

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
Факультет інформатики та обчислювальної техніки  
Кафедра автоматичного управління в технічних системах

«До захисту допущено»

Завідувач кафедри

О.І. Ролік

« \_\_\_\_ » \_\_\_\_\_ 2019 р.

## Дипломний проект

на здобуття ступеня бакалавра  
з напрямку підготовки б. 050201 «Системна інженерія»  
на тему: «Рецептурне керування виробничого процесу засобами CoDeSys»

Виконав :

студент (-ка) IV курсу, групи ІА-52

Кононенко Д.О.

Керівник:

Яланецький В.А.

Рецензент:

Засвідчую, що у цьому дипломному проекті немає  
запозичень з праць інших авторів без відповідних  
посилань.

Студент (-ка) \_\_\_\_\_

Національний технічний університет України

«Київський політехнічний інститут» ім. Ігоря  
Сікорського

Інститут (факультет) Факультет інформаційної та обчислювальної техніки

Кафедра \_\_\_\_\_ автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки \_\_\_\_\_ 6.050201 Системна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ А.І.Петренко  
(підпис) (ініціали, прізвище)

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломний проект студенту**

Кононенко Дмитро Олеговичу

1. Тема роботи Рецептурне керування виробничого процесу засобами CoDeSys,  
керівник роботи \_\_\_\_\_ ст.викладач Яланецький В.А \_\_\_\_\_ ,  
затверджені наказом по університету від « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи

1. Поверхневий струм дорівнює  $10-50 \text{ mA/cm}^2$ .
2. Кількість кислоти дорівнює 60000 г.
3. Температура для анодування  $68 \text{ }^{\circ}\text{F}$
4. Час анодування 45 хвилин

4. Зміст роботи

1. Постановка задач рецептурного керування.
2. Огляд та вибір найкращих технічних рішень щодо рецептурного керування
3. Компоненти системи рецептурного керування.
4. Приклад реалізації рецептурного керування виробничим процесом

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

1. Операційний щит — плакат.
2. Схема електрична структурна — плакат.
3. Схема електрична функціональна — плакат.
4. Презентація проекту.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 14.03.2019

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	14.03.2019	
2	Збір інформації	21.05.2019	
3	Проектування моделей системи	25.05.2019	
4	Реалізація рецептурної системи керування	02.06.2019	
5	Отримання допуску до захисту		

Студент \_\_\_\_\_

Кононенко Д.О.

Керівник роботи \_\_\_\_\_

Яланецький В.А.

# АНОТАЦІЯ

Тема роботи: Рецептатурне керування виробничого процесу засобами CoDeSys.

Виконавець: Кононенко Дмитро Олегович

Пояснювальна записка містить: 62 сторінки, в яких \_\_ рисунків, 4 діаграми, 2 таблиці, 1 додаток, \_\_ посилань.

Ключові слова: рецептатурна система керування, автоматизоване виробництво анодирування, засоби CoDeSys, ModeBus, XML, рецептура.

Мета роботи: створення рецептатурної системи керування виробничого процесу засобами CoDeSys.

Результати: комп'ютерна система, яка надає інформацію виконавчим пристроям через, закодованими в CoDeSys, мікроконтролери, що виконує процес анодування великих і малих авіаційних деталей на промисловому цеху.

№ рядка	Формат	Позначення	Найменування	К-сть аркушів	№ схеми	Примітка	
1			<u>Документація загальна</u>				
2							
3			Заново розроблена				
4							
5	A4	IA52.140БАК.002 ПЗ	Пояснювальна записка	61			
6	A3	IA52.140БАК.003 Е1.1	Структурна схема оптимального	1			
7			Керування виробничим процесом				
8	A3	IA52.140БАК.004 Е1.2	Структурна модель реалізацій	1			
9			системи рецептурного керування				
10	A3	IA52.140БАК.005 Д1	Діаграми діяльності рецептурного	1			
11			керування				
12	A3	IA52.140БАК.006 Д2	діаграма варіантів використання	1			
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
Зм	Арк.	№ докум.	Підпис	Дата			
Розроб.		Кононенко Д.О.			Літер.	Аркуш	Аркушів
Перевір.		Ялинецький В.А			Т	1	1
Н.контр.							
Затв.							

**Пояснювальна записка  
до дипломного проекту**

на тему: Рецептурне керування виробничого процесу засобами CoDeSys

Київ-2019 року

## ЗМІСТ

ВСТУП.....	3
1 ПОСТАНОВКА ЗАДАЧІ РЕЦЕПТУРНОГО КЕРУВАННЯ.....	5
1.1 Мета проекту.....	5
1.2 Огляд стандарту рецептурного керування.....	6
1.3 Призначення та галузь використання системи рецептурного керування.....	9
1.4 Формування технічних вимог до системи рецептурного керування.....	11
Висновки до розділу.....	13
2 ОГЛЯД ТА ВИБІР НАЙКРАЩИХ ТЕХНІЧНИХ РІШЕНЬ ЩОДО РЕЦЕПТУРНОГО КЕРУВАННЯ.....	14
2.1 Підсистема рецептурного керування гальванічного цеху .....	14
2.2 Підсистема рецептурного керування виготовлення шоколаду.....	16
2.3 Зони рецептурного керування виробництва безалкогольних напоїв.....	18
Висновки до розділу.....	20
3 КОМПОНЕНТИ СИСТЕМИ РЕЦЕПТУРНОГО КЕРУВАННЯ.....	21
3.1 Структурні моделі системи рецептурного керування.....	21
3.1.1 Створення структурної моделі реалізації системи рецептурного керування.....	21
3.1.2 Створення діаграми діяльності рецептурного керування....	22
3.1.3 Створення діаграми варіантів використання.....	23
3.2 Структура зберігання рецептури.....	24
3.2.1 HTML.....	24
3.2.2 XML.....	29

Зм	Арк.	№ докум.	Підп	Дата		Літер.	Арку	Аркушів
Розроб.		Кононенко Д.О.				Т	1	61
Перевір.		Яланецький В.Г.						
Н.контр.								
Затв.								

3.2.3 Специфікація XML – BatchML.....	31
3.2.4 Специфікація XML – B2MML.....	35
3.2.5 Рішення щодо зберігання рецептури.....	39
3.3 Збереження рецептів системи рецептурного керування в базі даних.....	40
3.4 Інтерфейс оператора для керування рецептури.....	41
3.5 Програма контролерів.....	43
Висновки до розділу.....	45
<b>4 ПРИКЛАД РЕАЛІЗАЦІЇ РЕЦЕПТУРНОГО КЕРУВАННЯ ВИРОБНИЧОГО ПРОЦЕСУ.....</b>	<b>46</b>
4.1 Опис та зберігання рецептури в XML файлі.....	46
4.2 Програма для контролерів.....	50
4.3 Імпорт файлу XML.....	55
4.4 Програмування взаємозв'язку між контролерами через MODBUS.....	56
4.5 Результати виконання.....	58
Висновки до розділу.....	59
ЗАГАЛЬНІ ВИСНОВКИ.....	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТОК А.....	62

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Сьогодні дуже складно уявити будь-яке виробництво без конвеєрних ліній. Будь то певного роду конвеєри для транспортування проміжної продукції між цехами, чи ціла пов'язана купа підсистем, які в результаті виконують повний цикл виробництва певного продукту. Зараз дуже поширеним напрямком автоматизації є галузі промисловості, де є можлива небезпека діяльності працівника. Проблема цієї галузі лежить на поверхні – необхідність точних, високо обрахованих рішень, що відповідають якості та безпеки підприємства. Навіть малий дефект може коштувати великих збитків, або навіть життя людей. Саме через це такі підприємства потребують автоматичний контроль виробничих ліній.

Цінність цього проекту – удосконалення системи, для полегшення виробничого процесу робітникам, методом заміни ручного пересування та прямого контролю на алгоритм рецептурного керування – проведення лінійного процесу, через ділення на виробничі комірки, завдяки записаним кількісним характеристикам в рецептурі, що зберігається окремо в деякому просторі: база даних, локальний файл, чи матеріальній носій інформації. Завдяки чому процес контролюється на безпечній відстані від самого виробництва, що в свою чергу зменшує ризик для здоров'я працівника, а також мінімізує похибку від людського фактору. Крім цього дана ідея сприяє підвищенню якості обробки й подальшої експлуатації матеріалу.

При дослідженні рецептурного керування, використано досвід при проектуванні курсової роботи на схожу тематику. Маючи попередні проекти основані на контролі мікропроцесорних систем в основі взяті з методів CoDeSys, було відтворено, з поправками введення бази даних, рецептурне керування процесу в масштабах невеликого цеху.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		



# 1 ПОСТАНОВКА ЗАДАЧІ РЕЦЕПТУРНОГО КЕРУВАННЯ

## 1.1 Мета проекту

Мета проекту – дослідження та практичне виконання завдання з розробкою системи рецептурного керування.

В цій роботі буде розглядатися резервуар з сірчаною кислотою, чия насиченість знаходиться в межах 1200-1300 грамів на літр, саме в контексті об'єкту керування та автоматизованої системи рецептурного керування в технічному процесі гальваніки. Також до цього входить електричне коло, що має динамічну характеристику й повинна керуватись відносно рецептури виробництва, чий діапазон складатиме 10-50 мА/см<sup>2</sup>, на рівні з часом та температурою розчину: -20 до +20 °С.

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					

## 1.2 Огляд стандарту рецептурного керування

Тенденція розвитку корпоративних систем привела до розуміння, що для ефективного управління такими складними виробництвами потрібен єдиний стандарт, ANSI / ISA 88 (часто також використовується скорочення S88).

Стандарт ISA-88 розроблявся на базі існуючого на той момент стандарту NAMUR N33 і був покликаний допомогти у вирішенні кількох фундаментальних проблем, таких як відсутність єдиної моделі рецептурного виробництва, складність узгодження вимог, труднощі інтеграції рішень різних постачальників, складне управління рецептурним виробництвом. Структура стандарту відповідає поставленим завданням і включає чотири частини:

- Models and Terminology - визначає стандартні моделі і термінологію для формалізації вимог до систем управління періодичним виробництвом
- Data Structures and Guidelines for Languages - визначає моделі даних для управління виробництвом, структури даних для обміну інформацією, а також форму записи рецептури
- General and Site Recipe Models and Representation - визначає моделі для подання узагальнених рецептур і обміну такими рецептурами між підрозділами підприємства, а також між підприємством і його партнерами;
- Batch Production Records - визначає моделі даних і орієнтовну модель системи для запису, зберігання, вилучення та аналізу даних про хід періодичного виробництва.

Крім того, готується до виходу п'ята частина стандарту Implementation Models & Terminology for Modular Equipment Control, мета якої полягає у визначенні методів формалізації інтеграції виробничого і пакувального обладнання, а також корпоративних інформаційних систем.

							Аркуш
Зм	Арк.	№ докум.	Підпис	Дата			

Також на основі концепції стандарту ISA 88 розроблені конструкції XML для обміну даними в системах управління періодичним виробництвом - BatchML.

Перш за все, стандарт ISA 88 визначає поняття періодичного процесу (batch process) - це процес, результатом якого є виробництво кінцевої кількості продукту шляхом виконання над деякою кількістю вихідних матеріалів (сировини) впорядкованої послідовності дій за обмежений період часу з використанням однієї або більше одиниць обладнання. Це визначення однозначно вказує на наявність обмеженого інтервалу часу (періоду) виготовлення кінцевої кількості продукту (тобто партії) і тим самим відділяє періодичний виробничий процес від безперервного або дискретного процесу.

Термін "партія" ("batch") має значення: по-перше, "матеріал, який виробляється в результаті однієї стадії batch process", а по-друге - якась сутність, яка визначає виробництво матеріалу на будь-якій стадії процесу. Під терміном "рецептура" ("recipe") визначається мінімально необхідний набір інформації, яка унікальним чином визначає вимоги до виробництва конкретного продукту. Для управління періодичним виробництвом необхідні три сутності (і саме їх охоплює стандарт):

- формальне визначення процесу виготовлення партії продукту - рецептури (recipe);
- інформація про обладнання, яким потрібно керувати (модель обладнання);
- формальне визначення керуючих впливів.

Стандарт ISA 88.01 визначає фізичну і процедурну моделі виробництва. Фізична модель в цілому визначає виробничу осередок обладнання, необхідного для виробництва партії продукції. Основним поняттям тут є модуль - основна одиниця устаткування, що виконує головний крок процесу. Фізична модель (модель обладнання) в загальному випадку включає сім рівнів:

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		



### 1.3 Призначення та галузь використання рецептурної системи керування

Рецептурна система керування процесом виробництва, проектування якої виконується в цьому проекті призначена для використання в будь-якій промисловості, де є чіткий протокол дій, що іменується як рецепт, чи рецептура виробництва. Проблемою дана система не володіє, окрім виявлення методичних похибок, що можливі на ранніх етапах виробництва, так зване експериментальне виробництво, при якому ми отримуємо окрім алгоритму послідовності дій та параметрів якості кінцевого продукту, ще й допуски кожного параметру рецептури на кожному етапі дій. Крім цього є варіант інструментальної похибки, що вже не є виною рецептури, яка виникає при зношенні виконавчих механізмів в системі. Тому на кожному виробництві, в кінці лінії, знаходиться особа, що має підтвердити якість, що знов закріплена за рецептом й не залежить від присутності чи відсутності рецептурного керування. Значним плюсом є максимальне унеможливлення особистої, людської похибки, бо усі рішення та розрахунки лягають безпосередньо на раніше протестовану рецептуру. Процес створення рецептури йде завжди зберігає право на оновлення. Тому на сьогоднішній день, багато підприємств мають обернену систему опрацювання рецепту. Це може пряме втручання в файл з послідовними процесами та кількісними характеристиками, через зберігач інформації, такі як флешка, чи жорсткий диск. Або створити повноцінну базу даних, оснований на сучасних системах: Oracle, MySQL, тощо. Головне, обов'язково, мати систему з оберненим зв'язком для можливості контролю рецептури без різних проблем. Звісно система рецептурного керування призначена для використання в умовах великих виробничих комплексів та цехів майже в усіх галузях, її можливості є основним засобом, для утворення постійного автоматизованого виробництва, безперервного конвеєру, тощо.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		



## 1.4 Формування технічних вимог до системи рецептурного керування

Основним критерієм вимог до системи рецептурного керування є реалізація відносин в автоматизованій системі керування (Кресленик).

Як указано в кресленику, керування рецептури має впливати лише на планування і складання графіку виробничого процесу, керування технологічною коміркою та керування виробничої інформації.

Керування рецептури первинно виконує такий набір функцій:

- зберігання кількісних параметрів в блоці пам'яті рецептур;
- внесення змін в значення параметрів від зовнішнього пристрою;
- передача для виконання параметрів на пульт керування.

Головна задача керування це взаємодія з керівництвом технологічної комірки. Кожний цех на виробництві ділиться на деякі етапи зі своїм обладнанням. Данні частини процесу мають назву комірка. До комірки прив'язується крок в алгоритмі технічного процесу, засоби, інформація, що збирається з рецептури. Окремо виділяється вид обладнання та його контроль. У виробництві приймають участь статичні об'єкти керування (апарати), що представляють собою резервуари, та динамічні об'єкти (агрегати), що є сукупністю виконавчих пристроїв з резервуарами. Нижче цього виходить контроль цих виконавчих пристроїв через мікро-контролери, а саме керування процесом.

Окрім поділеного на фази алгоритму виробництва, керування рецептури має притримуватись критеріям, що зустрічаються на всіх підприємствах. Система повинна мати в результаті продукт, який відповідає заданій кількості, собівартості та якості товару у сукупності часових втрат для створення його. Цей перелік має бути обов'язково присутнім при реалізації рецептури.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

Кореляцію даних про кількість, якість тощо вводить як саме підприємство, так й замовник товару, через заклази, плани графіку. Відносно них, рецептура має підходити для реалізацій задуму.

Також рецептурне керування має зв'язок з виробничою інформацією. В даному випадку це є реалізація так званої оберненого керування. Так за даними, що ми збираємо на промисловості, можна ввести корективи в рецептуру.

Останнім моментом є паралельна дія з системою безпеки, що припинає дію рецептурного керування на роботу технологічної комірки, через можливі порушення роботи загальної системи.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		



## 2 ОГЛЯД ТА ВИБІР НАЙКРАЩИХ ТЕХНІЧНИХ РІШЕНЬ ЩОДО РЕЦЕПТУРНОГО КЕРУВАННЯ

### 2.1 Підсистема рецептурного керування гальванічного цеху

Керування рецептури є дуже головний аспект на небезпечних підприємствах. Одне з яких є гальванічний цех, що розташований на державному підприємстві (ДП) «Антонов».

Процес анодування вимагає кількість сірчаної кислоти в розчині у діапазоні від 1200 до 1300 грамів на літр. При цьому температура резервуару контролюється в діапазоні від -20 до +20 градусів по Цельсію.

Процес починається з того що рецептура задає час, при якому об'єкт буде утримуватись в розчині-знезараження. Дана процедура необхідна для знищення усього органічного з поверхні об'єкту. Окрім цього, перед анодуванням, рецептура задає час відмочування поверхні в дисцельованній воді, чия температуру задає теж рецепт.

Далі перед початком процесу йде перевірка на хімічну густину розчину кислоти та починається нагрів її до необхідної температури (утримування температури виділяє небезпечний газ, що вимушує також за рецептурою вмикати в деякий час потужну вентиляцію).

Коли все готово й об'єкт занурили в сірчаний розчин, на резервуар буде подаватись струм, що буде змінюватись у часі. Функція значення току є параметричною й буде залежить не лише від часу а й від температури, що також буде отримувати зміни під час процесу.

В деякий час рецептура буде задавати нове значення температурі розчину, але на відміно від печі, резервуар не зможе моментально нагрітись (охолонути), тому дана реалізація є інтегрування функції часу, що означає при значеннях печі які будуть рівнятись константі, резервуар буде змінювати свою температуру по лінійній функції.

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					

У випадку з током, його функція буде вже другого порядку й буде мати параболічну криву на графіку, бо інтегрування вже лінійної функції за часом дасть квадратичну.

Крім цього в деякий час йде продування системи повітрям, для зниження концентрації небезпечного газу, що виділяється при процесі, що теж має місце в рецептурі.

Під кінець процесу, об'єкт занурюють в ще один резервуар з теплою, підбраною рецептурою, воду й залишають остуджуватись знов да деякий час.

Увесь процес рецептурного керування має місце на спеціальних дошках, де окрім значень рецептури, зберігається паспорт реагентів й техніки. Сам процес автоматизованого рецептурного керування має один суттєвий недолік: рецептура вже завчасно була запрограмована в обладнанні цеху й контролюється однією кнопкою на пульті управління процесу. Сама кнопка має підпис «Цикл». Крім цього через технічну прив'язаність, техніку складно амортизувати й з часом вона дає помилки, при яких диспетчер з працівниками цеху повинні вручну проводити процес анодування.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

## 2.2 Підсистема рецептурного керування виготовлення шоколаду

При виготовлення шоколаду, процес вимагає набагато точнішого плану дій ніж при рецептурі анодування. Зв'язано це тим, що рецептура повинна пройти стандарти ДСТУ, що вводить строгі границі до кількості сировини на кожному етапі.

На першому етапі, керування рецептури пов'язано з добирання співвідношення продуктів, що будуть готуватись. Так для різних видів шоколаду, різного призначення, різної кількості, процес, на даному етапі буде дуже відрізнятись. Усе це контролюється диспетчером, що має вибрати конкретний вид рецепту.

Основними параметрами є кількість какао, масла, цукру, молока.

Вже другий процес починається з агрегату, до якого підключена промисловий змішувач що має функцію змінювати частоту обертів, в залежності від часу й команди рецепту. Сама же цистерна підключена до печі, що буде нагрівати суміш поступово, допоки не дійде до визначеного рецептом температури й буде триматись на всьому другому етапі. Окрім нагрівання, змішування, процес доповнює додавання інших продуктів, наприклад горіхів. Час та кількість знову визначає рецепт. Готовність сировини повідомляється зворотнім шляхом по контролеру, що дозволяє коректувати процес рецептурного керування при несанкціонованих змінах в процесі.

Вже потім таймер дає трохи часу простояти сировині в резервуарі, після чого йде процес наповнювання, при умові що диспетчер вибрав процес виготовлення саме шоколадних плиток.

Отриману сировину відправляють в холодильник для охолодження й затвердіння плитки. Як й з пічкою, даний процес, рецептура контролює температуру й час, при яких плитка може втратити свою складову, при помилкових діях.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		



## 2.3 Зони рецептурного керування виготовлення безалкогольних напоїв

Деякі виробництва відомі новаторськими підходами, інші якістю та соціально-культурною приналежністю продукту відносно регіону виготовлення. Усе це збільшує собівартість, а головне ціну товару. Але є також особливість у розрекламованості товару, що дає світову славу.

При таких обставинах товар має створюватись у великих об'ємах та мати вже готові точки збуту. Маючи в розрахунках транспортування, компанія буде нести збитки. Для цього створюються місцеві точки виробництва.

У випадку компанії «Кока-кола», виробничий процес був поділен на 2 частини: приготовчий та розливний процес.

Рецептура маючи свої унікальні особливості є дуже ціною, бо за унікальність платять більше.

Так материнський завод створює первину секретну сировину, що поділяє раніше рецептуру на дві або більше автономні рецепти.

Кожна частина має усі особливості рецептури й може реалізовувати ті ж функції, що були при повній складовій рецептури. Така гнучкість контролю дозволяє проводити виконавчий процес не тільки паралельно, де усі процеси безперервно проводяться, а й збільшує масштаб реалізації продукту.

Материнська фабрика «Кока-кола» маючи роздроблений рецепт можуть продавати частини свого патенту, чия повна реалізація всерівно залишиться у первинного виробника.

Так пропустивши першу частину, що секретно проводиться за межами заводів, підприємство отримує вже оброблену сировину та автономну частину рецепту, що має реалізувати у себе.

Передаючи на контролери значення, в агрегат, разом зі секретною сировиною, починає добавлятися цукор, смакові підсоложувачі та природні барвники.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

Після цього продукт стоїть на нагріві, відповідно за рецептом, на високій температурі, близької до кипіння води. Далі, вже після нагрівання, йде процес повільного охолодження, що є підготовкою до газування напою.

Вже після цього йде рівномірне збагачення речовини вуглекиснем (CO<sub>2</sub>), що і дає газованість.

Вкінці продукт проходить етап розливу й пакування, що є вже за межами впливу рецептурного контролю виробничого процесу.

Даний приклад дає зрозуміти, що основним плюсом рецептури є складеність. Її функціонал та вплив можна розділити на різні цехи, переміщаючи локально в межах одного заводу, або при великих масштабах виробництва, пересилати частину рецепту й повноважень іншій фабриці, через більш прогресивну систему з використанням великих баз даних, хмарових системах тощо.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		



### 3. КОМПОНЕНТИ СИСТЕМИ РЕЦЕПТУРНОГО КЕРУВАННЯ

#### 3.1 Структурні моделі системи рецептурного керування

##### 3.1.1 Створення структурної моделі реалізацій системи рецептурного керування

Основою проектування є розуміння зв'язків структурних елементів, їх виконання, обов'язки та рівень виконання вищевказаних задач та функцій.

При виконанні системи рецептурного керування, а саме її проектування, присутні реалізації таких рівнів як: диспетчерський, контролю та виконавче-польовий.

Диспетчерським рівнем є інтерфейс, завдяки якого можливо керувати самою системою рецептури та виконувати задані функції системи. В умовах проекту, було обрано реалізацію у вигляді локального місця оператора (ЛМО).

Виконавче-польовим рівнем в системі рецептурного керування є контролери, що насамперед приймають рішення з рецептури й передають на виконання, а також можуть давати зворотні результати, для дослідження.

Рівень контролю є центром усієї системи, так як він дає вказівки базі, передає та приймає рішення інтерфейсу. Реалізація цього сегменту виконується саме в CoDeSys.

Проектування схеми було виконано в кресленні.

Кожен з цих рівнів має свою реалізацію, яка є унікальною в межах цього проекту. Окрім вибору реалізацій рівнів, також чинником є вибір області зберігання рецептури. Це можуть бути окремі додатки, віддалені бази даних, хмарова система, чи більш локальне зберігання на матеріальних носіях, таких як жорсткі диски, флешки.

Вибір та порівняння інструментів є частиною моделювання, що й буде розглянуто далі.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

### 3.1.2 Створення діаграми діяльності рецептурного керування.

Діаграма діяльності - в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Діаграми активностей будуються з обмеженої кількості фігур, з'єднаних стрілочками. Найважливіші типи фігур:

- округлені прямокутники позначають дії;
- ромби позначають рішення;
- риски позначають початок (розподіл) чи кінець (об'єднання) паралельних активностей;
- чорний кружок позначає старт (початковий стан) процесу;
- чорний кружок в колі позначає кінець (кінцевий стан).

Стрілки ведуть від старту до кінця і позначають порядок в якому відбуваються активності.

Особливість даної діаграми в тому, що завдяки їй будується логічний опис очікування учасника від системи. Огляд розповідає покроковий алгоритм, стандартами UML, що буде відбуватись для того, щоб отримати результат, який очікується.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

### 3.1.3 Створення діаграми варіантів використання

Візуальне моделювання в UML можна уявити як певний процес порівневого спуску від найбільш загальної і абстрактної концептуальної моделі вихідної системи до логічної, а потім і до фізичної моделі відповідної програмної системи. Для досягнення цих цілей спочатку будується модель у формі так званої діаграми варіантів використання (use case diagram), яка описує функціональне призначення системи або, іншими словами, те, що система буде робити в процесі свого функціонування. Діаграма варіантів використання є вихідним концептуальним поданням або концептуальною моделлю системи в процесі її проектування і розробки.

Розробка діаграми варіантів використання переслідує мети:

- визначити загальні межі і контекст модельованої предметної області на початкових етапах проектування системи;
- сформулювати загальні вимоги до функціонального поведінки проектованої системи;
- розробити вихідну концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей;
- підготувати вихідну документацію для взаємодії розробників системи з її замовниками і користувачами.

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					



```

<!DOCTYPE html>
<html>
<head>
<title>This is document title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>Document content goes here.....</p>
</body>
</html>

```

Для того щоб перевірити результат цього коду HTML, або протестувати, треба зберегти його в архіві HTML test.htm використовуючи редактор тексту. Остаточно розкрийте його використовуючи браузер павутини як Internet Explorer або Google Chrome, або Firefox etc. воно має показати наступний результат, як вказано на рисунку 3.1.

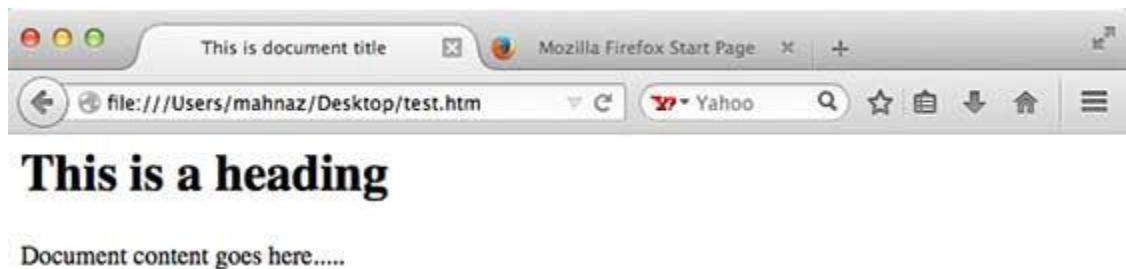


Рисунок 3.1 – Результат роботи HTML

Як сказано раніше, HTML як мова підвищення ціни використовує різні бирки для того щоб форматувати вміст даних. Ці бирки укладені всередині <Tag Name> кутових зв'язків. За винятком небагатьох бирок, велика частина з бирок має їх відповідаючі заключні бирки. Наприклад <html> має свою заключну бирку </html> і бирка <body> має свою заключну бирку etc.

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					

Основні види бирок для форматування документа описано в таблиці 1.

Таблиця 3.1 – Основні бирки HTML

Синтаксис бирки	Опис
<!DOCTYPE ...>	Ця бирка визначає тип документа і версію HTML.
<Html>	Ця бирка укладає повний документ HTML і головним чином складається з колектора документа який представлений <head> ... </head> і тілом документа яке представлено бирками <body> ... </body>.
<Head>	Ця бирка представляє колектор документа який може тримати інші бирки HTML як <title>, <link> etc.
<Title>	Бирка <title> використана всередині бирки <head> для того щоб згадати назву документа.
<Body>	Ця бирка представляє тіло документа яке тримає інші бирки HTML як <h1>, <div>, <p> etc.
<H1>	Ця бирка представляє рубрику.
<P>	Ця бирка представляє параграф

Для того щоб правильно використовувати HTML, потрібно знати різні бирки і розуміти як вони надходять форматний текстуальний документ. Використовувати HTML у міру того як поставлена задача, для того щоб форматувати певну базу, чи файл з зображенням.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

Типовий документ HTML матиме наступну структуру:

<Html>

<Head>

Бирки колектора документа родинні

</ Head>

<Body>

Бирки тіла документа родинні

</ Body>

</ Html>

Данна робота представляє собою, як за допомогою бирок можливо об'явлення документу.

<! Бирка оголошення DOCTYPE> використовується браузером павутини для того щоб зрозуміти яку версію HTML використовує документ. Прикладом є версія HTML 5 і вона використовує таке оголошення:

<! Html> DOCTYPE

Багато інших типів оголошення можуть знадобитися в документі HTML в залежності від версії HTML, яку використовують. Крім цього, <! DOCTYPE ...> бирка разом з іншими бирками, HTML маркує.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		



### 3.2.2 XML

Розширювана мова розмітки (XML) використовується для опису даних. Стандарт XML - це гнучкий спосіб створення інформаційних форматів та електронного обміну структурованими даними через публічний Інтернет, а також через локальні, корпоративні мережі.

Код XML, офіційно має рекомендацію від Консорціуму Всесвітньої павутини (W3C), подібно з мовою розмітки гіпертексту (HTML). XML і HTML містять символи розмітки (теги) для опису вмісту сторінки або файлу.

XML дані відомі як само-описуючі або само-визначаючі, що означає, що структура даних вже раніше вбудована разом з самими даними, тому, коли дані надходять, немає необхідності попередньо реконструювати структуру для зберігання даних. Він динамічно строється всередині XML. Формат XML може використовуватися будь-якою особою або групою осіб або усією компанією, які бажають поділитися, використати, або переписати інформацію. XML насправді є простішим при використанні Стандартної узагальненої мови розмітки (SGML), яка є стандартом для створення структури документів.

Документ XML вважається "добре налаштованим", тобто може бути прочитаний і зрозумілий XML-парсером. У перекладі даний термін має значення синтаксичний аналізатор - це програма, зазвичай частина компілятора, яка отримує вхідні дані у вигляді послідовних команд програм-джерел, інтерактивних онлайн-команд, тегів розмітки або іншого визначеного інтерфейсу і розбиває їх на частини. Наприклад, іменники - об'єкти, дієслова - методи та їх атрибути або параметри, які потім можна керувати іншими програмами. Синтаксичний аналізатор також може перевірити, чи введено всі необхідні дані.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		



### 3.2.3 Специфікація XML - BatchML

BatchML – спеціальна реалізація XML, відповідно до стандарту ISA-88. Даний стандарт створив індивідуальну конструкцію, конструкція чия створена спеціально для виробництв, що мають періодичний характер.

Стандарт ISA-88 визнаний у всьому світі як стандарт для серійної обробної промисловості. BatchML надає набір типів XML і визначень елементів, які можуть використовуватися частково або в цілому для пакетного, головного рецепта, сайту і загального рецепта, запису виробництва і даних обладнання. BatchML є відмінним інструментом для використання при обміні даними на основі ISA-88. Будь-яка компанія може використовувати роялті BatchML безкоштовно, за умови надання кредиту MESA. Міжнародна асоціація виробничих рішень (MESA) International - це глобальна спільнота виробників, виробників, лідерів галузі та постачальників рішень, які орієнтовані на досягнення результатів у виробництві інформації. Так усі стандарти, що виходять з цієї асоціації мають реалізацію у вигляді конструкції XML формату.

BatchML охоплює чотири основних концепта:

- як зобразити те що є на виробництві (утворення фізичної моделі промисловості);
- як отримати те, що ви збираєтесь створити на заводі (створення рецептури);
- реалізація того, що ви можете зробити на заводі (обумовленість логікою обладнання);
- як на основі, вище вказаних пунктів, створити інтелектуальну, багаторазову, правильну реалізацію.

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					

Особливості ISA-88:

а) дизайн об'єкту автоматизації:

- повторне використання;
- управління знаннями;
- надійність.

б) гнучке проектування автоматизованих додатків , без потрібного програмування:

- модифікації рецептів, виготовлення різних виробів з однаковим об'єктом;
- використання альтернативного обладнання для того самого етапу виробництва;
- послідовність виробництва проходить для різних продуктів одночасно.

в) оперативна сумісність:

- складання програмних компонентів різного походження.

індустріалізація продукту:

- нейтральна специфікація фізико-хімічних перетворень продукту;
- перетворення цієї специфікації в робочу процедуру для цільових засобів.

г) інформація про виробництво:

- структуризація даних для історії виробничої інформації;
- включає кілька моделей ISA-95.

г) програми:

- спочатку розроблений для пакетних процесів, але може застосовується до будь-якого типу процесу;
- функціональна специфікація, керування партіями, PDM / PLM.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

д) автоматизація функціональної інженерної практичності:

– значною мірою застосовується в усьому світі, доведена вигода та ефективність методу.

е) галузь:

- найбільш відомість має в переробній промисловості;
- менше в інших областях.

є) усі постачальники автоматизації пропонують своє рішення їхнього завдання:

- як правило, обмежуються підтримкою рецептів (створення, виконання, звітування);
- рідше на рівні контролю обладнання.

ж) застосовується здебільшого до всіх видів виробництв:

- немає еквівалентних стандартів для дискретних і безперервних процесів;
- легко адаптується.

з) більшість переваг при глобалізованому застосуванні:

- всі типи процесів;
- всі локації.

У реалізації, BatchML має вигляд діаграми, приклад якого є на рисунку

3.3

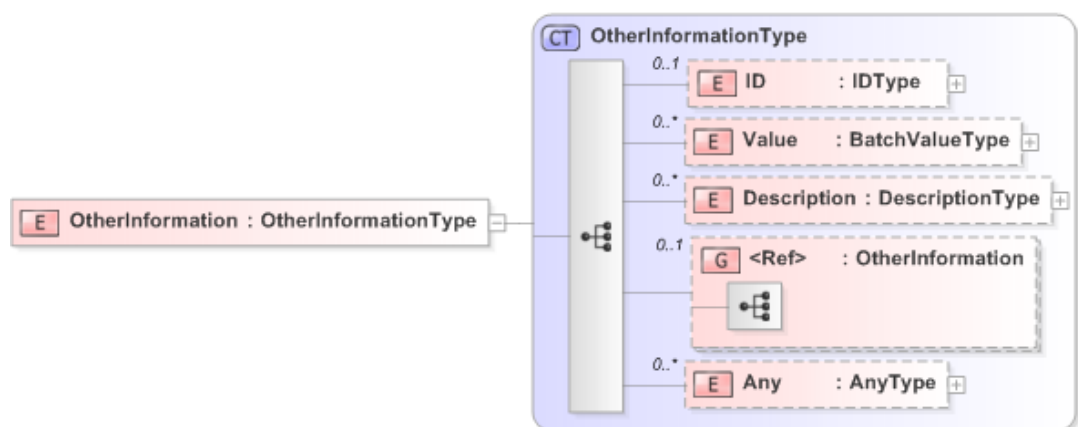


Рисунок 3.3 – Приклад реалізації BatchML





б) переваги:

- зниження витрат і ризиків проектів MES та B2M;
- удосконалення експлуатаційних характеристик: СТР, гнучке виробництво, планування зворотного зв'язку, зменшення помилок;
- динамізація ІТ / бізнесу;
- B2MML стає розповсюдженим стандартом для обміну інформацією B2M;
- більшість виробників автоматизації / MES вимагають приєднання до ISA 95. Але ERP ігнорує це.

Побудова діаграми вказано на рисунку 3.5.

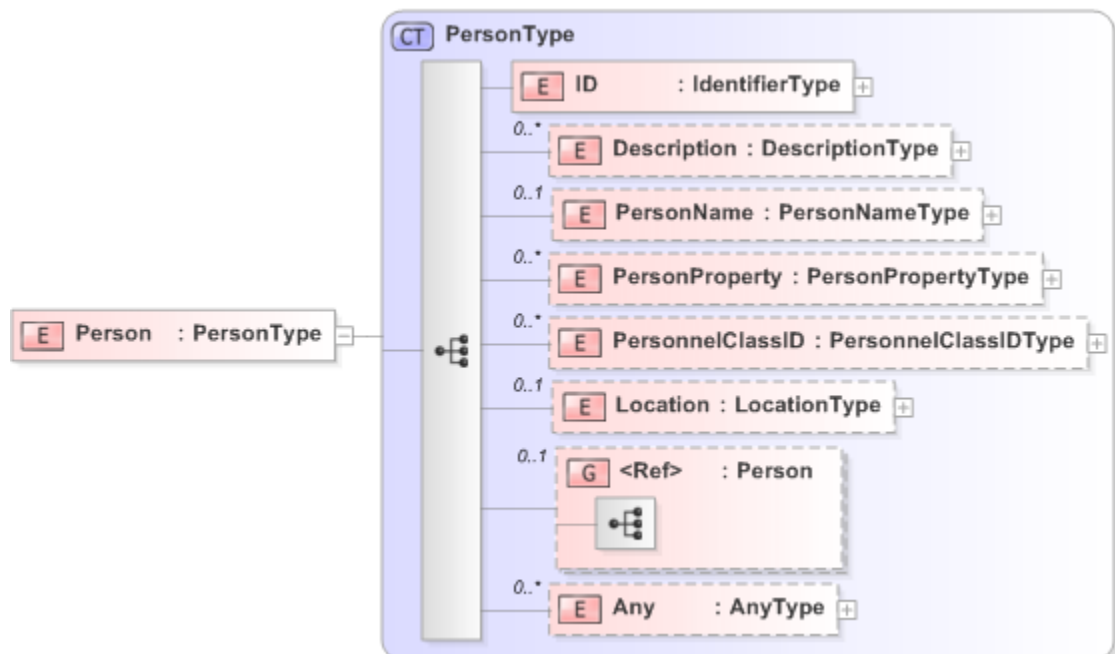


Рисунок 3.5 – Структура B2MML

Як й випадку з BatchML, будова знову має аналог з класами, об'єктами, тощо. Також присутні елементи, типи та архітектура дерева. Ідеї та стандарти цього рішення знов відсилають до оформлення XML файлів. Єдина відмінність від XML (як й BatchML) є вузько направлена призначення даної специфікації та основна термінологія, яка наведена в прикладі, а саме таблиця.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.2 – Порівняння термінів XML та B2MML

XML	B2MML
Bill of Material	Product definition
External ID	ID
Article Name	Description
Range Name	Equipment specification
Range charging post name	Equipment specification description
Line of BOM article name	Material specification
Valid from	Duration
Valid until	Duration
Quantity article	Quantity

Порівнюючи з B2MML та B2MML, основною різницею між ними стає різне використання за стандартами інтеграції, що вказано на рисунку 3.6

Так стандартний XML структура може охоплювати усі рівні (окрім Plant control). У випадку його специфікацій, структура має більше можливостей саме в деякому рівні контролю.

Як було сказано раніше, B2MML контролює рівень коміркового виконання, що означає надання певній частині виробництва певну, потрібну інформацію. Крім цього був надано контроль апаратного, диспетчерського рівню (SCADA) - програмний пакет, призначений для розробки або забезпечення роботи в реальному часі систем збору, обробки, відображення та архівування інформації про об'єкт моніторингу або управління.

У випадку В2ММЛ, контроль виробничих комірок був ідентичним, але при цьому аспект контролю був прикладений до контролю на бізнес-логістичному рівні. Це давало системі керування поставки, оснащення, планів, графіків, тощо.

						<i>Аркуш</i>
<i>Зм</i>	Арк.	№ докум.	Підпис	Дата		

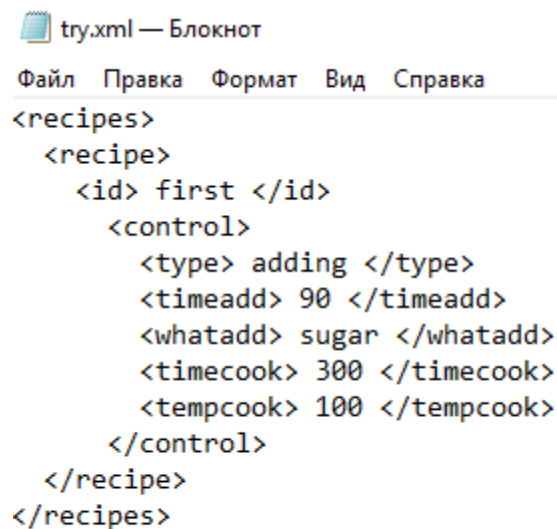


### 3.2.5 Рішення щодо зберігання рецептури

Основним рішенням є використання для реалізації проекту простий XML формат файлу. Це рішення дозволяє не тільки при простих умовах реалізовувати їх, також просто, а й дозволяє модернізувати її, додаючи нові умови.

При усьому цьому, треба розуміти, що вибір інших варіантів є теж можливий, бо кожен з них має свою особливість, плюси та мінуси.

Приклад виконання XML для рецептурного керування має бути архітектурною скомпонованим для чіткої реалізації й розуміння. Даний приклад описано нижче, на рисунку 3.7.



The image shows a screenshot of a text editor window titled "try.xml — Блокнот". The menu bar includes "Файл", "Правка", "Формат", "Вид", and "Справка". The main content area displays the following XML code:

```
<recipes>
  <recipe>
    <id> first </id>
    <control>
      <type> adding </type>
      <timeadd> 90 </timeadd>
      <whatadd> sugar </whatadd>
      <timecook> 300 </timecook>
      <tempcook> 100 </tempcook>
    </control>
  </recipe>
</recipes>
```

Рисунок 3.7 - Приклад зберігання рецептури в XML

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		





Trace Mode – інструментальний програмний комплекс класу SCADA НМІ. Призначений для розробки програмного забезпечення, систем телемеханіки, автоматизації будівель, систем обліку електроенергії, води, газу, тепла, а також для забезпечення їх функціонування в реальному часі.

Наглядний приклад реалізації через Trace Mode надано на рисунку 3.8.

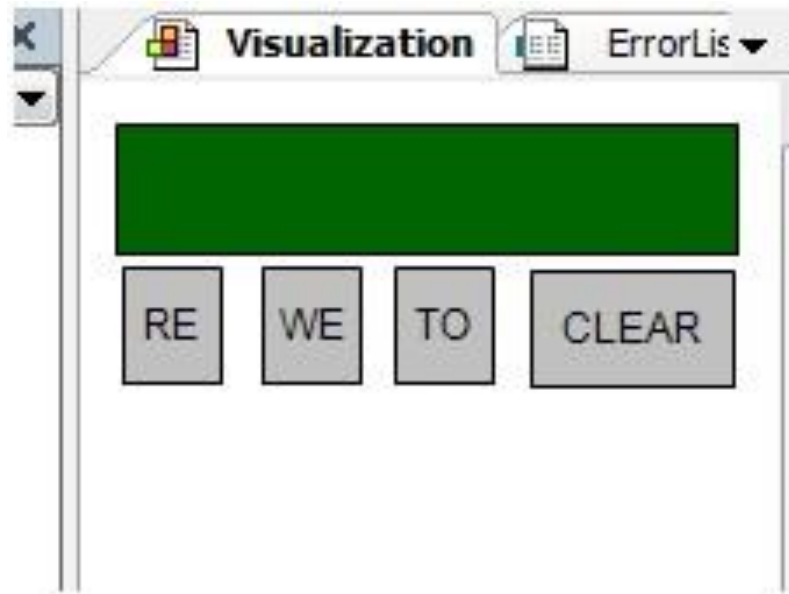


Рисунок 3.8 – Приклад інтерфейсу місця оператора

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		



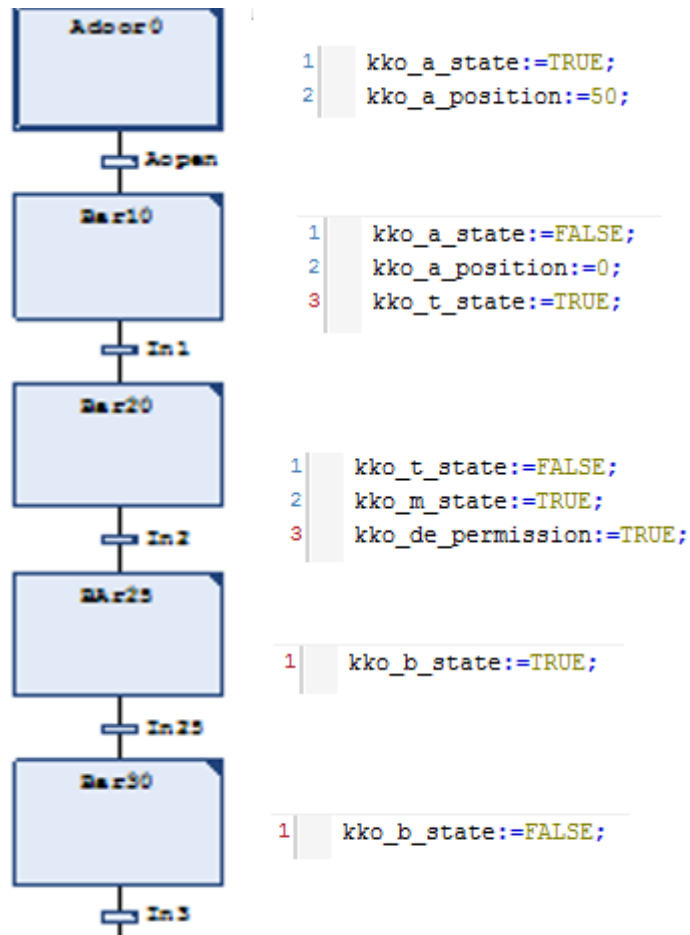


Рисунок 3.9 – Приклад програми для рецептурного керування мовою SFC та ST

Окрім SFC, в блоках виконання використовується стандартна мова для CoDeSys – ST (структурний текст), що і є тілом усього програмного коду.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

## Висновки до розділу

На етапі моделюванні вже вибрано необхідні засоби, інструменти, специфікації та протоколів.

При правильному збори схеми, отримується більш конкретна мета кожної частини. Опис процесу, дозволяє проглянути процес виробництва з декількох точок зору й оцінити рентабельність думок та рішень.

Так моделювання структури дає зрозуміти, що проект має можливість розвиватись як і внутрішньо – замінюючи просте зберігання в локальному виді на централізований простір серверу, чи обновляючи програмне забезпечення для покращення процесу, так і зовнішньо.

Зовнішні вдосконалення мають на меті пов'язати нові функціонали, або розширити межі старих, завдяки підключені нових елементів структури, як прикладом зворотній зв'язок для більшого вдосконалення рецептур.

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					

## 4 ПРИКЛАД РЕАЛІЗАЦІЇ РЕЦЕПТУРНОГО КЕРУВАННЯ ВИРОБНИЧОГО ПРОЦЕСУ

### 4.1 Опис та зберігання рецептури в XML файлі

Основним нововведенням в систему є рецептура, кожна яка має своє значення для контролерів. Для зберігання було обрано XML файл, через загальну практичність та просту в редагуванні.

Але при цьому, файл стає малоефективним на більш прогресивному виробництві, де процес пов'язан не лише з одним цехом. Тому у майбутньому систему з локальним файлом змінить вже прогресивна база даних, чи хмарова система на HTML.

Далі наведено частини XML файлу:

```
<recipes>  
  <recipe>  
    <id>recipe1</id>
```

Рисунок 4.1 – Об'явлення нового рецепту

Попередньо описана частка, дає заголовок рецепту, завдяки якому ми зможемо знайти його через інтерфейс. В збірнику рецептів може зберігатись безліч одиниць рецептів. Відзначається кожен своїм індивідуальним ім'ям чи номером.

```
<ingredient>  
  <id>acid</id>  
  <timeStart>30</timeStart>  
  <weight>60000</weight>  
</ingredient>
```

Рисунок 4.2 – Об'явлення нового інгредієнту

Ця частина дає необхідну частину й стосується виробничої клітини дозатора. Так при старті процесу, на тридцятій хвилині, в резервуар подається шістдесят тисяч грам (шістдесят кілограм) сірчаної кислоти.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

Це значення достаньте для стандарту 1200 грамів на літр дистильованій води, так як басейн резервуара завжди буде наповнений до п'ятдесяти літрів води.

```
<controls>
  <control>
    <type>boling</type>
    <timeStart>15</timeStart>
    <during>15</during>
    <anodFg>220</anodFg>
  </control>
  <control>
    <type>anodizing</type>
    <timeStart>60</timeStart>
    <during>45</during>
    <electric>10</electric>
    <anodFg>68</anodFg>
  </control>
</controls>
```

Рисунок 4.3 – Об'явлення основних характеристик

Зважаючи на алгоритм дій, можливо помітити, що процес нагріву (boling), хоч описано пізніше, але за часом подій є більш первинним. Це також стосується й інших речей та взагалі поняття зберігання в XML. Файл не запускає ніяких процесів, єдине що він зберігає це кількісну характеристику, яка вже потім знадобиться в процесі.

Більш важливішим для XML становиться структура. Ієрархічність будови додає простоту в розумінні рівня інформації та, що воно дає. Так розділ процеси впливають лише на агрегат, а саме резервуар з дистильованою водою. Кожен «метод» зберігає в собі інформацію, що в сумісності з алгоритмом контролерів, запускають процес виробництва.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

```

<controls>
  <control>
    <type>boling</type>
    <timeStart>15</timeStart>
    <during>15</during>
    <anodFg>220</anodFg>
  </control>
  <control>
    <type>anodizing</type>
    <timeStart>60</timeStart>
    <during>45</during>
    <electric>10</electric>
    <anodFg>68</anodFg>
  </control>
</controls>

```

Рисунок 4.3 – Об’явлення основних характеристик

Зважаючи на алгоритм дій, можливо помітити, що процес нагріву (boling), хоч описано пізніше, але за часом подій є більш первинним. Це також стосується й інших речей та взагалі поняття зберігання в XML. Файл не запускає ніяких процесів, єдине що він зберігає це кількісну характеристику, яка вже потім знадобиться в процесі.

Більш важливішим для XML становиться структура. Ієрархічність будови додає простоту в розумінні рівня інформації та, що воно дає. Так розділ процеси впливають лише на агрегат, а саме резервуар з дистильованою водою. Кожен «метод» зберігає в собі інформацію, що в сумісності з алгоритмом контролерів, запускають процес виробництва.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

```
</ingredient>
</recipe>
```

Рисунок 4.4 – Завершення об'явлення елемента

Окремим нюансом є завершення. В випадку XML та його «класів» знак «/» буде означати кінець ітерації. Її важливість є першочерговою, бо саме ітерація створює рамки елементів, налагоджує ієрархічну будову, по якій систематично краще будувати базу значень.

```
</recipe>
<recipe>
  <id>recipe2</id>
  <controls>
    <control>|
    . . .
  </control>
</controls>
  <ingredient>
  . . .
</ingredient>
</recipe>
</recipes>
```

Рисунок 4.5 – Об'явлення нового рецепту

Новий рецепт будить мати такий же процес та кількість параметрів. Це пов'язано з тим, що кожна рецептура є об'єктом кореневого образу, що зберігається в XML. Це пов'язано з тим, що CoDeSys, буде зчитувати лише чітку константу кількість змінних. При спробах додати зміну, програма буде ігнорувати її або доповнить, але при цьому буде необхідно переписати увесь файл.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

## 4.2 Програма для контролерів

У цій частині й прописано алгоритм дій виробництва, який заповнюється результатами з контролерів й вхідними даними з рецептури і є серцем проектування.

```
PROGRAM PLC_PRG
VAR
    StartBoil:TIME;
    DarBoil:TIME;
    DarIn:TIME;
    DarAnod:TIME;
    Nbte: REAL;
    Nate: REAL;
    Te: REAL;
    In:BOOL;
    Inf:REAL;
    Ninf:REAL;
    Ele:REAL;
    Nele:Real;
END_VAR
```

Рисунок 4.6 – Оголошення змінних

Перед початком утворення алгоритму йде створення для змінних комірки пам'яті, що описані зверху.

Маючи в програмній частині динамічні зміни, які витікають від результатів контролера: час, температура, сила поверхневого току, кількість речовини.

Разом з цим до них прикріплюються очікувані зміни, що задаються рецептурою та обмежують їх значення, утворюючи переходи між діями.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

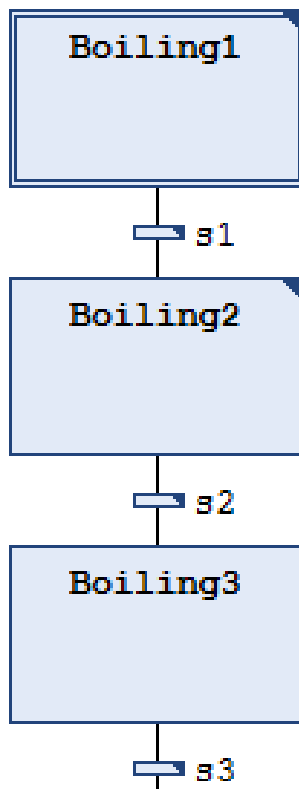


Рисунок 4.7 – Опис першої виробничої комірки

Процес починається з того, що ми доводимо температуру до стану кипіння, для того щоб потім додати кислоту:

```

1   Te:=77;
2   In:=FALSE;
3   Inf:=0;
4   Ele:=0;
-----
1   Boiling1.t>=StartBoil
-----
1   Te:=Te+1;
-----
1   Te>=Npte

```

Рисунок 4.8 – Програма виконання нагрівання

На початку вводяться початкові умови, при яких процес має право запуснитись. Після часового відліку, контролерам дозволяється почати нагрів, допоки значення температури не дійде до рецептурного мінімуму.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

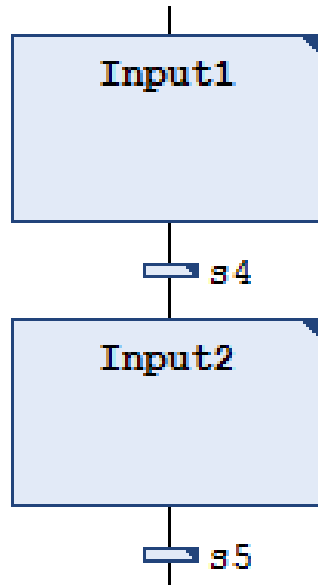


Рисунок 4.9 – Опис другої виробничої комірки

Процес другої частини зав'язан на додаванні сірчаної кислоти у воду та отримання готового розчину для анодування.

```

1  In:=TRUE;
2  Inf:=Inf+50;
1  Inf>=NInf
1  In:=false;
1  Input2.t>=DarIn

```

Рисунок 4.10 – Програма виконання додавання

Після дозволу, відкривається клапан, що буде подавати кислоту 50 грамів за один період. Після достатнього насичення, клапан закривається, а розчин настоюється ще декілька хвилин.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

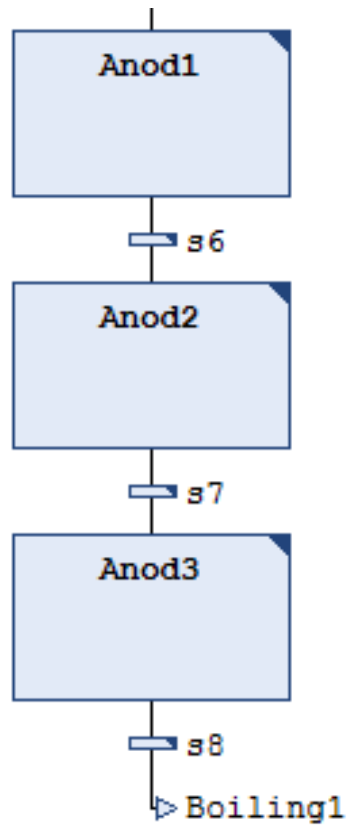


Рисунок 4.11 – Опис третьої виробничої комірки

При повній готовності, починається анодування зміним током, з послідовно змінням температури.

```

1  Ele:=Ele+1;
1  Ele=Nele;
1  Te:=Te-1;
1  Te=Nate
1  Anod3.t>=DarAnod

```

Рисунок 4.12 – Програма виконання анодування

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

Остання частина починається з підвищення поверхневого току. При досяганні потрібній величини, йде кореляція температури для зменшення можливості пошкодження обладнання. Одночасно з цим йде таймер, за яким і йде анодування до кінцевого процесу.

Останнім кроком є повернення в початкове положення, де система не має току, має початкову температуру й відсутній вже опрацьований розчин з сірчаною кислотою. Після цього процес анодування може починатись заново.

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					

### 4.3 Імпорт файлу XML

Список текстів - це об'єкт, який управляється в вікні POU або присвоєно додатком у вікні Пристрої. Він є основою:

а) багатомовними підтримки для статичних або динамічних текстів і підказок у візуалізації і в управлінні тривогами;

б) обміну динамічним текстом.

Списки текстів можна експортувати і (пере) імпортувати. Експорт необхідний, якщо Цільовий візуалізації потрібно мовної файл xml-формату.

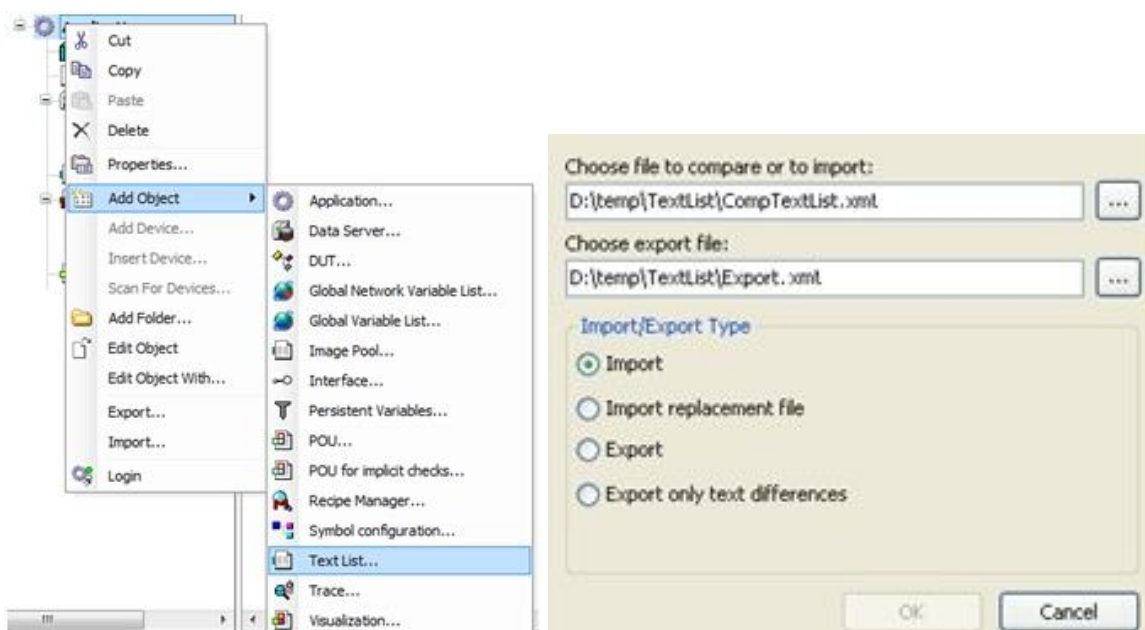


Рисунок 4.13 – Створення та імпортування текстового файлу в CoDeSys

Імпорт / експорт списків текстів - команда забезпечує обмін даними з іншими програмами, такими як, наприклад, Excel, XML.

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

#### 4.4 Програмування взаємозв'язку між контролерами через MODBUS

Маппінг – це процес встановлення відповідності між каналами введення/виведення та внутрішніми змінними програми.

Основним способом передачі інформації між контролерами є протокол MODBUS.

На початку до проекту доповнюється нові пристрої, що пов'язані з даним протоколом: Modbus\_COM, Modbus\_Master\_COM\_Port, Modbus\_Slave\_COM\_Port. Приклад показано на рисунку 4.14.

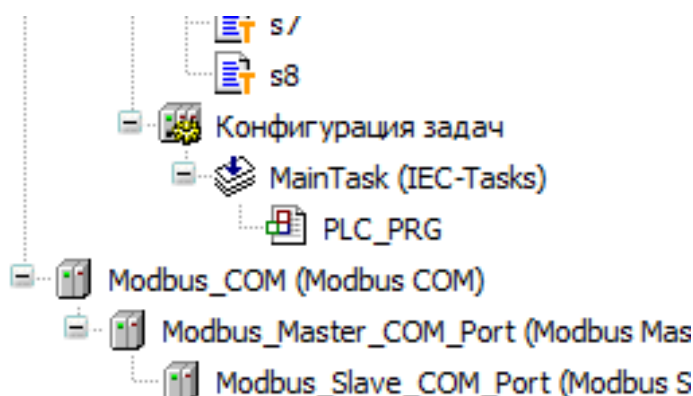


Рисунок 4.14 – Підключення MODBUS портів

Після цього, в розділі каналів, створюються самі канали. Вони зроблені для того, щоб переносити інформацію по контролерами. Приклад зображено на рисунку 4.15.

Общее		Имя	Тип доступа	
Канал Modbus Slave		0 StartBoil	Read Discrete Inputs (Код функции 02)	I
Modbus Slave Init		1 DarBoil	Read Discrete Inputs (Код функции 02)	I
ModbusGenericSerialSlave Соотнесение входов/выходов		2 DarAdd	Read Discrete Inputs (Код функции 02)	I
ModbusGenericSerialSlave IEC Objects		3 DarAnod	Read Discrete Inputs (Код функции 02)	I
Состояние		4 NBTe	Read Discrete Inputs (Код функции 02)	I
Информация		5 NATe	Read Discrete Inputs (Код функции 02)	I
		6 Ninf	Read Discrete Inputs (Код функции 02)	I
		7 Nele	Read Holding Registers (Код функции 03)	I
		8 Te	Write Multiple Coils (Код функции 15)	I
		9 Inf	Write Multiple Coils (Код функции 15)	I
		10 Ele	Write Multiple Coils (Код функции 15)	I

Рисунок 4.15 – Утворення каналів маппінгу

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

Після цього йде прив'язка значень з програм контролерів та змінних мапінгу, що й утворює повноцінно працюючий канал. Приклад на рисунку 4.16.

Общие		Найти переменную		Фильтр	П
		Переменная	Соотнесение	Канал	
Канал Modbus Slave		Application.PLC_PRG.StartBoil	↔	StartBoil	
Modbus Slave Init		Application.PLC_PRG.DarBoil	↔	DarBoil	
ModbusGenericSerialSlave	Соотнесение входов/выходов	Application.PLC_PRG.DarIn	↔	DarAdd	
ModbusGenericSerialSlave IEC Objects		Application.PLC_PRG.DarAnod	↔	DarAnod	
Состояние		Application.PLC_PRG.Nbte	↔	NBTe	
Информация		Application.PLC_PRG.Nate	↔	NATe	
		Application.PLC_PRG.Ninf	↔	Ninf	
		Application.PLC_PRG.Nele	↔	Nele	
		Application.PLC_PRG.Te	↔	Te	
		Application.PLC_PRG.Inf	↔	Inf	
		Application.PLC_PRG.Ele	↔	Ele	

Рисунок 4.16 – Перенесення змінних з програми в мапінг

При подальшому, повну роботу буде оцінюватись зеленим сигналом. Приклад є на рисунку 4.17.

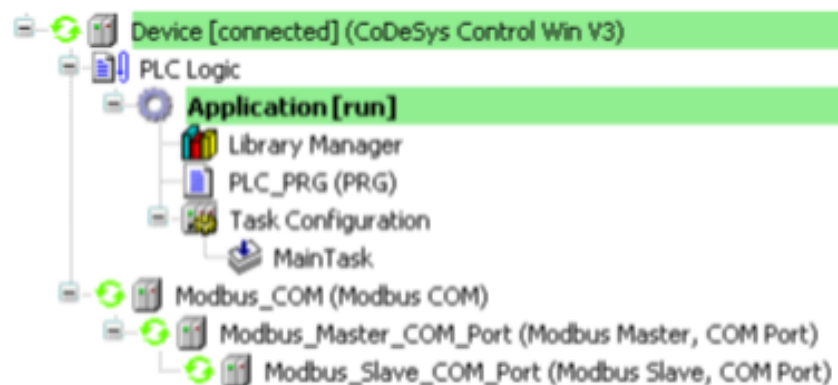


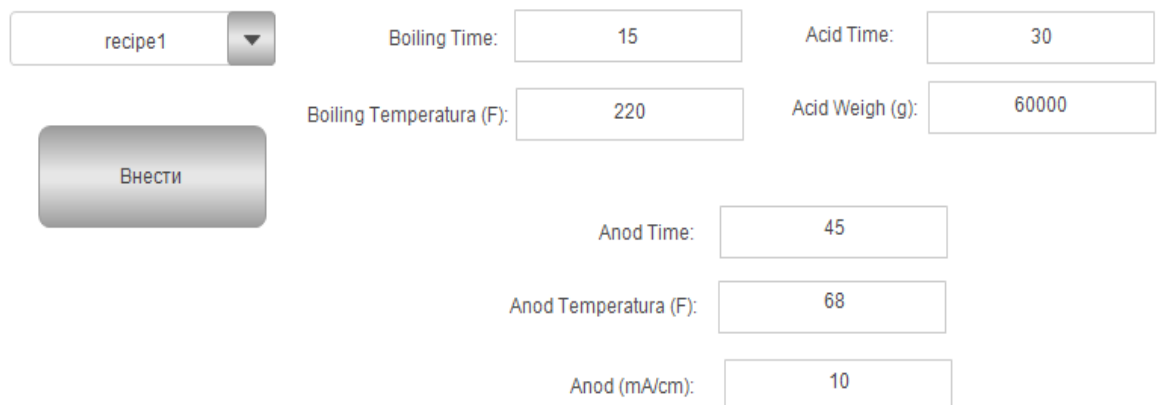
Рисунок 4.17 – Зображення правильної активності MODBUS

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					

## 4.5 Результати виконання

```
<recipe>
  <id>recipe1</id>
  <controls>
    <control>
      <type>anodizing</type>
      <timeStart>60</timeStart>
      <during>45</during>
      <electric>10</electric>
      <anodFg>68</anodFg>
    </control>
    <control>
      <type>boling</type>
      <timeStart>15</timeStart>
      <during>15</during>
      <anodFg>220</anodFg>
    </control>
  </controls>
  <ingredient>
    <id>acid</id>
    <timeStart>30</timeStart>
    <weight>60000</weight>
  </ingredient>
</recipe>
<recipe>
```

Рисунок 4.18 – Дані рецепта з файлу



recipe1

Внести

Boiling Time: 15

Acid Time: 30

Boiling Temperatura (F): 220

Acid Weigh (g): 60000

Anod Time: 45

Anod Temperatura (F): 68

Anod (mA/cm): 10

Рисунок 4.19 – Дані рецепта в інтерфейсі

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

## Висновки до розділу

Було реалізовано приклад системи рецептурного керування. Саме виконання ґрунтувалось на знаннях мов ST та SFC. Підкріплювалось це використанням додаткових протоколів та дійових об'єктів.

Сам проект більш розкинувся завдяки використанню XML файлу для збирання та зберігання рецептури.

При цьому, на жаль система вийшла примітивною, без можливості оновлювати рецептуру. Але при цьому дану особливість можна додати. Тай сам проект може без всяких проблем.

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					

## ЗАГАЛЬНІ ВИСОНВКИ

Насамперед проект показав універсальність даного підходу до покращення систем. Так як система рецептурного керування не має негативних ефектів, воно є стандартним на більшості типів підприємств.

Було виконано дослідження рецептурного керування, на прикладах різних цехів, з різних галузях. Дана концепція істотно розвивається і є привичною в середовищі підприємств.

Основною частиною обговорення було вибір місця зберігання рецептури, бо в залежності від рівня підприємства, її задача, амортизованості та рівню доходу, можна взяти різні оптимальні варіанти.

В випадку проекту, де в більшості є вивчення даних систем, при виконанні було змінено лише основні деталі ручного керування на основні рецептурного керування.

При наступних діях, можна змінити файл на базу даних, що може централізувати процес рецептурного керування, а також зменшити навантаження на саму систему.

Оцінюючи результати, можна сказати, що на даний момент утворено скелет системи рецептурного керування, який можна доповнювати та модернізувати під особливі задачі підприємства, чи збільшення масштабів підприємства.

Отже, система рецептурного керування є універсальною, можливо доповнювати проект без повного переписування та модернізується з кожним разом .

									Аркуш
Зм	Арк.	№ докум.	Підпис	Дата					

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Зм	Арк.	№ докум.	Підпис	Дата	Аркуш

ДОДАТОК А

```
<recipes>
  <recipe>
    <id>recipe1</id>
    <controls>
      <control>
        <type>anodizing</type>
        <timeStart>60</timeStart>
        <during>45</during>
        <electric>10</electric>
        <anodFg>68</anodFg>
      </control>
      <control>
        <type>boling</type>
        <timeStart>15</timeStart>
        <during>15</during>
        <anodFg>220</anodFg>
      </control>
    </controls>
    <ingredient>
      <id>acid</id>
      <timeStart>30</timeStart>
      <weight>60000</weigth>
    </ingredient>
  </recipe>
</recipes>
```

						Аркуш
Зм	Арк.	№ докум.	Підпис	Дата		

