрат на утримання і експлуатацію обладнання. Сума всіх попередніх статей дорівнює 3369,23 грн., 5% від суми складають 168,46 грн. Розраховані статті витрат на утримання і експлуатацію обладнання внесені в табл. 5.4.

Таблиця 5.4 - Кошторис витрат на утримання і експлуатацію обладнання

Найменування статей витрат	Сума, грн.
Амортизація обладнання	2125
Експлуатація обладнання (крім витрат на поточний ремонт)	615,28
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальні заходи	373,95
Поточний ремонт обладнання	255
Інші витрати	168,46
Разом	3537,69

Витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів визначаються за формулою (5.9):

$$C_{\text{мо}} = P_{\text{екс}} \cdot t_{\text{мо}}, \quad (5.9)$$

де $C_{\text{мо}}$ – витрати на оплату машинного часу, грн;

$P_{\text{екс}}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн. / машино–год.;

$t_{\text{мо}}$ – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино–год.

Експлуатаційні витрати на одну годину машинного часу використовуваної ЕОМ розраховують діленням суми витрат за кошторисом «Витрати на утримання та експлуатацію обладнання (ЕОМ)» (табл. 5.4) на річний ефективний фонд часу роботи ЕОМ. Річний ефективний фонд часу роботи ЕОМ дорівнює 1800 годин [22]. В результаті експлуатаційні витрати на одну годину машинного часу рівні:

$$P_{\text{екс}} = 3537,69/1800 = 1,97 \text{ грн. / машино–год.}$$

ЕОМ експлуатується 64 дні в одну зміну, що становить в сумі 512 годин. Таким чином, витрати на оплату машинного часу складуть:

$$C_{\text{мо}} = 1,97 \cdot 512 = 1008,64 \text{ грн.}$$

5.2.6 Інші прямі витрати

В інші прямі витрати включаються витрати на комерційне програмне забезпечення (ПЗ), що використовується при розробці системи, які розраховуються за формулами (5.10) і (5.11):

- дольове ПЗ, що використовується постійно при роботі ПК (Windows 7 Home basic) - 850 грн. без ПДВ;

- цільове ПЗ, що купується для даного конкретного завдання (Microsoft Visual Studio 2015 Professional) - 4345 грн. без ПДВ.

$$S_{\text{дол.ПЗ}} = \frac{Ц_{\text{ПЗ.Windows}} \cdot T_{\text{КТС}}}{\Phi_{\text{еф.КТС}} \cdot T_{\text{с.ПЗ}}}, \quad (5.10)$$

$$S_{\text{цел.ПЗ}} = Ц_{\text{ПЗ.vs}}, \quad (5.11)$$

де $S_{\text{дол.ПЗ}}$ - витрати на дольове ПЗ, грн .;

$S_{\text{цел.ПЗ}}$ - витрати на цільове ПЗ, що купується виключно для написання ПЗ, що розробляється, грн .;

$Ц_{\text{ПЗ.Windows}}$ - ціна ПЗ Windows (без ПДВ), грн;

$Ц_{\text{ПЗ.vs}}$ - ціна ПЗ Microsoft Visual Studio 2015 Professional (без ПДВ), грн;

$T_{\text{КТС}}$ - машинний час КТС, необхідний користувачеві для розробки системи, машино-год / рік;

$\Phi_{\text{еф.КТС}}$ - річний ефективний фонд часу роботи КТС, машино-год / рік;

$T_{\text{с.ПЗ}}$ - термін служби дольове ПЗ, років.

Отримуємо:

$$S_{\text{дол.ПО}} = \frac{850 \cdot 512}{1800 \cdot 5} = 48,36 \text{ грн.}$$

$$S_{\text{цел.ПО}} = 4345 \text{ грн.}$$

Та розрахуємо загальну сумму:

$$S_{\Sigma} = 49,11 + 4345 = 4393,36 \text{ грн.}$$

5.2.7 Розрахунок накладних витрат

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і т.д.), утримання та експлуатацію будівель. В даному випадку накладні витрати становлять 40% до основної заробітної плати розробників, що складає 12000 грн.

Результати визначення витрат на розробку програми у вигляді калькуляції кошторисної вартості робіт наведені в табл. 5.5.

Таблиця 5.5 - Калькуляція кошторисної вартості робіт по розробці програми

№	Найменування статей витрат	Сума, грн	Питома вага до підсумку, %
1	Основна заробітна плата	30000,00	51,834
2	Додаткова заробітна плата	3000,00	5,183
3	ЄСВ	7260	12,54
4	Матеріали та комплектуючі	215,25	0,372
5	Витрати на спец. обладнання	1008,64	1,743
6	Інші прямі витрати	4393,36	7,59
7	Накладні витрати	12000	20,734
8	Разом	57877,25	100,00

5.3 Розрахунок економічної ефективності програмного продукту

Програма дозволяє скоротити час і трудомісткість процесу прийняття рішень шляхом генерації порад проєктувальнику. Знайдені в результаті роботи програми рішення будуть обґрунтованими з точки зору вимог ергономіки, що може гарантувати отримання найбільш оптимального результату.

Економічна ефективність програми досягається за рахунок скорочення часу розрахунків на ЕОМ з 1000 до 450 годин на рік і автоматизації додаткової

роботи по обслуговуванню аналітика, що в сумі призводить до економії заробітної плати. Річна економія по заробітній платі з урахуванням нарахувань становить за формулою (5.12):

$$E_{зп} = T_e \cdot C_{зп} \cdot (1 + K_{отч}) \cdot (1 + K_{доп}), \quad (5.12)$$

де $E_{зп}$ - річна економія по заробітній платі з відрахуваннями на соціальні заходи, грн.;

$C_{зп}$ - заробітна плата за одну годину роботи, грн.;

T_e - час який економиться при впровадженні програмного засобу;

$K_{доп}$ - коефіцієнт, що враховує додаткову заробітну плату.

В результаті отримуємо:

$$E_{зп} = 550 \cdot 50 \cdot (1 + 0,22) \cdot (1 + 0,11) = 36905 \text{ грн.}$$

Річна економія визначається за формулою (5.13):

$$E = (T_{M_1} - T_{M_2}) \cdot B_{екс.ктз} + E_{\kappa}^{\text{а}} \cdot \frac{\Phi_{б.ктз} (T_{M_1} - T_{M_2})}{\Phi_{эф.ктз}} - \frac{S_{рп}}{T_c} + E_{зп}, \quad (5.13)$$

де $T_{M_1} - T_{M_2}$ - машинний час, необхідний для розв'язання поставлених завдань, відповідно в старому та у новому варіантах, машино-год./рік;

$B_{екс}$ - експлуатаційні витрати, що доводяться на 1 год машинного часу ЕОМ, грн / машино-год.;

$E_{\kappa}^{\text{а}}$ - нормативний коефіцієнт ефективності капітальних вкладень у засоби автоматизації;

$\Phi_{б.ктз}$ - балансова вартість ЕОМ (КТЗ), грн.;

$\Phi_{эф.ктз}$ - річний ефективний фонд часу роботи ЕОМ, машино-год.;

$S_{рп}$ - сумарні витрати на розробку програми, грн.;

T_c – термін служби впроваджуваної програми до її морального зношування, років. Приймаємо цей термін 5 років;

$E_{зп}$ – економія на заробітній платі.

Розраховуємо:

$$E = (1000 - 450) \cdot 1,97 + 0,2 \cdot (8500 \cdot (1000 - 450)) / 1800 - 57877,25/3 + 36905 = 19215,53 \text{ грн.}$$

Коефіцієнт ефективності капітальних вкладень E_ϕ і термін їх окупності $T_{ок}$ розраховуються за формулами (5.14) та (5.15):

$$E_\phi = \frac{E}{S_{рп}}, \quad (5.14)$$

$$T_{ок} = \frac{S_{рп}}{E}. \quad (5.15)$$

Таким чином:

$$E_\phi = 19215,53 / 57877,25 = 0,33,$$

$$T_{ок} = 57877,25 / 19215,53 = 3,01 \text{ років.}$$

$$T_o = 57877,25 / 19215,53 = 3,012 \text{ років.}$$

Термін окупності програми становить 3,01 року, що менше нормативного терміну окупності. Коефіцієнт ефективності капітальних вкладень дорівнює 0,33, що більше нормативного (0,2). Річна економія при використанні програми становить 19215,53 грн. Таким чином, впровадження програмного продукту є економічно доцільним.

Техніко-економічні показники наведені в табл. 5.6.

Таблиця 5.6 - Техніко-економічні показники

Показники	одиниця виміру	варіанти	
		базовий	новий
1. Витрати на розробку програми	грн.		57877,25
2. Експлуатаційні витрати	грн.	38507,94	19292,42
3. Річна економія	грн.		19215,53
4. Термін окупності програми	років		3,012
5. Тривалість розрахунків		більше	менше
6. Якість ЛМІ		менше	більше

6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1 Аналіз потенційних небезпек.

У дипломній кваліфікаційній роботі магістра проводиться дослідження і програмна реалізація методів інтелектуальної підтримки проектувальників інтерфейсів SCADA-систем.

При роботі постійно або періодично діють наступні небезпечні і шкідливі чинники.

Невідповідність ергономічних характеристик устаткування нормованим величинам внаслідок нераціональної організації робочого місця.

Негативний вплив при роботі з відео дисплейними терміналами на функціональний стан людини який обумовлений електростатичними полями, які створюються моніторами з електронно-променевою трубкою, що може призвести до зниження імунітету, захворювань периферичної нервової системи або бронхо-легеневих та шкіряних захворювань.

Можливість ураження електрострумом внаслідок небезпечного рівня напруги в електричному ланцюзі, тому у приміщенні можуть виникнути аварійні ситуації: замикання, загоряння проводки, поразка електрострумом, що може привести до електротравм або летальних наслідків.

Негативний вплив випромінювань відео дисплейних терміналів на органи зору людини має основною причиною перенапруження м'язового апарату ока, що пов'язане з недостатньою контрастністю та мерехтінням зображення, відблисками та відбитим світлом від моніторів. Крім того, втома очей настає при частій зміні огляду більш освітленого монітора та менш освітленої клавіатури, що призводить до «астенопії» (відчуття дискомфорту та біль в очах).

Статична електрика.

Невідповідність нормам параметрів мікроклімату.

Незадовільні умови освітлення робочого місця, в наслідок не рівномірного розподілу освітлювальних приладів.

Небезпеки, пов'язані з підвищеним рівнем шуму, який створюється устаткуванням та людьми, що знаходяться у приміщенні, і які призводить до погіршення слуху.

Можливість загорянь внаслідок порушень правил пожежної безпеки, що може призвести до пожеж.

Неправильні дії персоналу під час НС внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків.

6.2 Заходи із забезпечення безпеки

Забезпечення безпеки праці, організації оптимальних умов праці, а також захисту навколишнього середовища досягається шляхом дотримання норм і проведення заходів, передбачених законодавчими актами України.

У розділі 4 ДСанПіН 3.3.2.007-98 «Гигиенические требования к организации работы с визуальными дисплейными терминалами электронно-вычислительных машин» [23] передбачені гігієнічні вимоги до організації устаткування робочих місць ВДТ ЕОМ і ПЭВМ. Конструкція робочого стола повинна відповідати сучасним вимогам ергономіки й забезпечувати оптимальне розміщення на робочій поверхні використовуваного устаткування (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого стола із ВДТ повинні регулюватися в межах 680...800 мм, а ширина й глибина - забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600...1400 мм, глибина - 800...1000 мм). Приймаємо робочу поверхню з розмірами 1400x800. Щоб уникнути відблисків на екрані вікон доцільно розмістити робоче місце торцем до вікна не ближче чим на 1 м. Площа дозволяє нам розмістити додаткові меблі для зберігання документів, магнітних дисків, полками, стелажми, тумбами й т.п. [24]

Велике значення надається ергономічним та естетичним вимогам. Цього домоглися за допомогою колірної рішення приміщення й устаткування з урахуванням вимог ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для

виконання робіт у положенні сидячи. Загальні ергономічні вимоги» [25] - у приміщеннях стіни пофарбовані фарбами пастельних тонів.

Під час організації робочого простору необхідно враховувати індивідуальні антропометричні параметри користувача з відповідними допусками на можливі зміни робочих поз та потребу у переміщеннях.

Заходи мінімізації негативного впливу випромінювань від відео дисплейних терміналів на органи зору.

Передбачено такі внутрішньозмінні режими праці та відпочинку при роботі з ЕОМ при 8-годинній денній робочій зміні в залежності від характеру праці:

- для розробників програм із застосуванням ЕОМ слід призначати регламентовану перерву для відпочинку тривалістю 15 хвилин через кожну годину роботи за ВДТ;

- для операторів із застосування ЕОМ слід призначати регламентовані перерви для відпочинку тривалістю 15 хвилин через кожні дві години;

- для операторів комп'ютерного набору слід призначати регламентовані перерви для відпочинку тривалістю 10 хвилин після кожною години роботи за ВДТ.

У всіх випадках, коли виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи з ВДТ не повинна перевищувати 4 години.

При 12-годинній робочій зміні регламентовані перерви повинні встановлюватися в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4-х годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин [23].

З метою зменшення негативного впливу монотонності є доцільним застосовувати чергування операцій усвідомленого тексту і числових даних (зміна змісту роботи).

Для зниження нервово-емоційного напруження, втомлення зорового аналізатору, поліпшення мозкового кровообігу, подолання несприятливих

наслідків гіподинамії, запобігання втомі доцільні деякі перерви використовувати для виконання комплексу вправ.

В окремих випадках - при хронічних скаргах працюючих з ВДТ ЕОМ і ПЕОМ на зорове втомлення, незважаючи на дотримання санітарно-гігієнічних вимог до режимів праці і відпочинку, а також застосування засобів локального захисту очей - допускаються індивідуальних підхід до обмеження часу робіт з ВДТ, зміни характеру праці, чергування з іншими видами діяльності, не пов'язаними з ВДТ [23].

Активний відпочинок має полягати у виконанні комплексу гімнастичних вправ, спрямованих на зняття нервового напруження, м'язове розслаблення, відновлення функцій фізіологічних систем, що порушуються протягом трудового процесу, зняття втоми очей, поліпшення мозкового кровообігу і працездатності.

Згідно ПУЕ приміщення виробництва відноситься до категорії «Приміщення без підвищеної небезпеки» [26].

Апаратура, що перебуває під небезпечною для життя напругою, припускає забезпечення електробезпечності. Всі металеві частини повинні бути заземлені на третій провід спеціальних триконтактних розеток. Струмопровідні кабелі повинні бути розташовані під дерев'яною підлогою, зверху покритою лінолеумом, або у вигляді схованої проводки в стінах, таким чином, контакт із кабелем виключається. Над кожною розеткою повинен бути попереджувачий надпис.

Електрична ізоляція забезпечується конструкціями пристроїв і способами їхнього підключення. У процесі експлуатації періодично проводиться перевірка якості ізоляції, шляхом виміру її опору. Відповідно до ПУЕ в електроустановках до 1000В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менш 0,5 МОм.

Відповідно до ПУЕ захисне заземлення 4 Ом застосовується у всіх електроустановках змінного струму під напругою 220 В і вище й постійного

струму 440В і вище. У приміщеннях з підвищеною небезпекою заземлення використовується в електроустановках змінного струму напругою 42 В та постійного струму 110В. Захисному заземленню підлягають корпуси трансформаторів, оболонки кабелів, металеві огороження електроустановок.

Вирівнювання потенціалів досягаються шляхом устрою контурних заземлювачів. У приміщеннях вирівнювання потенціалів здійснюється з'єднанням металевих стоек ЕОМ, корпусів апаратів світильників і іншого устаткування з усіма доступними для дотику металевими конструкціями будинку, спорудження, а також з контуром шиною захисного заземлення.

Електричний поділ мережі припускає розділ мережі на окремі електрично непов'язані між собою ділянки за допомогою розділових трансформаторів. Підвищення електробезпечності відбувається за рахунок зменшення ємкості мережі й підвищення опору ізоляції фаз мережі щодо землі.

Подвійна ізоляція забезпечує застосування крім основної, робочої ізоляції струмоведучих частин ще одного шару додаткової ізоляції, що ізолює людину від металевих неструмоведучих частин, які можуть випадково виявитися під напругою. При застосуванні електроустаткування з подвійною ізоляцією не потрібно ні заземлення, ні занулення їхніх корпусів [27].

Накопичена статична електрика, зокрема, на екрані монітора притягує пил, бруд та інші частини присутні в повітрі. Причому електризується не тільки екран, а і повітря на робочому місці, а також одяг користувача, особливо якщо він з синтетичного та шерстяного матеріалу.

В якості заходів захисту запропоновано:

- кілька разів на протязі робочого дня мити руки і обличчя водою, а після закінчення роботи вимити руки й лице з милом;
- щоденно протирати екран монітора, клавіатуру, пристрій «миша», а якщо є приєднаний фільтр, то і його антистатичною серветкою;
- щоденно в приміщенні з ПК проводити вологе прибирання;
- установити нейтралізатори статичної електрики;

- підтримувати у приміщенні вологість повітря зазначену в нормативних документах;
- користувачу ПК бажано носити одяг з природних (льняних) волокон, в т.ч. і шкарпетки [23].

6.3 Заходи з виробничої санітарії та гігієни праці

Оптимальні норми температури повітря в робочій зоні виробничих приміщень у холодний і перехідний періоди року при II категорії робіт згідно ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [28] – 20...23°C, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек. У теплий період року температура – 22...25°C, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек.

Для забезпечення вищевказаних параметрів передбачене застосування вентиляції - природної (аерації й провітрювання), механічної (загальобмінної, місцевої приточної і витяжної), кондиціювання повітря.

У приміщенні забезпечений приплив свіжого повітря, кількість якого визначається техніко-економічним розрахунком і вибором схеми системи вентиляції. Мінімальна витрата повітря визначається з розрахунку 50...60 м³/ч на одну людину, але не менш дворазового повітрообміну в годину.

При невеликому забрудненні зовнішнього повітря кондиціювання приміщень здійснюється із перемінними витратами зовнішнього повітря й рециркуляційного.

Всі норми по освітленості беруться згідно ДБН В.2.5-28-20018 «Природне та штучне освітлення» [29]. Штучне освітлення в приміщеннях здійснюється системою загального рівномірного освітлення. При цьому штучне освітлення застосовують не тільки в темний, але й у світлий час доби.

Найбільш прийнятними для приміщень є люмінесцентні лампи ЛБ (білого світла) потужністю 20, 40 або 80 Вт, які встановлюються в світильники типу

ЛСП. Допускається застосування ламп накаливання у світильниках місцевого освітлення.

Загальне освітлення варто виконувати у виді суцільних чи переривчастих ліній світильників, розташованих збоку від робочих місць.

Розрахуємо систему загального рівномірного штучного освітлення в адміністративному приміщенні [30]. Приймаємо розміри приміщення довжина $A = 11$ м, ширина $B = 7$ м, висота $H = 5$ м. Висота робочої поверхні складає $h = 0,7$ м.

Коефіцієнти відбиття поверхонь приміщення: колір стелі, стін - сірий, незначне пиловиділення, $\rho_c = 50\%$; $\rho_{ст} = 30\%$; $\rho_{п} = 10\%$.

Розряд зорової роботи – III б.

Згідно з ДБН В.2.5-28-2018 «Природне і штучне освітлення. Норми проектування» рівень нормованого загального освітлення в приміщенні нормуються як $E_n = 400$ Лк.

Обираємо світильник типу ЛПП, призначений для приміщень висотою до 6 м. IP=65 – ступінь захисту світильника, $[L/h] = 1,4$ – числове значення коефіцієнта світильника.

Коефіцієнт запасу k_z враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо $k_z = 1,6$.

Коефіцієнт нерівномірності (мінімального) освітлення z (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює $z = 1,1$.

Розрахуємо кількість рядів світильників у приміщенні за формулою (6.1):

$$N_p = \frac{B}{(H - h_p) * [L/h]} \quad (6.1)$$

Отримаємо:

$$N_p = \frac{7}{(5 - 0,7) \cdot 1,4} = 1,16 \approx 2 \text{ ряди світильників}$$

Визначимо максимально припустиму відстань між рядами світильників за формулою (6.2):

$$L_{\max} = \frac{B}{N_p}, \quad (6.2)$$

де B – ширина приміщення, N_p – кількість рядів, розрахунок:

$$L_{\max} = \frac{7}{2} = 3,5 \text{ м.}$$

Розраховуємо висоту підвісу світильника над робочою поверхнею за формулою (6.3):

$$h = \frac{L_{\max}}{[L/h]}. \quad (6.3)$$

Отримаємо результат:

$$h = \frac{3,5}{1,4} = 2,5 \text{ м.}$$

Знайдемо висоту звисання світильника від стелі за формулою (6.4):

$$h_3 = H - h_p - h, \quad (6.4)$$

де H – висота приміщення, h_p – висота робочої поверхні, h – висота підвісу світильника над робочою поверхнею.

$$h_3 = 5 - 0,7 - 2,5 = 1,8 \text{ м.}$$

Чисельне значення індексу приміщення визначаємо за рівнянням за формулою (6.5):

$$i = \frac{AB}{h(A+B)}, \quad (6.5)$$

де A – довжина приміщення, $м$;

B – ширина приміщення, $м$;

h – висота розміщення світильників над робочою поверхнею, $м$.

Розрахуємо значення індексу:

$$i = \frac{11 \cdot 7}{2,5 \cdot (11 + 7)} = 1,71$$

Значення коефіцієнта використання світлового потоку η вибирається в залежності від виду джерела світла, типу обраного світильника, коефіцієнтів відбиття поверхонь приміщення (ρ_c , ρ_{st} , ρ_p) та індексу приміщення. В нашому випадку $\eta = 54\%$.

Визначаємо сумарний світловий потік освітлювальної установки у даному виробничому приміщенні за формулою (6.6):

$$\Phi_{\Sigma} = \frac{E_n \cdot S \cdot k_z \cdot z}{\eta}, \quad (6.6)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, $лм$;

E_n – нормоване значення освітленості, $лк$;

S – площа освітлюваної поверхні, $м^2$;

k_z – коефіцієнт запасу;

z – коефіцієнт нерівномірності (мінімальної) освітленості;

η – коефіцієнт використання світлового потоку.

Знайдемо розрахункове значення сумарного світлового потоку:

$$\Phi_{\Sigma} = \frac{400 \cdot 11 \cdot 7 \cdot 1,6 \cdot 1,1}{0,54} = 100385 \text{ лм}$$

Визначаємо умовну загальну кількість світильників у приміщенні за формулою (6.7):

$$N^* = AB/L_{\max}^2. \quad (6.7)$$

Визначимо кількість світильників:

$$N^* = \frac{11 \cdot 7}{3,5 \cdot 3,5} = 6,3 \approx 8 \text{ шт.}$$

Розрахуємо світловий потік умовного джерела світла за формулою (6.8):

$$\Phi_{\text{л}}^* = \frac{\Phi_{\Sigma}}{N_{\text{л}}} \text{ лм}, \quad (6.8)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм;

$N_{\text{л}}$ – загальна кількість ламп у приміщенні шт., за формулою (6.9):

$$N_{\text{л}} = N^* \cdot n, \quad (6.9)$$

де n – кількість ламп у світильнику (в ЛПП світильнику знаходиться 2 лампи).

Розрахуємо загальну кількість ламп у приміщенні:

$$N_{\text{л}} = 8 \cdot 2 = 16 \text{ шт.}$$

Тоді,
$$\Phi_{\text{л}} = \frac{100385}{16} = 6274 \text{ лм.}$$

Обираємо тип стандартної лампи з найближчим значенням фактичного світлового потоку лампи $\Phi_{\text{л}}$ і знайти коефіцієнт m (співвідношення між

розрахунковим світловим потоком лампи Φ_l^* та фактичним світловим потоком вибраної стандартної лампи Φ_l) за формулою (6.10):

$$m = \frac{\Phi_l^*}{\Phi_l}. \quad (6.10)$$

Обираємо лампу та проведемо розрахунок за формулою ЛБ із $P_l=40$ Вт, $\Phi_l = 5800$ лм.

$$m = \frac{6274}{5800} = 1,081.$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні за формулою (6.11):

$$N = N \cdot m, \quad (6.11)$$

Підрахуємо оптимальну кількість світильників:

$$N = 8 \cdot 1.081 \approx 8 \text{ шт.}$$

Фактична кількість ламп у приміщенні за формулою (6.12):

$$N_{\text{фл}} = N_{\text{ф}} \cdot n, \quad (6.12)$$

Підрахуємо кількість ламп у приміщенні:

$$N_{\text{фл}} = 8 \cdot 2 = 16 \text{ шт.}$$

Визначаємо загальну розрахункову освітленість E_p у приміщенні, що створюється при застосуванні стандартних ламп, за формулою (6.13):

$$E_p = \frac{\Phi_{\text{л}} N_{\text{фл}} \eta}{Sk_3 z} \text{ лк. ,} \quad (6.13)$$

Підрахуємо загальну розрахункову освітленість:

$$E_p = \frac{5800 \cdot 16 \cdot 0,54}{11 \cdot 7 \cdot 1,6 \cdot 1,1} = 370 \text{ лк.}$$

Тобто виконується умова $E_p = (-10\% \dots +20\%) \cdot E_n \text{ лк.}$

Таким чином, на основі розрахунків для освітлення приміщення площею 77 м² встановлено 8 світильників, розташованих у 2 рядах по 4 світильники у кожному ряді, що дозволяє отримати освітленість на робочому місці 400 лк відповідно до ДБН В.2.5-28-2018 «Природне і штучне освітлення. Норми проектування».

Згідно з ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» [30] зниження сильного шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму проникаючого ззовні, є дуже важливим завданням. Зниження рівню шуму в джерелі виникнення забезпечується застосуванням пружних прокладок між основою машини, приладу й опорною поверхнею. Як прокладки зазвичай використовуються гума, повсть, пробка, різної конструкції амортизатори. Під настільні шумливі апарати можна підкладати м'які килимки із синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, - прокладки з м'якої гуми, повсті, товщиною 6 - 8 мм.

6.4 Заходи безпеки у надзвичайних ситуаціях

6.4.1 Заходи з пожежної безпеки

Заходи з протипожежної безпеки розроблені для виробничого приміщення обладнаного ПК з ВДТ розроблені згідно вимог НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні» [31].

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні, обладнаному ПК з ВДТ:

- відповідно до вимог ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою» [32], воно належить до категорії «Д» з пожежної небезпеки – простір у приміщенні, у якому перебувають тверді горючі речовини та матеріали, у приміщенні, обладнаному ПК з ВДТ можлива пожежа класу – А та Е;.

Оскільки приміщення належить до категорії «Д» з пожежної небезпеки, тому відповідно до вимог ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» [33] воно має II ступінь вогнестійкості.

У разі виникнення пожежі у приміщенні обладнаному ПК з ВДТ для евакуації персоналу відповідно до вимог ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» передбачені виходи, по обидві сторони приміщення, з одного боку вікно (на пожежні сходи), а з іншого – вхідні двері.

Обладнання, силові та освітленні мережі приміщення, обладнаного ПК з ВДТ відповідають вимогам пожежної безпеки, оскільки виконані відповідно до вимог НПАОП 40.1-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок» [34], та мають ступінь захисту ізоляції обладнання IP44 яка відповідає класу пожежанебезпечної зони П-Па до якої належить приміщення.

Для виявлення початкової стадії загоряння, швидкого й точного оповіщення служби пожежної охорони про час і місце виникнення пожежі встановлені системи автоматичної пожежної сигналізації (АПС). Вони можуть самостійно пускати в хід установки пожежогасіння, коли пожежа не досягла великих розмірів. Системи АПС складаються з пожежних сповіщувачів, ліній зв'язку й прийомних пультів (станцій). Згідно вимог ДБН В.2.5-56:2014 «Системи протипожежного захисту», в приміщенні обладнаному ПК з ВДТ встановлена система пожежної й охоронної сигналізації

«Фотон-ОФІС», яка забезпечує виявлення теплових і димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика.

Оскільки приміщення (дослідницької лабораторії, конструкторського бюро, тощо) що обладнане ПК з ВДТ має площу 77 м², тому відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» [35] для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 4 штуки (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг. і більше, на 20 м² площі приміщення). Відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м.

6.4.2 Заходи з цивільного захисту

Укриття населення у захисних спорудах цивільного захисту недотримання якого може призвести до механічних травм, зараження хімічними викидами або летального наслідку.

Укриття населення в захисних спорудах — це комплекс заходів із завчасним будівництвом захисних споруд, а також пристосуванням наявних приміщень для захисту населення та підтримання їх у готовності до використання [36].

Укриттю в захисних спорудах у надзвичайних ситуаціях підлягає все населення України. Фонд захисних споруд створюється шляхом обстеження й обліку підземних та наземних будівель і споруд, що відповідають вимогам захисту населення; дообладнання з урахуванням реальної обстановки підвалів, погребів та інших заглиблених приміщань; обстеження і взяття на облік підземних і наземних будівель та споруд гірничих виробок і природних порожнин, що відповідають вимогам захисту; у разі необхідності переобладнання цих приміщень; будівництво заглиблених споруд пристосованих для захисту, що окремо розташовані від об'єктів виробничого

призначення; масового будівництва в період загрози надзвичайних ситуацій найпростіших сховищ та укриттів; будівництво окремих сховищ та протирадіаційних укриттів.

Центральний орган виконавчої влади, до компетенції якого належать питання захисту населення і територій від надзвичайних ситуацій визначає перелік сховищ, протирадіаційних укриттів та інших захисних споруд, які необхідно будувати, що затверджується Кабінетом Міністрів України.

Потреби в захисних спорудах визначають, виходячи з необхідності укриття всіх працюючих за місцем роботи і проживання, усього непрацюючого населення за місцем проживання.

Укриття населення в захисних спорудах є надійним способом захисту від уражаючих факторів ядерної, хімічної, бактеріологічної, звичайної зброї, у разі аварій і деяких стихійних лих (ураганів, снігових заносів).

Захисні споруди за своїм призначенням і захисними властивостями поділяються на сховища, протирадіаційні укриття (ПРУ) і найпростіші укриття - щілини.

Сховища і протирадіаційні укриття будують завчасно, вони мають подвійне призначення: для потреб об'єктів народного господарства (навчальні класи, для спортивних секцій та ін.) і укриття населення.

Сховища це інженерні споруди, які забезпечують надійний захист людей від усіх уражаючих факторів ядерного вибуху, отруйних і СДЯР, бактеріальних засобів і уражаючих факторів звичайної зброї, обвалів і уламків зруйнованих будівель і споруд.

Класифікуються вони за захисними властивостями, місткістю, місцем розміщення, забезпеченням фільтровентиляційним обладнанням і часом побудови.

За захисними властивостями (від дії вибухової хвилі) сховища поділяються на п'ять класів.

За місткістю сховища поділяються на: малі — до 150 осіб, середні — від 160 до 450, великі — понад 450 осіб.

За місцем розташування — на вбудовані, які розміщені у підвальних приміщеннях будівель, та окремо побудовані поза будівлями.

За забезпеченням фільтровентиляційним обладнанням — промислового виготовлення і спрощене, виготовлене з підручних матеріалів.

За часом побудови на: побудовані завчасно і швидко споруджені.

Протирадіаційне укриття (ПРУ) — це захисна споруда, яка забезпечує захист у ній людей від радіоактивних речовин і опромінення в зонах радіоактивного забруднення місцевості, отруйних і сильнодіючих ядучих речовин, біологічних засобів у краплинно-рідинному вигляді та світлового випромінення ядерного вибуху, наслідків урагану.

Захисні властивості протирадіаційних укриттів оцінюються коефіцієнтом захисту, який показує, у скільки разів доза радіації на відкритій місцевості на висоті 1 м більша від дози радіації в укритті, тобто коефіцієнт захисту показує, у скільки разів ПРУ послаблює дію радіації, а відповідно і дозу опромінення людей.

Протирадіаційні укриття можуть обладнуватись насамперед у підвальних поверхах будинків і споруд. Підвали в дерев'яних одноповерхових будинках ослаблюють дозу радіації в 7 разів, а в житлових одноповерхових кам'яних (цеглиних) будинках — у 40, у двоповерхових — у 100, середня частина підвалу кількоповерхового кам'яного будинку — у 800—1000 разів. При невисоких рівнях радіації, а також для захисту від бактеріальних засобів, парів отруйних і сильнодіючих ядучих речовин можна використовувати кам'яні (цегляні) або дерев'яні будівлі [37].

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи магістра було виконано такі задачі:

- розглянуто призначення та структуру SCADA-систем;
- досліджено існуючі технології побудови SCADA-систем та обрано для розробки web технологію;
- проаналізовано стан досліджень у області проектування ЛМІ SCADA-систем та обрана базова ІТ, яка дістала подальшого розвитку у даній роботі;
- проведено системний аналіз діяльності проєктувальника інтерфейсів та виявлено номенклатуру проблемних ситуацій, у яких він опиняється під час розробки інтерфейсів SCADA-систем;
- розроблено ІТ інтелектуальної підтримки проєктувальників ЛМІ SCADA-систем для двох типів модальностей – зорової та слухової;
- розроблено у вигляді вирішуючих дерев стратегію кодування технологічної інформації слуховою модальністю;
- розроблено у вигляді UML діаграм моделі програмного забезпечення для автоматизації діяльності проєктувальників ЛМІ SCADA-систем;
- розроблено програмне та інформаційне забезпечення компонентів системи, що інтелектуально підтримує проєктувальника;
- перевірено розроблене програмне забезпечення на прикладі проєктування фрагментів ЛМІ SCADA-системи «Система охолодження».

Наукова новизна результатів дослідження полягає у тому, що дістала подальшого розвитку відома ІТ інтелектуальної підтримки процесу ергономічного проєктування ЛМІ шляхом введення в неї засобів та правил кодування технологічної інформації слуховою модальністю.

Практична цінність роботи полягає в тому, що на підставі досліджень розроблено програмне та інформаційне забезпечення, що забезпечує інтелектуальну підтримку проєктувальника ЛМІ SCADA-систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Russell, Stuart J.; Norvig, Peter (2009). Artificial Intelligence: A Modern Approach (3rd ed.). / Stuart J. Russell, Peter Norvig. – Prentice Hall, Upper Saddle River, New Jersey: Prentice Hall. 1999. – 259 p.
2. Венда, В. Ф. Инженерная психология и синтез систем отображения информации / В. Ф. Венда. – М.: Машиностроение, 1982. — 344 с.
3. Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытание. Справочник / А. Н. Адаменко, А. Т. Ашерев, И. Л. Бердников и др.; под общ. ред. А. И. Губинского. – М.: Машиностроение, 1993. – 528 с.
4. Мунипов, В.М. Эргономика: человекоориентированное проектирование техники, программных средств и среды: Учебник / В. М. Мунипов, В. П. Зинченко. – М.: Логос, 2001. – 356 с.
5. Поспелов, Д. А. Ситуационное управление: теория и практика / Д. А. Поспелов. – М.: Наука. - Гл. ред. физ.-мат. лит., 1986. – 288с.
6. Попов, Э. В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ / Э. В. Попов. – М.: Наука., Гл. ред. физ.-мат. лит., 1987. – 288 с.
7. Kaminska, Z. Intellectual support of control system human-machine interface designers / Kaminska, Z. // Proceedings of Workshop: CEUR, 2019. – P. 241–245.
8. Матвейкин, В. Г. Применение SCADA-систем при автоматизации технологических процессов. / В. Г. Матвейкин, С. В. Фролов, М. Б. Шехтман. – М: Машиностроение, 2000. – 176 с.
9. Сергеев, С. Ф. Введение в инженерную психологию и эргономику иммерсивных сред: Учебное пособие. / С. Ф. Сергеев. – СПб: Изд-во СПбГУ ИТМО, 2011. – 258 с.
10. Лоенен, Э. В. Новые технологии в SCADA-системах для эксплуатации трубопроводов / Э.В. Лоенен, А.В. Петров // ИСУП. – 2010. –

№ 1(25). – С. 17–20.

11. Камінська Ж. К. Аналіз проблем автоматизації процесу ергономічного проектування інформаційних моделей технологічних процесів / Ж. К. Камінська // Вісник ЖДТУ. Серія "Технічні науки". – 2010. – № 1(52). – С. 103–108.

12. А. Т. Ашеро́в, В. В. Сабадаш. Судебно-эргономическая экспертиза несчастных случаев в системах "человек-техника-среда". Монография / А.Т. Ашеро́в, В. В. Сабадаш. – Харьков: УИПА, 2008. – 145 с.

13. Безпечність машин. Ергономічні принципи проектування. Частина 1. Термінологія та загальні принципи: ДСТУ EN 614-1 – 2001 – [Чинний від 2001-12-26]. – К.: Держстандарт України, 2001. – 15 с. – (Національний стандарт України).

14. Галактионов, А. И. Основы инженерно-психологического проектирования АСУ ТП / А.И. Галактионов. – М.: Энергия, 1978. – 208 с.

15. Рыбаков, Ф.И. Системы эффективного взаимодействия человека и ЭВМ / Ф.И. Рыбаков. – М.: Радио и связь, 1985. – 200 с.

16. Полетыкин, А. Г. Исследование и разработка методов и программных средств для информационно-управляющих систем верхнего блочного уровня АСУ ТП АЭС: дис. ... д-ра техн. наук: 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» / А. Г. Полетыкин. – М.: Институт проблем управления РАН, 2008. – 168 с.

17. Сухов, С.А. Разработка системы мониторинга оперативных данных на основе WEB-технологий: дис. ... канд. физ.-мат. наук: 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» / С. А. Сухов. – Протвино: Институт физики высоких энергий, 2007. – 87 с.

18. Сухов, А.А. Программное обеспечение системы повышения качества интерфейса компьютерных тренажерных систем: дис. ... канд. техн. наук: 05.13.11 «Математическое и программное обеспечение вычислительных

машин, комплексов и компьютерных сетей» / А. А. Сухов. – М: Моск. гос. ун-т приборостроения и информатики, 2010. – 132 с.

19. Мнемосхемы авиационные. Общие эргономические требования: ГОСТ 28392-89 – 1991 – [Чинний від 1991-01-01]. – М.: Издательство стандартов, 1990. – 15 с. – (Стандарт СССР).

20. O'Brien, L. ISA-101 Standard Addresses the HMI Lifecycle [Electronic resource] / L. O'Brien. – Access mode: <https://www.arcweb.com/blog/isa-101-standard-addresses-hmi-lifecycle>

21. Сердюк, С. Н. Разработка метода интеллектуальной поддержки процесса эргономического проектирования информационных моделей: дис. ... канд. техн. наук: 05.02.20 «Эргономика» / С. Н. Сердюк. – С-Петербург: СПбЭТУ, 1993. – 364 с.

22. Остапенко, В. В. Методичні вказівки по економічному обґрунтуванню дипломних проєктів для студентів спеціальностей 7.080403 «Програмне забезпечення автоматизованих систем», 7.080402 «Інформаційні технології проектування» і 7.091501 «Комп'ютерні та інтелектуальні системи і мережі» всіх форм навчання / В. В. Остапенко, Е. М. Касьян, Є. М. Сердюк. – Запоріжжя : ЗНТУ, 2009. – 22 с.

23. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. НПАОП 0.00-7.15-18. Державні санітарні правила і норми. / Харків: Форт, 2018 – 48 с.

24. Курс лекцій по «Ергономіці». Ю. М. Мадієвський. – Під ред. д-ра. техн. наук., проф. Березуцкого В.В. – Х.: НТУ «ХПИ», 2015. – 294 с.

25. ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги». Наказ від 21.12.2015 № 204 Про прийняття нормативних документів України, гармонізованих з міжнародними та європейськими нормативними документами, національних стандартів України та змін до національних стандартів України.

26. Правила улаштування електроустановок [Текст]. Міненерговугілля України, 2017. – 617с.

27. Основи професійної безпеки та здоров'я людини : підручник / В. В. Березуцький [та ін.] ; під ред. проф. В. В. Березуцького. – Харків : НТУ «ХПІ», 2018. – 553 с.
28. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. / Харків: Форт, 2000 – 24 с.
29. ДБН В.2.5-28:2018. Природне та штучне освітлення. Посібник: проблеми природного і штучного освітлення. / К: Етін, 2019. – 180 с.
30. Охорона праці: навч. посіб. / З. М. Яремко, С. В. Тимошук, О. І. Третяк, Р. М. Ковтун; за ред. проф. З. М. Яремка. - Львів: Видавничий центр ЛНУ імені Івана Франка, 2010. - 374 с.
31. ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку / МОЗ України, 1999 – 24 с.
32. Правила пожежної безпеки в Україні. 2015 / Харків : Форт, 2015 – 124 с.
33. ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги». / Київ: Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.
34. ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» / Київ Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.
35. НПАОП 40.1-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок». / Харків: Індустрія, 2010. – 118 с.
36. ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги»
37. Стеблюк М. І. Цивільна оборона та цивільний захист [Текст] : навч. посіб. для вузів / М. І. Стеблюк. – К. : Знання, 2013, - 487 с.

ДОДАТОК А
Текст програми

A.1 Текст розробленого застосунку

A.1.1 Фрагмент файлу StateIntellModule.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace IntellSystem
{
    public class StateModule
    {
        private Dictionary<string, Fact> facts;
        public StateModule() {
            facts = new Dictionary<string, Fact>();

            facts.Add("f12", new Fact(12, "Условные знаки"));
            facts.Add("f13", new Fact(13, "Абстрактные геометрические фигуры"));
            facts.Add("f14", new Fact(14, "Число точек или геометрических элементов"));
            facts.Add("f15", new Fact(15, "Тип линии"));
            facts.Add("f16", new Fact(16, "Площадь геометрических фигур"));
            facts.Add("f17", new Fact(17, "Длина линии"));
            facts.Add("f18", new Fact(18, "Ширина линии"));
            facts.Add("f19", new Fact(19, "Ориентация линии в пространстве"));
            facts.Add("f20", new Fact(20, "Позиция"));
            facts.Add("f21", new Fact(21, "Стереоскопическая глубина"));
            facts.Add("f22", new Fact(22, "Штриховка"));
            facts.Add("f23", new Fact(23, "Буквы и знаки пунктуации"));
            facts.Add("f24", new Fact(24, "Математические знаки"));
            facts.Add("f25", new Fact(25, "Цифры"));

            // -----
            facts.Add("f66", new Fact(66, "Синий")); // цвет (синий)
            facts.Add("f67", new Fact(67, "Черный")); // цвет (черный)
            facts.Add("f68", new Fact(68, "Зеленый")); // цвет (зеленый)
            facts.Add("f69", new Fact(69, "Красный")); // цвет (красный)
            facts.Add("f70", new Fact(70, "Оранжевый")); // цвет (оранжевый)
            facts.Add("f71", new Fact(71, "Белый")); // цвет (белый)
            facts.Add("f72", new Fact(72, "Желтый")); // цвет (желтый)
            facts.Add("f73", new Fact(73, "Пурпурный")); // цвет (пурпурный)
            facts.Add("f74", new Fact(74, "Коричневый")); // цвет (коричневый)
            facts.Add("f75", new Fact(75, "Фиолетовый")); // цвет (фиолетовый)
            facts.Add("f76", new Fact(76, "Голубой")); // цвет (голубой)
            facts.Add("f77", new Fact(77, "Серо-голубой")); // цвет (серо-голубой)
            facts.Add("f78", new Fact(78, "Серый")); // цвет (серый)

            // -----
        }
        public StateModule(int stateType) : this() {
```

```

        if (stateType == 1) facts["f87"].Value = true;
        else if (stateType == 2) facts["f86"].Value = true;
        else if (stateType == 3) facts["f88"].Value = true;
    }
    public void ShowShapeInterview() {
        Fact[] f = { facts["f50"], facts["f57"], facts["f58"] };
        FormMultipleChoice form = new FormMultipleChoice("Форма", f);
        if (form.ShowDialog() == DialogResult.OK) {
            ActivateShapeRules();
        }
        form.Dispose();
    }
// -----
    public string GetTextRecommendations() {
        var rec = "Рекомендуемый размер шрифта: 8pt, при этом ширина знаков должна быть
0,6-0,7 от их высоты. ";
        if (facts["f60"].Value) rec += "Рекомендуется использовать полужирный шрифт с
толщиной обводки контура 0,13-0,14 высоты знака.";
        else if (facts["f61"].Value) rec += "Рекомендуется использовать обычный или курсивный
шрифт с толщиной обводки контура 0,08-0,15 высоты знака.";
        return rec;
    }
// -----
    private void ActivateShapeRules() {
        r35();
        r351();
    }
    private void ActivateColorRules() {
        r53();
        r55();
        r61();
        r62();
        r63();
        r64();
        r65();
        r66();
        r67();
    }
// -----
    public Fact[] getCodeSizeFacts() {
        Fact[] codeSizeFacts = { facts["f16"], facts["f17"], facts["f18"] };
        return codeSizeFacts;
    }
// -----
    /// <summary>
    /// ЕСЛИ вид алфавита (форма) (Ф26),
    /// ТО код (форма, условные знаки) (Ф12)
    /// И код (форма, абстрактные геометрические фигуры) (Ф13)
    /// И код (форма, штриховка) (Ф14)

```

```

    /// И код (форма, тип линии) (Ф15)
    /// </summary>
    /// <param name="aFacts"></param>
    private void r4() {
        if (facts["f26"].Value) {
            facts["f12"].Value = true;
            facts["f13"].Value = true;
            facts["f14"].Value = true;
            facts["f15"].Value = true;
        }
    }
}
// -----
    /// <summary>
    /// ЕСЛИ условие (объект, состояние опасное) (Ф88),
    /// ТО цвет (желтый) (Ф72)
    /// </summary>
    /// <param name="aFacts"></param>
    private void r67() {
        if (facts["f88"].Value) {
            facts["f72"].Value = true;
        }
    }
}
}
}

```

A.1.2 Фрагмент файла Parameter.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using xmlSCD.FormDesign.Controls;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;
using System.Xml.Serialization;
using System.Runtime.Serialization;
namespace TechProcess
{
    [DataContract]
    public class Parameter : EncodeSub
    {
        public static readonly string xmlPath = "parameters/";
        public static readonly string xmlAttrUuid = "uuid";
        public static readonly string xmlAttrName = "name";
        public static readonly string xmlAttrValue = "value";
        public static readonly string xmlAttrControl = "controled";
        public static readonly string xmlAttrObserv = "observed";
    }
}

```

```

[DataMember]
public string Value {
    get { return mValue; }
    set {
        // Проверка значения на принадлежность интервалу параметра базового состояния
        if (InBaseRange(value)) {
            mValue = value;
            if (TechProcessObject != null)
                TechProcessObject.SetCurrentState();
            UpdateContainerValue();
        }
        else {
            throw new ArgumentOutOfRangeException("Введенное значение параметра (" +
value + ") находится вне интервала допустимых значений.");
        }
    }
}

[DataMember]
public bool isControlled { get; set; }
[DataMember]
public bool isObserved { get; set; }
public ParameterModule IntellModule { get; set; }
[OnDeserialized]
void OnDeserialized(StreamingContext c) {
    VisualEncoder.Container = new ResMovControl(new TextBox());
    VisualEncoder.Container.ChildControl.BackColor = VisualEncoder.BgColor;
    VisualEncoder.Container.ChildControl.Text = this.ToString();
    VisualEncoder.Container.Tag = VisualEncoder;
    VisualEncoder.TechProcessItem = this;
    var isQn = this is QuantitativeParameter;
    var isQl = this is QualitativeParameter;
    if (IntellModule == null)
        IntellModule = new ParameterModule(isQn, isQl, isControlled, isObserved);
}
public Parameter() {
    VisualEncoder = new VisualEncoder();
    VisualEncoder.Container = new ResMovControl(new TextBox());
    VisualEncoder.TechProcessItem = this;
    SoundEncoder = new SoundEncoder();
}
public override string ToString() {
    return Name + ": " + Value;
}
public virtual bool CheckValue(string value);
public virtual List<string> GetValues();
public virtual void UpdateContainerValue();
public virtual void CreateContainer(Type controlType);
private string mValue;
private bool InBaseRange(string value) {

```

```

        if (TechProcessObject.States == null) {
            return true;
        }
        var baseParam = TechProcessObject.States[(int)StateType.Base].Parameters.Find(p =>
p.UUID == this.UUID);
        if (baseParam != null) {
            return baseParam.CheckValue(value);
        }
        return false;
    }
}
// Количественный параметр
public class QuantitativeParameter : Parameter
{
    public static readonly string xmlTagName = "qnParameter";
    public static readonly string xmlMinValueTagName = "minValue";
    public static readonly string xmlMaxValueTagName = "maxValue";
    public int MinValue { get; set; }
    public int MaxValue { get; set; }
    public QuantitativeParameter() {
    }
    public QuantitativeParameter(string name, string uuid) {
        this.Name = name;
        this.UUID = uuid;
    }
    public QuantitativeParameter(string name, string uuid, int min, int max) : this(name, uuid) {
        this.MinValue = min;
        this.MaxValue = max;
    }
    public override bool CheckValue(string value) {
        bool isInRange = Convert.ToInt32(value) >= MinValue && Convert.ToInt32(value) <=
MaxValue;
        return isInRange;
    }
    public override List<string> GetValues() {
        List<string> values = new List<string>();
        values.Add(this.MinValue.ToString());
        values.Add(this.MaxValue.ToString());
        return values;
    }
    public override void UpdateContainerValue() {
        if (VisualEncoder.Container == null) return;
        var control = VisualEncoder.Container.ChildControl;
        if (control is TextBox) {
            control.Text = this.ToString();
        }
        else if (control is ProgressBar) {
            var progressBar = control as ProgressBar;
            progressBar.Value = Convert.ToInt32(Value);
        }
    }
}

```

```

    }
    else if (control is Chart) {
        var chart = control as Chart;
        chart.Series[0].Points[0].YValues = new Double[] { Convert.ToDouble(Value) };
    }
}

public override void CreateContainer(Type controlType) {
    // Удалить старый контейнер и все его связи
    if (VisualEncoder.Container.Parent != null) {
        VisualEncoder.Container.Parent.Controls.Remove(VisualEncoder.Container);
    }
    VisualEncoder.Container.Dispose();
    // Создать новый контейнер
    try {
        VisualEncoder.Container = new ResMovControl(Activator.CreateInstance(controlType) as
Control);
    }
    catch (Exception e) {
        MessageBox.Show(e.Message);
    }
    // Уникальные настройки для каждого типа контейнера
    var baseParam = TechProcessObject.States[(int)StateType.Base].Parameters.Find(p =>
p.UUID == this.UUID) as QuantitativeParameter;
    if (controlType == typeof(ProgressBar)) {
        var progressBar = VisualEncoder.Container.ChildControl as ProgressBar;
        if (baseParam != null) {
            progressBar.Maximum = baseParam.MaxValue;
            progressBar.Minimum = baseParam.MinValue;
        }
    }
    else if (controlType == typeof(Chart)) {
        var chart = VisualEncoder.Container.ChildControl as Chart;
        chart.ChartAreas.Add(Name);
        chart.Series.Add(new Series(Name));
        // Установка границ по базовому состоянию параметра
        if (baseParam != null) {
            chart.ChartAreas[Name].AxisY.Maximum = (double)baseParam.MaxValue;
            chart.ChartAreas[Name].AxisY.Minimum = (double)baseParam.MinValue;
            chart.Series[0].Points.AddY(Convert.ToDouble(Value));
        }
    }
}
}

// Качественный параметр
public class QualitativeParameter : Parameter
{
    public override bool CheckValue(string value) {
        string foundValue = Values.Find(s => s == value);
        bool isFound = foundValue != null;
    }
}

```

```

        return isFound;
    }
}

```

A.1.3 Фрагмент файла WebApp.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
namespace WebApplication
{
    public partial class WebForm : System.Web.UI.Page
    {
        private static System.Collections.IEnumerable Collection { get; set; }
        private static System.Collections.IEnumerator Inumerator { get; set; }
        private static Mode Mode { get; set; }
        protected void Page_Load(object sender, EventArgs e) {
            if (!Page.IsPostBack) {
                UpdateDB();
                SelectValuesFromDB();
                Mode = new Mode();
                Deserialize();
                CreateModel(mnemoPanel);
            }
        }
        protected void Timer1_Tick(object sender, EventArgs e) {
            Label1.Text = "";
            var row = Inumerator.Current as DataRowView;
            if (row != null) {
                var tick = row.Row.ItemArray[0].ToString();
                do {
                    var obj = row.Row.ItemArray[1].ToString().Trim();
                    var param = row.Row.ItemArray[2].ToString().Trim();
                    var value = row.Row.ItemArray[3].ToString().Trim();
                    Label1.Text += DateTime.Now.ToLongTimeString() + ": " + obj + param;
                    SetParamValue(obj, param, value);
                    if (!Inumerator.MoveNext()) {
                        Inumerator.Reset();
                        Inumerator.MoveNext();
                    }
                } while (true);
                row = Inumerator.Current as DataRowView;
            }
        }
    }
}

```

```

        } while (tick == row.Row.ItemArray[0].ToString());
    }
    CreateModel(mnemoPanel);
    UpdateCharts();
}
private void SelectValuesFromDB() {
    Collection = SqlDataSource1.Select(DataSourceSelectArguments.Empty);
    Inumerator = Collection.GetEnumerator();
    Inumerator.MoveNext();
}
private void Deserialize() {
    var stream = new FileStream(@"SCADA\App_Data\rss3.xml", FileMode.Open);
    var serializeManager = new SerializerManager();
    serializeManager.Deserialize(Mode, stream);
    stream.Dispose();
}
private void UpdateDB() {
    var doc = new XmlDocument();
    doc.Load(@"SCADA\App_Data\log.xml");
    var root = doc.DocumentElement;
    SqlDataSource1.Delete();
    var connection = new SqlConnection(SqlDataSource1.ConnectionString);
    var command = connection.CreateCommand();
    connection.Open();
    foreach (XmlNode tickNode in tickNodes) {
        var time = tickNode.Attributes.GetNamedItem("value").Value;
        var objectNodes = tickNode.ChildNodes;
        foreach (XmlNode objectNode in objectNodes) {
            var objectId = objectNode.Attributes.GetNamedItem("uuid").Value;
            var paramNodes = objectNode.ChildNodes;
            foreach (XmlNode paramNode in paramNodes) {
                var paramId = paramNode.Attributes.GetNamedItem("uuid").Value;
                var paramValue = paramNode.Attributes.GetNamedItem("value").Value;
                var query = "INSERT INTO Ticks (time, object_uuid, param_uuid";
                command.CommandText = query;
                command.ExecuteNonQuery();
            }
        }
    }
    connection.Close();
}
private void CreateModel(Panel panel) {
    foreach (var obj in Mode.Objects) {
        obj.SetCurrentState();
        UpdateImageContainer(obj.CurrentState.VisualEncoder);
        panel.Controls.Add(obj.CurrentState.VisualEncoder.Container);
        foreach (var param in obj.Parameters) {
            param.UpdateContainer();
            panel.Controls.Add(param.VisualEncoder.Container);
        }
    }
}

```

```

    }
  }
}
}

```

A.1.4 Фрагмент файла SerializedControl.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Drawing;
namespace Serialization
{
    public class SerializedControl
    {
        public string TypeName { get; set; }
        public VisualEncoder VisualEncoder { get; set; }
        public Color BackColor { get; set; }
        public Color ForeColor { get; set; }
        public Size Size { get; set; }
        public Point Location { get; set; }
        public string Text { get; set; }
        public Font Font { get; set; }
        public ContentAlignment TextAlign { get; set; }
        public bool Visible { get; set; }
        public Image BackImage { get; set; }
        public Image Image { get; set; }
        public SerializedControl() { }
        public SerializedControl(ResMovControl rscontrol) {
            Location = rscontrol.Location;
            Visible = rscontrol.Visible;
            Size = rscontrol.Size;
            var control = rscontrol.ChildControl;
            TypeName = control.GetType().AssemblyQualifiedName;
            BackColor = control.BackColor;
            ForeColor = control.ForeColor;
            Text = control.Text;
            Font = control.Font;
            TextAlign = ContentAlignment.TopLeft;
            BackImage = control.BackgroundImage;

            if (control is PictureBox) {
                Image = ((PictureBox)control).Image;
            }
        }
    }
}

```

```

    }
    if (control is Label) {
        TextAlign = ((Label)control).TextAlign;
    }
}
public SerializedControl(VisualEncoder visualEncoder) : this(visualEncoder.Container) {
    VisualEncoder = visualEncoder;
}
public ResMovControl GetControl(FormMode form) {
    ResMovControl rscontrol = null;
    var type = Type.GetType(TypeName);
    var control = Activator.CreateInstance(type) as Control;
    if (VisualEncoder != null) {
        if (VisualEncoder.TechProcessItem is Parameter) {
            var param = VisualEncoder.TechProcessItem as Parameter;
            param.CreateContainer(type);
            rscontrol = param.VisualEncoder.Container;
            rscontrol.Tag = VisualEncoder;
        }
        else {
            rscontrol = VisualEncoder.Container;
            rscontrol.SetChild(control);
        }
        ModifyRsControl(rscontrol, form);
        rscontrol.MouseDown += form.ShowProperties;
        rscontrol.MouseMove += form.SetScreenLocation;
        rscontrol.MouseMove += form.UpdateOtherStatesLocation;
        return rscontrol;
    }
    rscontrol = new ResMovControl(control);
    ModifyRsControl(rscontrol, form);
    rscontrol.MouseDown += form.SetActiveControl;
    rscontrol.Tag = new HmiModule();
    return rscontrol;
}
}
}

```

ДОДАТОК Б
Копії слайдів презентації

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ПРОЄКТУВАЛЬНИКІВ ІНТЕРФЕЙСІВ SCADA-СИСТЕМ

РОЗРОБИВ: ст. групи КНТ-119м

Корсунов К.А.

КЕРІВНИК: доцент, к.т.н.

Сердюк С.М.



Рисунок Б.1 – Слайд №1

- **Об'єкт дослідження** – ергономічне забезпечення проєктування інтерфейсів SCADA-систем.
- **Предмет дослідження** – інформаційна технологія інтелектуальної підтримки процесу ергономічного проєктування людино-машинних інтерфейсів SCADA-систем.
- **Мета роботи** – підвищення рівня автоматизації та ефективності роботи проєктувальників інтерфейсів SCADA-систем шляхом розроблення і застосування програмного забезпечення на основі методів і моделей штучного інтелекту.



НУ "ЗП", Кафедра програмних засобів

2

Рисунок Б.2 – Слайд №2

Завдання дослідження

- розглянути призначення та структуру SCADA-систем;
- дослідити існуючі технології побудови SCADA-систем;
- проаналізувати стан досліджень у області проектування ЛІМІ SCADA-систем;
- провести системний аналіз діяльності проєктувальників інтерфейсів та виявити номенклатуру проблемних ситуацій, у яких опиняється проєктувальник інтерфейсів SCADA-систем;
- розробити ІТ інтелектуальної підтримки проєктувальників ЛІМІ SCADA-систем для двох типів модальностей – зорової та слухової;
- розробити стратегію кодування технологічної інформації слуховою модальністю;
- розробити моделі програмного забезпечення для автоматизації діяльності проєктувальників ЛІМІ SCADA-систем;
- розробити програмне та інформаційне забезпечення компонентів системи, що інтелектуально підтримує проєктувальника;
- перевірка розробленого програмного забезпечення на прикладі проєктування фрагментів ЛІМІ SCADA-системи.



НУ "ЗП", Кафедра програмних засобів

3

Рисунок Б.3 – Слайд №3

Узагальнена схема реалізації сучасних АСУТП



НУ "ЗП", Кафедра програмних засобів

4

Рисунок Б.4 – Слайд №4

Реалізація SCADA-системи за допомогою web технології



НУ "ЗП", Кафедра програмних засобів

5

Рисунок Б.5 – Слайд №5

Задачі, що необхідно вирішити при розробці програмного та інформаційного забезпечення

- формалізація знань предметної області "Ергономічне проектування інформаційних моделей";
- реалізація інтерфейсу для проведення опитування експерта в конкретній предметній області "Технологічний процес і його об'єкти контролю та управління";
- інтелектуальна підтримка проектувальника ЛМІ (аналіз даних, отриманих в результаті опитування експерта і формування рекомендацій на всіх етапах проектування ЛМІ);
- для забезпечення в реальному часі збору, обробки, відображення та архівування інформації про об'єкти контролю та управління необхідно розробити модуль візуалізації.



НУ "ЗП", Кафедра програмних засобів

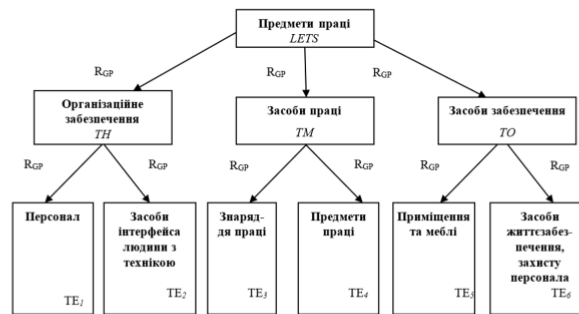
6

Рисунок Б.6 – Слайд №6

Системний аналіз діяльності проєктувальника інтерфейсів

1. Компонентний аналіз – для виявлення всіх сутностей (об'єктів, функцій, станів), що складають вміст даної системи "людина-техніка-середовище" СЛТС(N).

Побудовано комплекс наступних компонентних моделей: компонентно-системних KSS(N); компонентно-елементних KES(N) і компонентно-функціональних KFS(N).



НУ "ЗП", Кафедра програмних засобів

7

Рисунок Б.7 – Слайд №7

Системний аналіз діяльності проєктувальника інтерфейсів

2. Морфологічний аналіз – для відображення різного роду взаємозв'язків (організаційних, технологічних, часових та ін.), що існують між системами й компонентами, виділеними в результаті компонентного аналізу.

Побудовано:

2.1. Функціонально-часову структуру (FTS) - алгоритмічну функціональну мережу процедур (вершинами графа є імена процедур, а дуги відповідають елементам відношення Rr: «процедура P_i , безпосередньо передувє процедурі P_{ij} »)

2.2 Функціонально-організаційну структуру (FOS). Структури FOS являють собою набір графів, число яких дорівнює числу функцій, що досліджується. Кожний граф складається з вершин трьох типів: перший - ім'я функції F_i ; другий – узагальнене ім'я комплексу компонентів K_i , що реалізують виконання цієї функції; третій-імена елементів, що входять у цей комплекс.

3. Еволюційний аналіз – не виконувався.



НУ "ЗП", Кафедра програмних засобів

8

Рисунок Б.8 – Слайд №8

Результати аналізу правил для алфавіту звукові сигнали

Цілі	Факти	Правила
Ц1. Вибір частоти	Ф98, Ф99, Ф85	П80, П85, П87, П97
Ц1.1. Зміна частоти	Ф100	П88
Ц2. Вибір тривалості	Ф102, Ф85	П82, П84, П91
Ц2.1. Зміна тривалості	Ф97	П100, П101
Ц3. Інтервал між сигналами	Ф85	П89, П90
Ц3.1. Тривалість інтервалу	Ф85, Ф95	П83, П92, П95
Ц4. Рівень звукового тиску	Ф85, Ф98	П81, П86, П94, П96, П98
Ц4.1 Зміна рівня звукового тиску	Ф101	П99



НУ "ЗП", Кафедра програмних засобів

11

Рисунок Б.11 – Слайд №11

Дерево рішень для визначення частоти звукового сигналу

Ф98	Ф99	Ф85	Правила, що використовуються
0	0	0	П80
0	0	1	П97
0	1	0	П87
0	1	1	П87, П97
1	0	0	П85
1	0	1	П97
1	1	0	П87
1	1	1	П87, П97

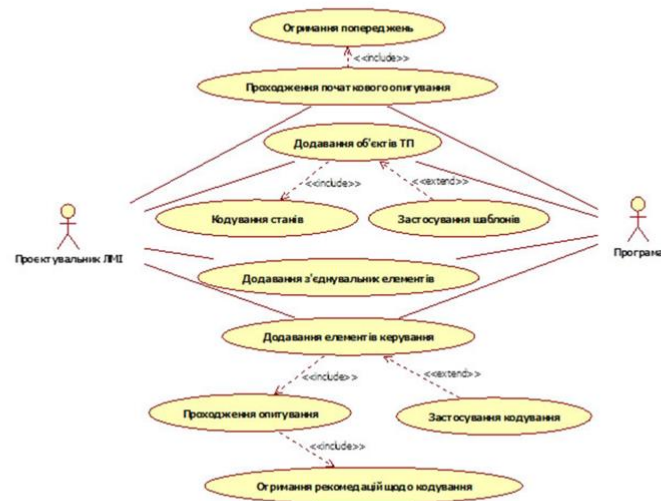


НУ "ЗП", Кафедра програмних засобів

12

Рисунок Б.12 – Слайд №12

Діаграма прецеденту «Розробка ЛМІ»

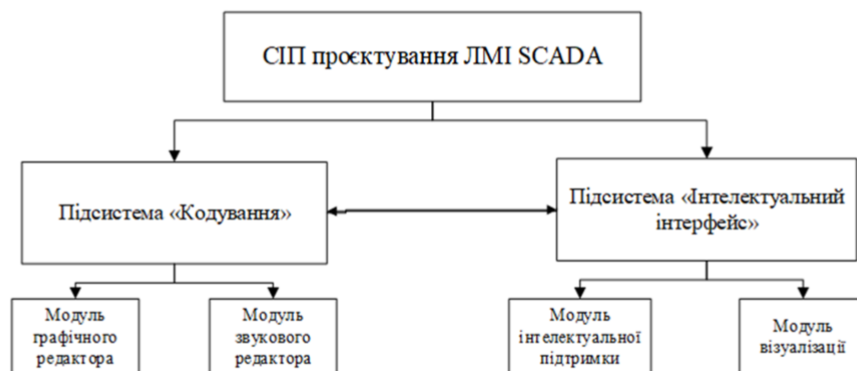


НУ "ЗП", Кафедра програмних засобів

13

Рисунок Б.13 – Слайд №13

Загальна структура системи

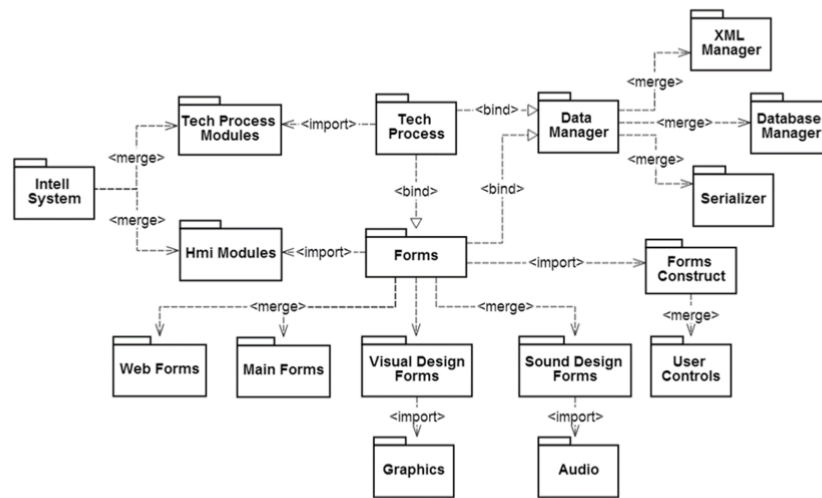


НУ "ЗП", Кафедра програмних засобів

14

Рисунок Б.14 – Слайд №14

Діаграма пакетів системи

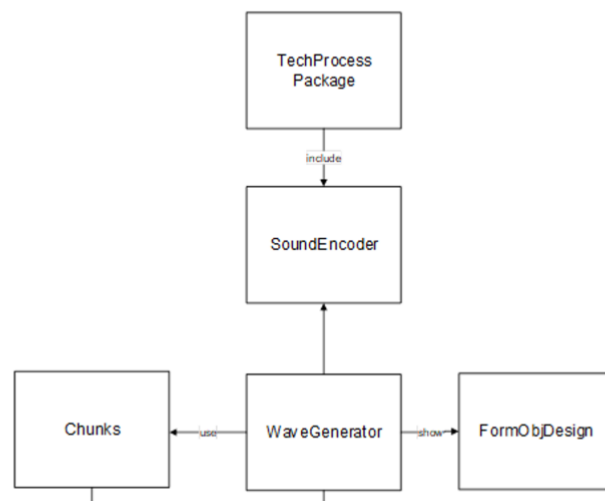


НУ "ЗП", Кафедра програмних засобів

15

Рисунок Б.15 – Слайд №15

Структура класів модуля звукового редактора

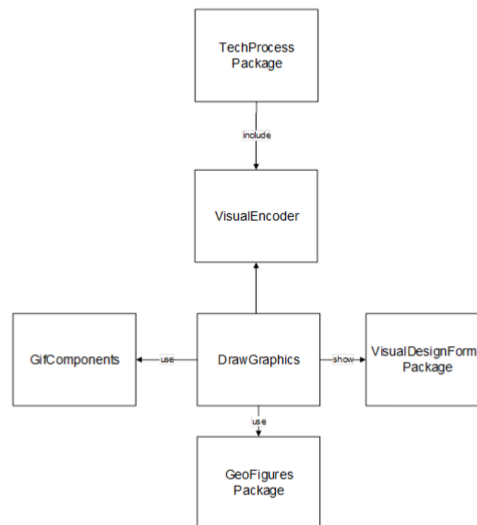


НУ "ЗП", Кафедра програмних засобів

16

Рисунок Б.16 – Слайд №16

Структура класів модуля графічного редактора



НУ "ЗП", Кафедра програмних засобів

17

Рисунок Б.17 – Слайд №17

Приклад основних даних та методів класу IntelStateModule

Атрибути	Тип/Значення, що повертається	Назва	Параметри	Опис
1	2	3	4	5
private	Dictionary<string, Fact>	facts		Словник фактів модуля
public	void	ShowGoal Interview		Проводить опитування розробника за мету стану
public	void	ShowShape Interview		Проводить опитування розробника за форму знаків
public	void	ShowText Interview		Проводить опитування розробника за кодування текстом
public	void	ShowColor Interview		Проводить опитування розробника за кодування кольором
public	void	ShowBrightnessInterview		Проводить опитування розробника за кодування яскравістю
public	void	ShowSound Interview		Проводить опитування розробника за кодування звуком
public	void	SetFactsFrom StateType	StateType stateType	Встановлює факти відповідно типу стану
public	string	GetShapeRecommendation		Надає рекомендації за формою

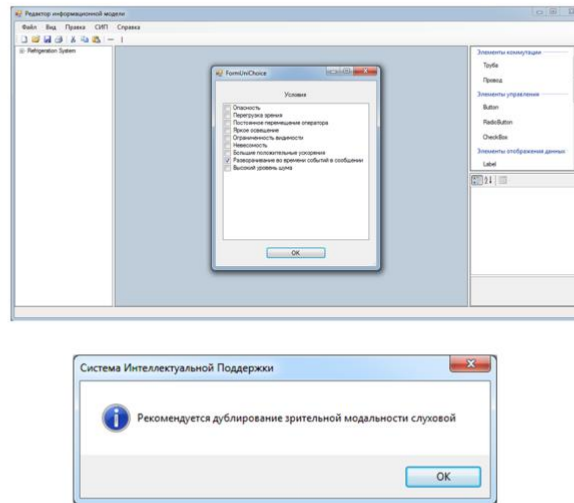


НУ "ЗП", Кафедра програмних засобів

18

Рисунок Б.18 – Слайд №18

Опитування стосовно зовнішніх умов роботи оператора

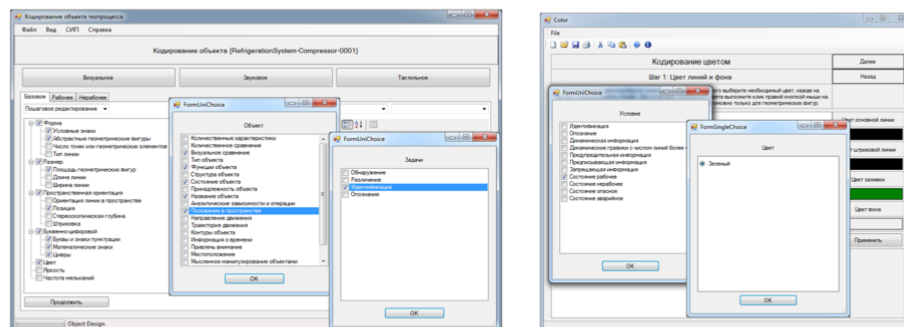


НУ "ЗП", Кафедра програмних засобів

19

Рисунок Б.19 – Слайд №19

Вибір алфавіту кодування

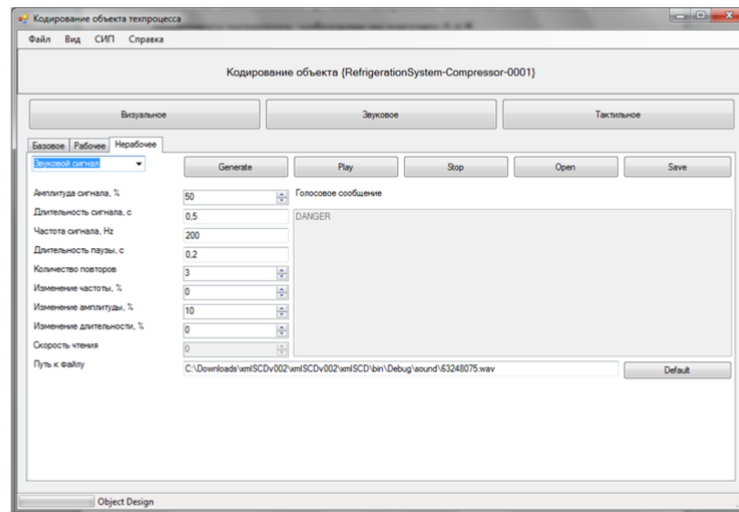


НУ "ЗП", Кафедра програмних засобів

20

Рисунок Б.20 – Слайд №20

Вікно звукового редактору

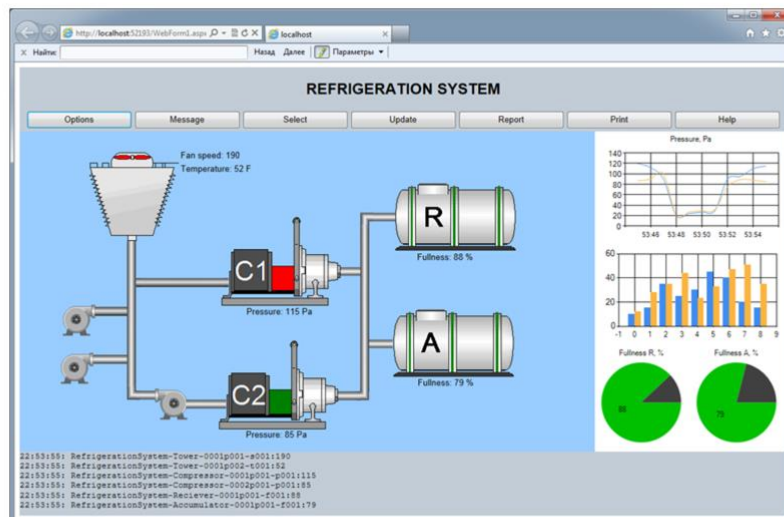


НУ "ЗП", Кафедра програмних засобів

21

Рисунок Б.21 – Слайд №21

Приклад роботи web застосунку



НУ "ЗП", Кафедра програмних засобів

22

Рисунок Б.22 – Слайд №22

Висновки

В результаті виконання кваліфікаційної роботи магістра було виконано такі задачі:

- розглянуто призначення та структуру SCADA-систем;
- досліджено існуючі технології побудови SCADA-систем та обрано для розробки web технологію;
- проаналізовано стан досліджень у області проектування ЛІМІ SCADA-систем та обрана базова ІТ, яка дістала подальшого розвитку у даній роботі;
- проведено системний аналіз діяльності проєктувальника інтерфейсів та виявлено номенклатуру проблемних ситуацій, у яких він опиняється під час розробки інтерфейсів SCADA-систем;
- розроблено ІТ інтелектуальної підтримки проєктувальників ЛІМІ SCADA-систем для двох типів модальностей – зорової та слухової;
- розроблено у вигляді вирішуючих дерев стратегію кодування технологічної інформації слуховою модальністю;
- розроблено у вигляді UML діаграм моделі програмного забезпечення для автоматизації діяльності проєктувальників ЛІМІ SCADA-систем;
- розроблено програмне та інформаційне забезпечення компонентів системи, що інтелектуально підтримує проєктувальника;
- перевірено розроблене програмне забезпечення на прикладі проєктування фрагментів ЛІМІ SCADA-системи «Система охолодження».



НУ "ЗП", Кафедра програмних засобів

23

Рисунок Б.23 – Слайд №23

Висновки

Наукова новизна результатів дослідження полягає у тому, що дістала подальшого розвитку відома ІТ інтелектуальної підтримки процесу ергономічного проєктування ЛІМІ шляхом введення в неї засобів та правил кодування технологічної інформації слуховою модальністю.

Практична цінність роботи полягає в тому, що на підставі досліджень розроблено програмне та інформаційне забезпечення, що забезпечує інтелектуальну підтримку проєктувальника ЛІМІ SCADA-систем.



НУ "ЗП", Кафедра програмних засобів

24

Рисунок Б.24 – Слайд №24

**Доклад закінчено,
дякую за увагу!**



НУ "ЗП", Кафедра програмних засобів

25

Рисунок Б.25 – Слайд №25