

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Інформаційних правляючих систем  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження моделей та методів інтеграції ІТ-сервісів інформаційної системи в бізнес-транзакцію  
(тема)

Виконав:  
студент 2 курсу, групи ІУСТМ-20-1

(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі системи та технології

(повна назва освітньої програми)

Керівник проф. Євланов М. В.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Петров К.Е.  
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
Кафедра Інформаційних управляючих систем  
Рівень вищої освіти другий (магістерський)  
Спеціальність 122 Комп'ютерні науки  
(код і повна назва)  
Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)  
Освітня програма Інформаційні управляючі системи та технології  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)  
«\_\_\_\_» \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Штонденко Анастасії Олександрівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження моделей та методів інтеграції ІТ-сервісів інформаційної системи в бізнес-транзакцію

затверджена наказом університету від «08» листопада 2021 р. № 1645См

2. Термін подання студентом роботи до екзаменаційної комісії «08» грудня 2021 р.

3. Вихідні дані до роботи Науково-технічні публікації та інтернет джерела з тематики кваліфікаційної роботи, опис організації як об'єкта управління (підприємство важкого машинобудування).

4. Перелік питань, що потрібно опрацювати в роботі аналіз загальних відомостей про інтеграцію іт-сервісів інформаційної системи в бізнес-транзакцію, аналіз структури взаємодії ІТ-сервісів з бізнес-транзакціями, вивчення типів інтеграції в інформаційних системах на рівні бізнес-транзакції та проблем реалізації інтеграції бізнес-транзакцій, Теоретичне вирішення задачі «інтеграції іт-сервісів в бізнес-транзакції», реалізація побудови бізнес-транзакції за допомогою «САГА», побудова сценарію задачі «Обліку конструкторської документації», розробка бізнес-транзакції «Обліку конструкторської документації» за допомогою офарбованих мереж Петрі, використання теоретичних результатів з вирішення задачі інтеграції іт-сервісів в бізнес-транзакції, опис отриманих теоретичних та практичних досліджень поставленої задачі.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз літератури та Інтернет-джерел	08.11.21	
2	Постановка задачі	09.11.2021-11.11.2021	
3	Обробка матеріалу	12.11.2021-14.11.2021	
4	Дослідження сучасних відомостей про інтеграцію іт-сервісів інформаційної системи в бізнес-транзакцію	15.11.2021-20.11.2021	
5	Дослідження методів моделювання бізнес-транзакцій	21.11.2021-25.11.2021	
6	Розроба ІТ-сервісу обліку конструкторської документації побудованого на основі обраного методу моделювання бізнес-транзакції	26.11.2021-28.11.2021	
7	Опис отриманих теоретичних та практичних досліджень поставленої задачі	29.11.2021-30.11.2021	
8	Написання пояснювальної записки	01.12.2021-04.12.2021	
9	Підготовка презентації	05.12.2021-06.12.2021	
10	Перевірка на плагіат	06.12.2021-07.12.2021	
11	Нормоконтроль	07.12.2021-08.12.2021	
12	Захист	09.12.21	

Дата видачі завдання «08» листопада 2021 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Євланов М. В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка атестаційної роботи містить: 123с., 14 табл., 30 рис., 2 дод., 45 джерел.

БАЗА ДАНИХ, БІЗНЕС-ПРОЦЕС, БІЗНЕС-ТРАНЗАКЦІЯ, ДІАГРАМА ПОТОКІВ ДАНИХ, ІНЖЕНЕР-КОНСТРУКТОР, ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ, ІТ-СЕРВІС, КРЕСЛЕННЯ, МЕРЕЖА ПЕТРІ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, САГА, СУБД, СХЕМА ДАНИХ, СХЕМА ОРГАНІЗАЦІЙНОЇ СТРУКТУРИ, СПЕЦИФІКАЦІЯ, ФУНКЦІОНАЛЬНА СТРУКТУРА, ER-ДІАГРАМА, JAVA.

Об'єкт дослідження – методи інтеграції ІТ-сервісів в бізнес-транзакцію.

Мета роботи – дослідження інтеграції ІТ-сервісів в бізнес-транзакцію.

З метою дослідження методів інтеграції ІТ-сервісів в бізнес-транзакцію було розглянуто основні методи побудови бізнес-транзакцій та розроблено сервіс обліку конструкторської документації на підприємствах.

В ході кваліфікаційної роботи розглядалися такі питання: огляд сучасного стану проблем з моделювання бізнес-транзакцій; опис та аналіз структурних і функціональних особливостей підприємств машинобудування для розробки сервісу; постановка задачі кваліфікаційної роботи; обґрунтування мети вирішення поставленої задачі; опис та аналіз інформаційних технологій, які використовуються при вирішенні задачі; опис рішень з інформаційного забезпечення ІТ-сервісу; опис рішень з питання інтеграції ІТ-сервісу в бізнес-транзакцію; опис рішень з програмного забезпечення задачі інтеграції сервісу; опис рішень з технічного забезпечення задачі; вибір і розробка мовних засобів взаємодії користувачів задачі розробленого сервісу обліку конструкторської документації; опис методичних рекомендацій щодо використання інтегрованого ІТ-сервісу; опис рішення щодо впровадження результатів дослідження; опис необхідності подальшого розглядання досліджень.

## ABSTRACT

Explanatory note of attestation work: 123 pages, 14 tables, 30 pictures, 2 appendices, 45 sources.

DATABASE DESIGN ENGINEERS, DIAGRAM FLOWS DATA, BUSINESS TRANSACTIONS, BUSINESS PROCESSES, SAGA, PETRI NETS, DATABASES, IT SERVICES, INFORMATION SOFTWARE, DATA SCHEMA, ORGANIZATIONAL CHART, DRAWINGS, SOFTWARE, SPECIFICATIONS, FUNTSIONALNA STRUCTURE, ER-DIAGRAM, JAVA.

The object of research is methods of integrating IT services into a business transaction.

The purpose of the work is to study the integration of IT services into a business transaction.

In order to study the methods of integrating IT services into a business transaction, the main methods of building business transactions were considered and a service for accounting design documentation at enterprises was developed.

During the qualification work the following issues were considered: review of the current state of problems in business transaction modeling; description and analysis of structural and functional features of machine-building enterprises for service development; setting the task of qualification work; substantiation of the purpose of solving the task; description and analysis of information technologies used in solving the problem; description of decisions on information support of IT service; description of decisions on the integration of IT services in the business transaction; description of software solutions for the task of service integration; selection and development of language means of interaction of users of the task of the developed service of the account of the design documentation; description of methodical recommendations on the use of integrated IT service; description of the decision to implement the results of the study; description of the need for further research

research

## ЗМІСТ

Скорочення та умовні позначки .....	8
Вступ.....	9
1 Загальні відомості про інтеграцію іт-сервісів інформаційної системи в бізнес-транзакцію .....	11
1.1 Загальні відомості інформаційної системи. ....	11
1.1.1 Визначення інформаційної системи.....	11
1.1.2 Типологія інформаційних систем.....	12
1.2 Поняття бізнес-транзакцій та їх властивостей.....	15
1.3 Структура взаємодії ІТ-сервісів з бізнес-транзакціями. ....	19
1.4 Типи інтеграції в інформаційних системах на рівні бізнес-транзакції. ....	22
1.5 Проблеми реалізації інтеграції бізнес-транзакцій .....	26
1.6 Задачі проектування та розробки бізнес-транзакцій .....	31
2 Теоретичне вирішення задачі «інтеграції іт-сервісів в бізнес-транзакції»	33
2.1 Реалізація побудови бізнес-транзакції за допомогою шаблону «САГА» .	33
2.1.1 Побудова сценарію задачі «Обліку конструкторської документації»	33
2.1.2 Використання концепції шаблону «САГА».....	37
2.2 Розробка бізнес-транзакції «Обліку конструкторської документації» за допомогою офарбованих мереж Петрі.....	40
2.2.1 Моделювання бізнес-транзакції за допомогою мережі Петрі.....	43
2.2.2 Конструкція сервісної взаємодії.....	46
2.2.3 Переваги використання офарбованих мереж при інтеграції сервісів у бізнес-транзакції.....	47
3 Використання теоретичних результатів з вирішення задачі інтеграції іт-сервісів в бізнес-транзакції .....	49
3.1 Постановка задачі реалізації інтеграції ІТ-сервісу у бізнес-транзакцію...	49
3.2 Опис сценарію бізнес-транзакції обліку конструкторської документації.	50
3.2.1 Опис об'єкту дослідження та постановка основних задач поставлених до бізнес-транзакції.....	50

3.3 Побудова сутностей задачі.....	64
3.3.1 Опис атрибутів сутностей і їх доменів .....	66
3.4 Опис схеми даних задачі .....	72
3.5 Висновки практичного рішення інтеграції ІТ-сервісів в бізнес-транзакції реалізованого власним методом. ....	75
4 Опис отриманих теоретичних та практичних досліджень поставленої задачі	
77	
4.1 Обґрунтування необхідності використання інтегрованих сервісів.....	77
4.1.1 Опис технічних характеристик необхідних для використання ІТ-сервісів.....	77
4.1.2 Вимоги до взаємодій користувачів задачі.....	83
4.1.3 Методичні рекомендації щодо застосування сервісу інтегрованого під час практичного проектування. ....	84
4.2 Підсумкова характеристика практичного використання ІТ-сервісу інтегрованого в бізнес-транзакцію. ....	85
Висновки .....	87
Перелік джерел посилань .....	89
Додаток А Вихідні та вхідні повідомлення задачі «Облік конструкторської документації», яку обрано для розробки ІТ-сервісу.....	92
Додаток Б Графічний матеріал атестаційної роботи .....	103

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

- БД – база даних;
- БП – бізнес процес;
- БТ – бізнес-транзакція;
- ІС – інформаційна система;
- ІТ – інформаційні технології;
- КД – конструкторська документація;
- НКМЗ – новокраматорський машинобудівний завод;
- ОС – операційна система;
- ПЗ – програмне забезпечення;
- ПК – персональний комп'ютер;
- СУБД – система управління базами даних;
- ТОВ – товариство з обмеженою відповідальністю;
- ACL – Access Control List;
- API – Application Programming Interface;
- DSS – система підтримки прийняття рішень;
- DFD – Data Flow Diagram;
- DML – Data Manipulation Language;
- ESS – система підтримки керівництва;
- ER – Entity relationship;
- ESB – enterprise service bus;
- EDM – Erwin Data Model;
- GUI – Graphical User Interface;
- IDEF3 – Integrated Definition for Process Description Capture Method;
- MIC – системи управлінської інформації;
- TPS – система обробки транзакцій;
- UML – Unified Modeling Language.
- XML – eXtensible Markup Language;

## ВСТУП

Зростаюча потреба у міжкорпоративній бізнес-взаємодії, що відбувається швидко без втручання людини, призвела до прискореного створення та доступності веб-сервісів, що сьогодні є основною тенденцією в інтернеті. Замість традиційних потокових моделей, що вимагають виділених мереж між компаніями з високою вартістю та обмеженою масштабованістю та можливістю повторного використання.

Інтеграція на рівні бізнес-транзакції розрізняється залежно від рівня інтеграції. При внутрішній інтеграції взаємодіє велика кількість сервісів, а при зовнішній інтеграції – в основному два. Головним призначенням процесу інтеграції є об'єднання функцій додатків або модулів для надання нової функціональності. При інтеграції додатків можна виділити два основні типи завдань:

- 1) корпоративних застосунків;
- 2) додатків з різних ІС.

Для вирішення завдань першого типу застосовують системи ЕАІ, які іноді називаються А2А (Application – to – Application Integration), а для вирішення завдань другого типу застосовуються системи В2В (Business – to – Business Integration). У деяких випадках складно визначити різницю між інтеграцією А2А і В2В, оскільки складність деяких рішень усередині ІС може перевищувати складність рішень для їх спільного функціонування.

Беручи за об'єкт розробки ІТ-сервісів для обліку документацій машинобудівних підприємств можемо чітко зазначити, що підстава до інтеграції сервісів в бізнес-транзакції у наш час має велику актуальність оскільки як зазначено раніше потреба у міжкорпоративній бізнес-взаємодії даних структур має велике значення як і для атоматизації виробничих процесів так і для покращення існуючих взаємодій у сфері обліку документації. Сучасний стан подібних структур не відповідає світовим тенденціям, вимагає

розробки практичних рішень обумовлених необхідністю вдосконалення існуючих систем для автоматизації бізнес-процесів, що неможливо покращити без використання подібного роду досліджень. Слід зауважити, що у наш час необхідність переходу підприємств на більш сучасні принципи роботи сприятимуть лише покращенню індустрії в тих або інших галузях.

Мета даної кваліфікаційної роботи полягає в дослідженні методів інтеграції ІТ-сервісів в бізнес-транзакції.

Предметом дослідження є методи моделювання бізнес-транзакцій за допомогою підходу «САГА» та мереж Петрі.

Використовуючи данні з отриманих досліджень ініціюється розробка ІТ-сервісу обліку конструкторської документації побудованого на основі обраного методу моделювання бізнес-транзакції, та розробленого з використанням сучасних рішень web індустрії.

# 1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ІНТЕГРАЦІЮ ІТ-СЕРВІСІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ В БІЗНЕС-ТРАНЗАКЦІЮ

## 1.1 Загальні відомості інформаційної системи.

### 1.1.1 Визначення інформаційної системи.

Інформаційна система (ІС) – це набір взаємопов’язаних компонентів, які збирають, маніпулюють, зберігають і поширюють інформацію та забезпечують механізм зворотного зв’язку для досягнення мети. Механізм зворотного зв’язку допомагає організаціям досягати своїх цілей за рахунок збільшення прибутку, покращення обслуговування клієнтів [1] та підтримки прийняття рішень і контролю в організаціях [2].

Компанії використовують інформаційні системи для збільшення доходів і зниження витрат.

В організаціях інформаційні системи структуруються навколо чотирьох основних елементів, запропонованих у 1960-х роках Гарольдом Лівіттом зображених на рисунку 1.1. Рисунок відомий як «Діамант Лівітта».

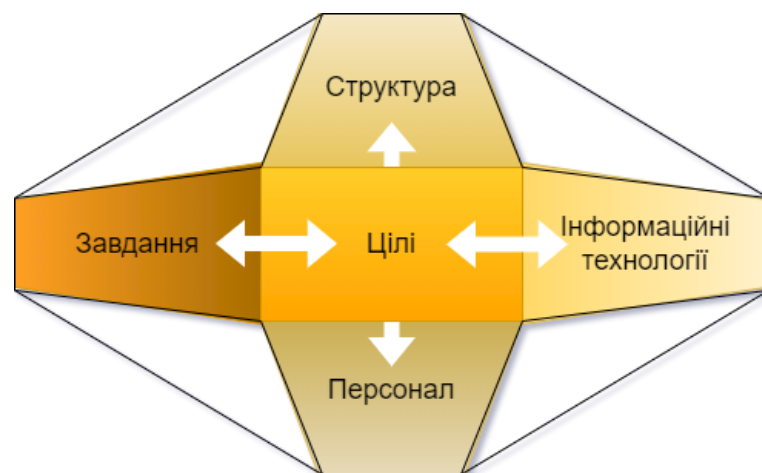


Рисунок 1.1 – Графічне зображення «Діамант Лівітта»

Технологія: ІТ (інформаційні технології) ІС включає в себе апаратне, програмне та телекомунікаційне обладнання, що використовується для збору, обробки, зберігання та поширення інформації. Сьогодні більшість ІС базуються на ІТ, оскільки сучасні ІТ дають змогу ефективно виконувати операції та ефективно керувати будь-якими розмірами.

Завдання: діяльність, необхідна для виробництва товару чи послуги. Ці заходи підтримуються потоком матеріалу, інформації та знань між різними учасниками [2].

Персонал: компонент людей інформаційної системи охоплює всіх людей, безпосередньо залучених до системи. До таких людей належать менеджери, які визначають цілі системи, користувачі та розробники.

Структура: компонент організаційної структури та інформаційних систем відноситься до взаємовідносин між окремими людьми, компонентами. Таким чином, він охоплює ієрархічні структури, відносини та системи оцінки людей.

### 1.1.2 Типологія інформаційних систем.

Компанія має системи підтримки різних управлінських рівнів. Ці системи включають системи обробки транзакцій, системи управлінської інформації, системи підтримки прийняття рішень і спеціальні системи бізнес-аналітики.

Компанії використовують інформаційні системи, щоб у разі потреби була доступна точна та актуальна інформація [2].

В межах однієї організації керівники на різних рівнях ієрархії мають дуже різні вимоги до інформації, різні типи інформаційних систем розвивалися, щоб задовольнити їхні потреби. Загальноприйнятим підходом до вивчення типів інформаційних систем, що використовуються в організаціях, є

їх класифікація відповідно до їхньої ролі на різних рівнях організаційної структури, цей підхід називається вертикальним підходом. Справді, організація вважається пірамідою управління на чотирьох рівнях(Рисунок 1.2).

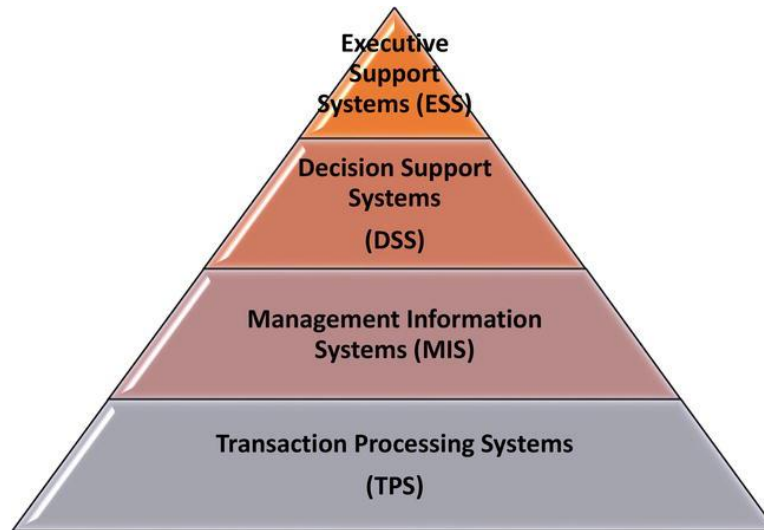


Рисунок 1.2 – Зображення чотирьох рівнів управління

На найнижчому рівні персонал виконує рутинні повсякденні операції, такі як продаж товарів і видача квитанцій про оплату. Операційне управління, при якому менеджери відповідають за нагляд за контролем транзакцій і вирішення проблем, які можуть виникнути. Тактичний менеджмент, який має прерогативу прийняття рішень щодо бюджетів, встановлення цілей, визначення тенденцій та планування короткострокової ділової діяльності. Стратегічне управління відповідає за визначення своїх довгострокових цілей і позиціонування щодо своїх конкурентів або галузі.

Розберемо ці рівні детальніше.

Система обробки транзакцій (TPS) – на операційному рівні менеджерам потрібні системи, які відстежують організацію для необхідних дій та операцій, таких як продажі та матеріальний потік на заводі. Система обробки транзакцій – це комп’ютерна система, яка виконує та записує рутинні (щоденні) операції, необхідні для управління справами, такі як ведення обліку працівників,

нарахування заробітної плати, відвантаження товарів, ведення записів, бухгалтерського обліку та казначейства.

На цьому рівні головне призначення систем – відповідати на рутинні запитання та контролювати потік транзакцій через організацію.

На оперативному рівні завдання, ресурси та цілі заздалегідь визначені та добре структуровані. Рішення про надання кредиту клієнту, наприклад, приймається головним наглядовим органом відповідно до заздалегідь визначених критеріїв. Все, що потрібно визначити, це те, чи відповідає клієнт критеріям.

Системи управлінської інформації (MIS) – менеджерам середньої ланки потрібні системи, які допомагають у нагляді, контролі, прийнятті рішень та адміністративній діяльності. Головне питання, на яке має відповісти такий тип системи: чи все працює правильно?

Його роль полягає в узагальненні та звітуванні про основні бізнес-операції з використанням даних, наданих системами обробки транзакцій. Дані первинної транзакції синтезуються та агрегуються, і зазвичай вони представлені у звітах, які регулярно створюються.

Системи підтримки прийняття рішень (DSS) – підтримує прийняття рішень щодо незвичайних питань, для яких не існує повністю визначених процедур. Цей тип системи намагається відповісти на такі запитання, як: що вплине на графіки виробництва, якщо ми подвоїмо продажі в грудні? Яким був би рівень рентабельності інвестицій, якби графік роботи заводу був відкладений більш ніж на шість місяців?

Хоча DSS використовують внутрішню інформацію від систем TPS і MIS, вони також використовують зовнішні джерела, такі як котирування акцій або ціни на продукти конкурентів. Ці системи використовують різноманітні моделі для аналізу даних. Система може відповісти на такі запитання, як: враховуючи графік доставки клієнта та пропонувану ставку фрахту, яке судно слід призначити та який рівень заповнення для максимізації прибутку? Яка

оптимальна швидкість, з якою судно може максимізувати прибуток, дотримуючись графіка доставки?

Система підтримки керівництва (ESS) – допомагає вищому керівництву приймати рішення. Вони стосуються виняткових рішень, які вимагають судження, оцінки та цілісного уявлення про бізнес-ситуацію, оскільки на цьому рівні немає процедури, якої слід дотримуватися для вирішення певного питання.

ESS використовує графіку та дані з багатьох джерел через інтерфейс, який легко розуміють старші менеджери. ESS призначений для інтеграції даних із зовнішнього середовища, таких як нові податки або дані конкурентів, а також для інтеграції сукупних даних з MIS і DSS. ESS фільтрує, синтезує та відстежує важливі дані. Особлива увага приділяється відображенню цих даних, оскільки це сприяє швидкому засвоєнню цих показників вищого керівництва. Все частіше ці системи включають інструменти аналізу бізнес-аналітики для визначення ключових тенденцій та прогнозів [3].

## 1.2 Поняття бізнес-транзакцій та їх властивостей.

Зростаюча потреба у міжкорпоративній бізнес-взаємодії, що відбувається швидко без втручання людини, призвела до прискореного створення та доступності веб-сервісів, що сьогодні є основною тенденцією в інтернеті. Замість традиційних потокових моделей, що вимагають виділених мереж між компаніями з високою вартістю та обмеженою масштабованістю та можливістю повторного використання, ІТ-сервіси є платформно-незалежними, слабо пов'язаними програмними компонентами, які забезпечують взаємодію між різномірними розподіленими компонентами. ІТ-сервіси це послуги, які стають доступними з сервера компанії для користувачів інших програм, підключених до мережі.

Бізнес-транзакція – це узгоджена зміна стану відносин двох і більше сторін, де кожна сторона готова до цієї зміни та знає, що її узгоджено приймуть усі сторони. З вимоги взаємної зміни стану випливає вимога атомарності, а виходячи з того, що зміна стану за масштабом може бути будь-яким – розмір бізнес-транзакції не обмежується [5].

Бізнес-транзакції виконують функції, які є критичними для бізнес-процесу, наприклад, управління ланцюжком поставок, і можуть координуватися безліччю незалежних партнерів, серед яких можуть бути великі відділи або навіть окремі підприємства [4]. Кожен учасник транзакції має власний стан, який узгоджено з іншими учасниками. Водночас учасники транзакції підтримують у певному стані спільне поле взаємин, яке відповідає розподіленим обмеженням, прийнятим між сторонами, і, отже, бізнес-транзакція є узгодженим переходом цих відносин до нового стану.

Тому бізнес-транзакція схожа зі звичайною транзакцією як семантично, яка є набором операцій, що переводять базу даних з одного стійкого стану в інший, так і технічно, вимагаючи для виконання певного середовища і протоколів, що підтримуються всіма учасниками. Але при цьому багато авторів зазначають, що бізнес-транзакції, на відміну від розподілених транзакцій СУБД, керуються не лише технічними вимогами, такими як координація, узгодженість даних, здатність до скасування та відновлення, а й економічними обмеженнями [6]. Наприклад, мета транзакції досягається лише тоді, коли всі учасники дійшли узгодженого рішення про те, що вона закінчена, наприклад, проведено остаточну оплату за повністю надані товари та послуги.

Так, до опису транзакції входять такі дані:

- сторони та виконавці транзакції;
- об'єкти, над якими провадиться транзакція, в т.ч. форми та документи, які обробляє транзакція;
- призначення транзакції (платежів, доставки та іншого);

- обмеження, які регламентують будь-які аспекти транзакції, зокрема, тимчасові обмеження (максимальний час, протягом якого транзакція може бути активною);

- бізнес-інваріанти – обмеження, зовнішні стосовно транзакції та її учасників, які висловлюють вимоги закону, правила торгівлі та терміни контракту, публічні, прийняті політики, закони та регулювання, які застосовуються для учасників транзакції;

- особливості зв'язку та взаємодії з іншими транзакціями;

- до технічної групи вимог відносять різні системні аспекти, пов'язані з виконанням транзакції та взаємодією її із зовнішнім середовищем.

Ці вимоги торкаються не тільки бізнес-транзакції, але і середовища їх виконання:

- підтримка тривалих, композитних взаємодій;

- визначення виняткових ситуацій та їх наслідків, включаючи послідовності відновлення початкового стану;

- підтримка компенсованих та відновлюваних транзакцій;

- використання альтернативних транзакцій для виконання тих самих функцій;

- можливість узгодження та взаємодії з іншими транзакціями;

- можливість підтримки різних моделей безпеки транзакцій: цілісності, конфіденційності, невідмовності, тощо;

- можливість здійснення аудиту, моніторингу та ведення журналу виконання транзакцій.

Через ці вимоги бізнес-транзакції можуть реалізовуватись як засобами інтеграції інформаційних систем, так і технологіями, що забезпечують комунікацію роз'єднаних та слабпов'язаних систем (наприклад, за допомогою технологій веб-служб [7], [9], [10], [11]).

Елементом системи є її проста неподільна частина. Як елемент системи можна виділити учасника господарської операції, який володіє багатьма функціями та послугами, а також окремими функціями та послугами, які

об'єднані в єдине ціле для виконання загального процесу, та приватні процеси, що виконуються учасниками. У той же час можливо, що кожен елемент господарської операції, яким би він не був, поза розглянутої бізнес-операції, може брати участь у вторинних взаємодіях, будучи частиною сторонньої системи. Загалом, для різних цілей розгляду господарської операції межа її поділу на елементи може бути різною, тому господарську операцію можна поділити, наприклад, на підсистеми, кожна з яких є самостійною частиною системи, що має властивості та підцілі системи. Поняття підсистеми використовується для багаторівневого поділу системи [6].

У господарських операціях залежно від їх структури підсистеми можуть бути, наприклад, учасниками зі своїм контролем або окремими службами з кількома функціями. Відтепер, залежно від контексту, вони будуть називатися елементами, підпроцесами або операціями бізнес-операції. Усі елементи господарської операції пов'язані між собою відносинами (зв'язками), які, з одного боку, визначають її структуру, а з іншого – поведінку та функціонування в цілому. Ланки обмежують ступінь свободи деталей. З технологічної точки зору, через властивості сервісно-орієнтованого середовища зв'язок між окремими учасниками, службами та операціями слабкий і спрямований. Однак з організаційної точки зору це важко класифікувати. Зв'язки, елементи та підсистеми господарської операції спостерігач (дослідник) повинен розрізняти залежно від характеру його ставлення до неї. Скажімо, що структура господарської операції представлена із сукупністю стійких зв'язків і зв'язків між її елементами, які описують пристрій (структуру) господарської операції. Залежно від характеру елементів, співвідношення між ними, цілей спостерігача і системи структура транзакції може мати різне втілення [12].

Наприклад, якщо функції служби виділені як елементи транзакції, то вони пов'язані відносинами передачі даних, які, у свою чергу, обмежені інтерфейсними контрактами. Не виключається можливість запровадження

ієрархії господарських операцій, включаючи інші підсистеми з власною структурою [8].

### 1.3 Структура взаємодії IT-сервісів з бізнес-транзакціями.

З моменту появи Інтернету підприємства відкрили свої основні функції для клієнтів, постачальників, ділових партнерів та фінансових установ. Інтенсивне зростання всесвітньої павутини створило можливості для всіх видів підприємств зробити свої цінні пропозиції доступними для споживачів як програмні послуги (тобто електронні послуги). Прикладом цього є поширення книжкових магазинів у мережі, які дозволяють користувачам Інтернету переглядати їхні каталоги, розміщувати замовлення та здійснювати платежі.

Проблема, спільна для акторів, які беруть участь у такій співпраці, полягає в тому, щоб визначити, які пропозиції вони мають зробити доступними як програмні послуги для інших. Ділове співробітництво між зацікавленими сторонами констеляцій цінності бізнесу можна описати за допомогою бізнес-моделей. Бізнес-модель створюється для того, щоб зрозуміти, хто є учасниками бізнес-транзакції, і пояснити їхні відносини, які формулюються в термінах економічних цінностей, якими обмінюються суб'єкти, або, точніше, ділові операції відбуваються між суб'єктами. Таким чином, бізнес-моделі фіксують бізнес-операції між акторами та події, які призводять до створення та розподілу цінностей між акторами.

Розробка, орієнтована на транзакції забезпечує основу для узгодження між бізнесом та IT, просуваючи роль транзакцій та автоматичного створення коду за допомогою попередньо визначених перетворень моделі. Сучасні рішення для веб-сервісів досягли успіху в узгодженні з бізнес-процесами, дозволяючи таким чином слабо інтегровані та багаторазові автоматизації

завдань – тут бізнес-інформація фіксується на процедурному, тобто тактичному рівні.

З технічної точки зору, веб-сервіси стали загальною технологією моделювання взаємодії веб-додатків. Поки що розвиток веб-сервісів був зосереджений на структурних та операційних аспектах. Розробка додатків безпосередньо для цих перспектив утомлива, схильна до помилок, а бізнес-функціональність залишається невидимою. Підвищення рівня абстракції, щоб відокремити бізнес-специфікації від деталей реалізації, є усталеною тенденцією в розвитку системи і є однією з головних цілей MDA, Model Driven Architecture, яка є особливим підходом до розробки систем, керованих моделлю, в парадигмі MDD. Одним з головних питань у дисципліні MDA є вибір типів моделей, які будуть використовуватися на різних рівнях абстракції. Згідно з роботами провідних авторів більшість існуючих пропозицій не використовують бізнес-вимоги як відправну точку. Вони також стверджують, що ті, хто навіть розглядає бізнес-вимоги, не використовують їх виключно для опису бізнес-вимог у контексті розробки, керованої моделлю (MDD), і, таким чином, служать вхідними для процесу трансформації, керованого моделлю [15].

У цьому розділі ми досліджуємо здатність моделей цінності бізнесу виявляти портфель бізнес-операцій на незалежному від обчислень, тобто на рівні бізнесу. Подальше вивчення таких бізнес-транзакцій протягом усього життєвого циклу співпраці, тобто від планування до постактуалізації, дає нам змогу отримати весь портфель послуг для всього підприємства на рівні бізнесу. На цьому ж рівні для опису поведінки служби використовуються моделі процесів. Щоб уможливити відображення виявлених бізнес-сервісів із програмними послугами на системному рівні, ми покладаємося на використання профілів UML як структурованого способу налаштувати фокус моделі на певний архітектурний стиль, наприклад, у цьому випадку – сервіс-орієнтований. Концептуалізований таким чином, метод, який ми пропонуємо, здатний підтримувати інтеграцію та узгодження пропозицій економічної

цінності бізнес-акторів, які співпрацюють, із інформаційними системами (ІС), створеними за допомогою веб-сервісів. Метод має практичне значення для більш глибокого вивчення моделей підприємства з точки зору бізнесу, щоб визначити програмні послуги та системи проектування відповідно.

Щоб полегшити окреслену бізнес-орієнтацію в інженерії веб-сервісів на основі моделі, ми визначили дві важливі потреби:

- використовувати адекватні моделі послуг та платформи моделювання. Послуги існують у спільному бізнес-середовищі, і як такі їх слід концептуалізувати відповідно;

- уможливити зв'язування бізнес- та програмно-сервісно-орієнтованих системних моделей більш точно та однозначно. Моделі можуть мати різні рівні абстракції, і, таким чином, потрібне правильне поширення інформації моделі від вищого рівня абстракції до нижчого.

Саме тому акцентом на вдосконалення є мета «Дослідження моделей та методів інтеграції ІТ-сервісів інформаційної системи в бізнес-транзакцію», також слід зауважити, що сучасні підприємства в умовах європейської конкуренції та економічної кризи прагнуть автоматизувати трудомісткі бізнес-процеси, використовуючи можливості пам'яті комп'ютерів, їх високу швидкість і зручний графічний інтерфейс для того, щоб комп'ютеризувати й об'єднати процеси проектування і виробництва, які раніше були ніяк не пов'язані один з одним, і створити єдину інформаційну систему (ІС) підприємства.

На ринку ІС технології вирішення конкретної задачі «інтеграція ІТ-сервісів інформаційної системи в бізнес-транзакцію» ІС машинобудівельного підприємства майже відсутні. Є значна кількість компаній, що надають інформаційні продукти, які пов'язані з автоматизацією обліку. Наприклад, компонента програмного продукту «ІС:Підприємство 8.3» під назвою «Робота с документами» дозволяє вести документообіг підприємства в електронній формі: завантажувати в програму документи підготовлені текстовими

процесорами або відскановані, присвоювати їм вхідну/вихідну нумерацію, підтримувати процеси узгодження і затвердження документів.

Наявність надлишкового функціонала ускладнює неспеціалісту роботу з програмою, перевантажений піктограмами інтерфейс збільшує час на прийняття рішень, знижує швидкість роботи та здатність користувача фокусуватися на своїх безпосередніх цілях та задачах, що збільшує витрати підприємства за рахунок збільшення часу на виконання операцій з обліку конструкторської документації. Інтерфейс повинен відповідати рівню користувача, тобто не мати надлишкових функцій, але містити найнеобхідніші.

#### 1.4 Типи інтеграції в інформаційних системах на рівні бізнес-транзакції.

Інтеграція на рівні бізнес-транзакцій передбачає реалізацію спільно використовуваних служб (сервісів). Служба може бути набором функцій, який використовується в декількох застосунках. Цей набір служб і буде бізнес-транзаціями. При використанні сервісно-орієнтованої архітектури бізнес-транзації можна розглядати як бізнес-сервіси, а при компонентному підході – як бізнес-об'єкти (бізнес-компоненти).

Інтеграція на рівні бізнес-транзації розрізняється залежно від рівня інтеграції. При внутрішній інтеграції взаємодіє велика кількість сервісів, а при зовнішній інтеграції – в основному два. Бізнес-процеси функціонують над виділеними службами, для управління якими існує спеціальна мова, що інтерпретується. Портали можна вважати графічними інтерфейсами бізнес-процесів, оскільки вони призначені для персоніфікованого доступу до інформації і консолідації даних з декількох джерел. Головним призначенням процесу інтеграції є об'єднання функцій додатків або модулів для надання

нової функціональності. При інтеграції додатків можна виділити два основні типи завдань: завдання інтеграції:

- корпоративних застосунків;
- додатків з різних ІС.

Для вирішення завдань першого типу застосовують системи ЕАІ, які іноді називаються А2А (Application – to – Application Integration), а для вирішення завдань другого типу застосовуються системи В2В (Business – to – Business Integration). У деяких випадках складно визначити різницю між інтеграцією А2А і В2В, оскільки складність деяких рішень усередині ІС може перевищувати складність рішень для їх спільного функціонування.

Існують три альтернативні топології інтеграції:

- точка–точка (Point – to – Point);
- шлюз (hub – and – spoke);
- шина (Bus).

У топології «точка–точка» усі об'єкти мають прямі зв'язки один з одним. Кожен зв'язок можна реалізувати будь–яким способом. Варіанти реалізації залежать від вимог і характеристик взаємодії між об'єктами. До недоліків топології можна віднести такі характеристики:

- зміни одного об'єкту впливають на об'єкти, що залишилися; логіка маршрутизації часто програмується в коді об'єктів; відсутність загальної моделі безпеки; використання різних АРІ;
- недостатня гнучкість; складність підтримки численних з'єднань «точка–точка»;
- низька надійність; складність створення фреймворків і підтримки асинхронної взаємодії.

Для скорочення числа використовуваних інтерфейсів слід використати топологію із загальним шлюзом або топологію із загальною шиною. Такі моделі інтеграції реалізуються на рівні проміжного ПЗ. Наступним кроком в розробці інтеграційної архітектури можна вважати появу корпоративної сервісної шини (Enterprise Service Bus – ESB).

Ряд авторів розглядає системи ESB, як наступний ступінь розвитку EAI. Проте є такі відмінності: EAI – централізована архітектура з обміном інформації через хаб (брокер), а ESB – шинна архітектура, яка може бути реалізована у вигляді декількох розподілених систем; вона орієнтована на використання відкритих стандартів. Ці дві відмінності демонструють можливість використання ESB як інтеграційної платформи, що дозволяє використати різні механізми.

ESB дозволяє проводити як внутрішню, так і зовнішню інтеграції, і є шиною, працюючою як слабо-зв'язна система, керована подіями. Концепції сервісно-орієнтованої архітектури (COA) і ESB сильно пов'язані. ESB підтримує принцип реалізації COA: розділення служби подання та її реалізації.

Функції ESB: надання інтерфейсів взаємодії; відправка і маршрутизація повідомлень; перетворення даних; реакція на події; управління політиками; віртуалізація. На підставі функцій ESB можна сформувавши типовий список вимог, які пред'являють користувачі: велика пропускну спроможність; підтримка декількох стилів інтеграції; забезпечення можливості додаткам працювати з сервісами як безпосередньо, так і через адаптери.

ESB є, по суті, логічним компонентом архітектури, що приводить інтеграційну інфраструктуру у відповідність принципу COA. Архітектурою, побудованою за принципом ESB, складніше управляти, але вона гнучкіша і масштабована (впровадження COA не потребує змін в усіх елементах системи, внаслідок чого зможе відбуватися поетапно). Можна подати ESB у вигляді п'ятирівневої структури:

- рівень сполучення (адаптери та інтерфейси);
- транспортна підсистема;
- рівень реалізації бізнес-логіки;
- рівень управління бізнес-процесами;
- рівень бізнес-управління.

Рівень сполучення покликаний вирішувати проблему використання різних інтерфейсів. На цьому рівні функціонують адаптери, які відстежують

події в додатках і в інтеграційній підсистемі, і забезпечують перетворення передаваних об'єктів при взаємодії з транспортною підсистемою.

Окрім заздалегідь створених адаптерів інтеграційної платформи існує можливість використати адаптери, створені самостійно. Адаптери можна розділити на дві категорії:

- технологічні (застосовуються для інтеграції технологічних компонент, за відсутності у них API);
- адаптери для додатків (застосовуються для інтеграції з конкретним застосунком).

Транспортна підсистема надає можливість асинхронної взаємодії інтегрованим застосункам. Цей рівень відповідає також за управління і безпеку інформації, може виконувати маршрутизацію повідомлень та їх обробку.

Рівень реалізації бізнес-логіки надає функції для трансформації і маршрутизації повідомлень. На цьому рівні функціонують брокери повідомлень, які обмінюються повідомленнями через транспортну підсистему. Брокер повідомлень може виконувати такі функції:

- прийняття повідомлень та їх відправка по вказаних адресах;
- перетворення форматів повідомлень;
- агрегація і фрагментація повідомлень;
- взаємодія з репозиторіями;
- Вибірка даних через виклики Web-служб;
- обробка помилок і подій;
- маршрутизація повідомлень за адресою, змістом, темою.

Управління бізнес-процесами на одноіменному рівні здійснюється за допомогою мови управління бізнес-процесами (Business Process Execution Language) на основі Web-сервісів. Рівень бізнес-управління є надбудовою над попереднім рівнем і призначений для управління бізнес-процесами в термінах відповідної предметної області. Підхід ESB має ряд переваг і дозволяє будувати інтеграційну архітектуру будь-якої складності.

### 1.5 Проблеми реалізації інтеграції бізнес-транзакцій

Визначення бізнес-транзакцій дає зробити важливий висновок: незавершена операція створення продукту практично завжди не має сенсу. Людська праця, робота безлічі контрагентів, витрати ресурсів і енергії виявляються марними, якщо кінцевий результат не був отриманий.

Компанія повинна орієнтуватися на багаторазове виконання бізнес-процесу, можливо, з різними вхідними ресурсами, але якої завжди створює передбачуваний результат. Тому друга небезпека незавершеності полягає в тому, що несподівана зупинка процесу може порушити можливості багаторазового виконання і тим самим зупинить бізнес. Уникнення цієї проблеми полягає у поверненні системи, всіх її учасників та ресурсів у деякий початковий стан, з якого вона може продовжити роботу та розпочати процес заново. Можливо, здаються безглуздими подібні міркування, коли звичний світ механізмів завжди після збою здатний автоматично повертатися в стан «готовність працювати», а програму на комп'ютері можна «вимкнути і включити». Проте, оснащення подібними можливостями процесів взаємодії незалежних учасників нерідко представляється складнішим завданням, на перший погляд, неактуальним, неможливим і непотрібним. Адже, дотримуючись підручників, бізнес-процес виконується за суворим алгоритмом, на вхід якого подаються ресурси, а результатом є продукт. Проектувальники процесів і системні аналітики, формальне мислення яких у кращому випадку вирощене інформаційними технологіями, упускають важливі властивості, що докорінно відрізняють алгоритм бізнес-процесу від алгоритму програмної інженерії:

— немає розвинених засобів перевірки запису алгоритму бізнес-процесів;

- бізнес-процес виконується незалежними учасниками, яких неможливо контролювати;
- бізнес-процес містить операції, що виконуються вручну, порядок виконання і результат яких може бути непередбачуваним;
- бізнес-процес породжує ефекти, які неможливо скасувати; бізнес-процес виконується так довго, що звичні методики програмування стають безглуздими;
- виконання бізнес-процесу коштує набагато дорожче виконання програми;
- алгоритм бізнес-процесу повинен бути зрозумілий всім учасникам.

Алгоритм кодується певною мовою програмування і виконується апаратурою. Переклад алгоритму на машинну мову супроводжується різними операціями перевірки, і отриманий двійковий код готовий до багаторазового виконання. Правильність запису алгоритму контролюється семантикою мови програмування та розвиненими засобами тестування та верифікації. А сучасна апаратūra, що виконує програму, настільки надійна, що ймовірність неправильного виконання правильного записаного алгоритму значно мала.

Алгоритм бізнес-процесу практично завжди записується в графічному вигляді. Він не містить змінних, функцій, точних виразів та інших інструментів, що зшивають його ділянки воедино. Засоби графічного моделювання практично не контролюють ніякі аспекти, крім, можливо, найпримітивніших, наприклад, наявності входів/виходів. Помилитись у графічному описі набагато простіше, ніж при записі алгоритму строгою мовою. Єдиним рішенням проблеми залишається те, що графічна модель часто згодом інтерпретується в програмний код і можливі помилки виправляються. Однак верифікація на детальному рівні не усуває проблем глобального, і помилка алгоритму виникне на стику кількох правильно працюючих програм.

Безсумнівно, існують засоби моделювання, вбудовані в програмні продукти виконання бізнес-процесів, такі як системи електронного

документообігу, CRM і ERP системи. У таких засобах здійснюється перевірка та верифікація графічних моделей. Однак вони орієнтовані на обмежену область контролю, і надійно описувати процеси взаємодії кількох незалежних підприємств (реально незалежних, а не є філіями або допоміжними організаціями) за допомогою них складно.

Алгоритм програми можна розділити на кілька частин. Кожна частина зазвичай виконується окремою підпрограмою, набагато рідше програмою на окремому комп'ютері. У будь-якому випадку, поведінка (під)-програми достеменно відомо і передбачувано. Перелік можливих проблем відомий програмісту, і він завжди впевнений у тому, що (під) програма повертатиме зрозумілий результат. Центральна програма керує усіма своїми сателітами, може впливати на їх роботу і визначає операції, що ними виконуються.

У бізнес-процесі бере участь кілька партнерів. Вони бувають настільки незалежні, що межі їх відкритості лежать у межах комерційної таємниці, а в інших площинах вони можуть бути навіть конкурентами. Не може бути й мови про допуск одного партнера до ресурсів іншого. Робота здійснюється строго в межах обумовлених контрактів, виконання яких контролюється кожним окремо. Те, як реалізовувати договір, теж є прерогативою партнера, особливості його роботи приховані від інших.

Алгоритм програми може припускати введення інформації користувачем. Він може змінювати поведінку програми в залежності від введених даних користувачем. Але ніколи алгоритм програми не містить алгоритм дій користувача, які аж ніяк не контролюються з боку обчислювальної техніки.

Алгоритм бізнес-процесу, навпаки, може містити вказівку для виконання набору ручних операцій або сам перераховувати ці операції. При проектуванні іноді перехід між операціями, що виконуються вручну, і автоматизованими операціями взагалі не можна чітко простежити на розглянутому рівні абстракції процесу. Ручні операції можуть мати непередбачуваний результат, залежать від кваліфікації оператора і, взагалі,

людського фактора. Неправильна інтерпретація службових інструкцій, забудькуватість або розсіяність можуть спровокувати помилки, які залишаються невиявленими і важко відслідковуються через свою фізичну або ручну природу.

Алгоритм програми найчастіше працює з об'єктами інформаційної природи. Він отримує дані на вхід та створює нові дані. Результати роботи такої програми легко відмінити – можна видалити нові дані, знищити створений об'єкт, очистити екран. У крайньому випадку, можна повернути механізм у початкове положення.

Алгоритм бізнес-процесу сприймає і інформацію, і матеріальні об'єкти однаково – як ресурси чи результат роботи. На відміну від інформації, ресурси мають властивість виснажуватися і незворотно однобічно перетворюватися. Тому ціна помилки бізнес-процесу набагато вища, ніж ціна помилки рядового алгоритму. Витрачені і поглинені ресурси неможливо повернути в початковий стан, створену суміш не розділити за складовими або не забрати розсіану в тепло енергію. Мало того, проміжні результати неуспішного процесу неможливо просто видалити. Швидше за все, необхідно запровадити спеціальні операції, які усувають наслідки невдачі, компенсують результати роботи, витрачаючи чергові ресурси. І, можливо, система буде повернена не в початковий стан, а в «схожий». Буває також, що розробка таких «зворотніх» операцій складніша і більш трудомістка, ніж «прямих», тому цю проблему вважають за краще «не помічати» і віддавати рішення до рук адміністраторів системи.

Алгоритми програм зазвичай пристосовані до швидкого виконання або низки виконання між реакціями користувачів. В очікуванні події з боку користувача програма може працювати нескінченно довго, здебільшого, не впливаючи на продуктивність решти системи. Будь-яка подія провокує каскад реакцій, в результаті якого програма або завершує роботу, або чекає наступної події. Усі операції, вироблені комп'ютером, програмуються як швидкі і точні. Тому можна блокувати ресурси, які в даний момент використовуються,

отримувати більш високий пріоритет роботи на короткий час, передавати дані іншій, що паралельно очікує програмі, і використовувати інші механізми, що добре працюють у невеликий відрізок часу.

Алгоритм бізнес-процесу може виконуватися кілька тижнів. Безперечно, його автоматизовані частини також здатні до практично миттєвого виконання. Однак решта складається з ручних операцій, тривалого очікування завершення інших процесів, використання ресурсів різної природи та якості. Використовувані ресурси не можна заблокувати, паралельно запущена програма не зможе перебувати в стані очікування даних нескінченно довго (її доведеться перезапустити), а інші програми багаторазово перезапускаються, що, наприклад, робить незручним використання примітивів синхронізації операційної системи. Проблеми тривалого виконання, пов'язані з програмним забезпеченням, зазвичай вирішуються за допомогою різних систем передачі повідомлень, використання подійної або сервісної архітектури, але проблеми множинного доступу до ресурсів таким чином вирішити неможливо.

Зазвичай запуск алгоритму програми не вимагає великої кількості ресурсів. Небагато підвищується завантаження процесора, систем пам'яті. Можливо, буде залучено користувача або запущено якийсь механізм. Але в самому загальному варіанті одноразовий запуск типової програми дещо збільшить витрату електроенергії.

Запуск бізнес-процесу практично завжди залучає кількох виконавців і безліч ресурсів. Навіть якщо ресурсами і результатом буде інформація, то часто вона створюється кількома користувачами, час роботи яких може коштувати дорого, не кажучи вже про матеріальні ресурси, що виснажуються, і ризику процесу бути незавершеним. Все це призводить до того, що одноразове виконання процесу набагато дорожче запуску програми. А разом з тривалим виконанням все це суттєво обмежує можливості тестування (а особливо тестування «зворотного», відновного процесу) алгоритму, що негативно впливає на його працездатність.

## 1.6 Задачі проектування та розробки бізнес-транзакцій

Завдання проектування та розробки бізнес-операції можна вирішити за допомогою методів системного аналізу, які передбачають поділ процесу рішення на окремі етапи. Сформулюємо власний перелік етапів, які найбільше підходять для розглянутої предметної області:

- визначення та аналіз проблеми взаємодії в розподіленому середовищі, що вимагає введення предмета господарських операцій. Оцінка актуальності та розв'язності проблеми. Аналіз існуючої системи та/або взаємодії;

- формування дослідницького завдання. Визначення господарської операції як системи. Визначення положення спостерігача, об'єктів, елементів та зв'язків між ними. Обмеження ділової операції з навколишнього середовища;

- визначення функцій учасників, елементів та бізнес-процесів. Визначення рівнів ієрархії елементів, підсистем, бізнес-процесів;

- визначення цілей учасників, елементів та бізнес-процесів. Визначення початкового та кінцевого станів. Оцінка взаємозалежності цілей, відсікання зайвих і суперечливих. Визначення спільної мети та початкового стану через множину кінцевих та початкових станів;

- визначення можливостей розробки системи. Оцінка наявних ресурсів та інструментів та аналіз необхідності розробки нових. Документування системних вимог;

- розробка альтернатив для структури та складу системи. Розробка альтернативних моделей системи. Оцінка та порівняння варіантів. Поєднання обраних варіантів і формування проекту остаточного рішення. Аналіз його параметрів. Удосконалення та корекція;

— реалізація рішення, окремих елементів, підсистем, функцій і процесів. Здійснення процесів обміну інформацією та управління. Оцінка результатів впровадження системи. Дослідна експлуатація. Робоча експлуатація. Технічне обслуговування та модернізація системи.

Таким чином, ми показали, що бізнес-транзакція може повністю володіти характеристиками системи, а довгострокові бізнес-транзакції або перетин кордонів підприємства є складними системами. З огляду на це, системний підхід до побудови складних бізнес-транзакцій, їх проектування та реалізації передбачає виконання ряду послідовних етапів, де власне програмування та впровадження розподіленої системи є лише невеликою частиною того, що необхідно зробити розробнику – створити надійне, ефективне середовище, яке забезпечує виконання господарських операцій.

## 2 ТЕОРЕТИЧНЕ ВИРІШЕННЯ ЗАДАЧІ «ІНТЕГРАЦІЇ ІТ-СЕРВІСІВ В БІЗНЕС-ТРАНЗАКЦІЇ»

2.1 Реалізація побудови бізнес-транзакції за допомогою шаблону «САГА»

2.1.1 Побудова сценарію задачі «Обліку конструкторської документації»

Візьмемо за приклад сервіс створення конструкторської документації на підприємстві. Розглянемо сценарій дій який застосовано для формування чіткого виду документу маючого свої чіткі характеристики та вимоги для його формування. Ініціаторами виконання даного процесу може бути клієнт додатків, періодична задача тощо.

Якщо програма працює з СУБД, яка підтримує механізм транзакцій, а запит клієнта додатка повністю вміщається в одну транзакцію і при цьому складається тільки з операцій із СУБД, то ви можете покластися на властивість ACID об'єкта угода. Інакше кажучи, СУБД гарантує атомарність, несуперечність, довговічність даних та ізоляцію транзакцій, у разі ізоляцію запитів від клієнтів докладання. Проте неможливо покладатися на СУБД у наданні запитів ACID, як у обробці запиту клієнта задіяно кілька транзакцій, виконуються такі операції, як читання, запис файлів, виклики сторонніх веб-служб тощо. буд. Необхідний інший механізм реалізації ACID.

Розглянемо ситуацію на такому прикладі: створимо бізнес-транзакцію облік конструкторської документації підприємства. У реальних системах цей процес досить складний, вимагає строгої коректності даних, обліку користувачів, документів, креслень, технічних специфікацій, тощо. Також враховуючи можливість формування змін багатьма авторами завжди необхідна перевірка авторизації [30].

Спрощений варіант виглядатиме так:

— спочатку відбувається процес створення документу з необхідними в ньому складовими (Item) в реєстрі (Stock). Якщо новий документ є коректний, то при виборі документу він переходить в режим редагування, інакше транзакція переривається;

— після цього виконується внесення змін до документу та внесення в нього необхідних даних;

— потім відбувається перевірка коректності внесених даних їх повноту, а також дотримання визначених стандартів;

— наступним етапом є збереження внесених змін та формування необхідного документу в результаті чого ми отримуємо готовий документ.

До цієї бізнес-транзакції пред'являються вимоги:

— бізнес-транзакція має бути виконана або повністю або не виконана зовсім, часткове виконання не допустиме;

— у будь-який момент часу є можливість робити звіти – журнали обліку розробленої документації. Ми не повинні бачити їхні проміжні стани [31], [32].

На основі отриманих даних та вимог можемо побудувати діаграму роботи сценарію (рисунок 2.1)

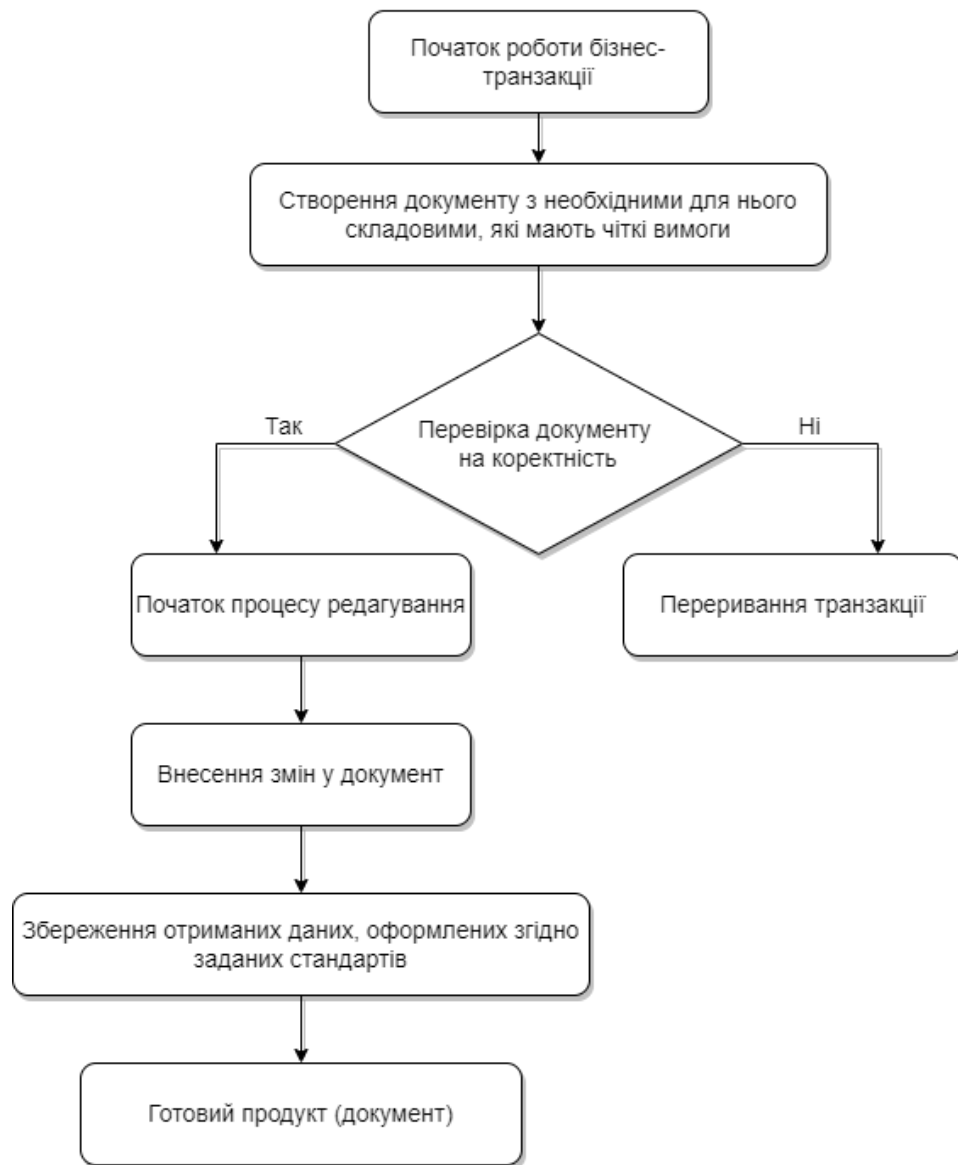


Рисунок 2.1 – Графічне зображення діаграми роботи заданої бізнес-транзакції

У першій операції ми створили новий документ і одночасно задали для нього основні характеристики (вид документу, складові, ДСТУ, тощо). Якщо результат запиту виконано коректно, то всі необхідні характеристики було обрано та задано. Виконав цей запит DML (Data Manipulation Language), ми блокували строчку в базі даних (розглядаємо СУБД Oracle). З однієї сторони ми вибрали можливість подвійних трат при конкурентних обробках замовлень, а з іншим рядом проблем, описаних нижче.

Проблема 1. Блокування запису. Поки не було зроблено коміт або відкат транзакції, рядок у таблиці складу блокується, що викликає очікування зняття блокування іншими транзакціями, що модифікують цей стовбець. Це може бути транзакції, які з погляду бізнес-логіки не заважають створенню документу [32].

Проблема 2. Довга відкрита транзакція до бази даних (Long lived transaction – LLT). Складно точно розрахувати, скільки буде виконуватися виклик стороннього веб-сервісу з урахуванням механізму повторної спроби (бажано зробити повторну спробу з експоненційним відстрочкою). Якщо сервіс тимчасово не доступний, повторна спроба може продовжитися хвилинами чи навіть годинами. Довга відкрита транзакція впродовж усього часу утримує наступні ресурси: сесія в СУБД, підключення з пулу

Проблема 3. Відкат транзакції у базі даних. Третя операція – внесення змін у документ, може закінчитися винятковою ситуацією – може спрацювати тригер у базі даних (тригер виконується в тій же транзакції). В результаті буде почато редагування документу, але час початку внесення змін не оновиться.

Проблема 4. Виключна ситуація при відправці нотифікацій. Якщо виникла така ситуація, то транзакція в базі даних буде відмінена, характеристики документу та його складові не буде заповнено, але пустий документ буде створено.

Проблема 5. Припинення роботи процесу між операціями. Процес може перерватися після або під час отримання відповіді від веб-сервісу. Це те, що стосується і проблеми з правильним завершенням роботи додатків, тому що ми не знаємо на якій операції зараз знаходиться процес. Крім того, відкат транзакцій в СУБД може виникнути через причини скінчення тайм-аута виклику стороннього веб-сервісу. В результаті це може привести до непостійного стану [13].

Один із варіантів вирішення цих проблем – застосування шаблону «САГА» [32].

### 2.1.2 Використання концепції шаблону «САГА»

Шаблон сага запропонував Гектор Гарсія-Моліна в 1987 році в якості механізму вирішення проблеми довгоживучих транзакцій у базі даних шляхом розбиття однієї великої транзакції в серії короткоживущих, що

$$\text{Saga} = T1 \Rightarrow T2 \Rightarrow T3 \Rightarrow \dots \Rightarrow Ti \quad (2.1)$$

Розглянутим виконання вимог, які видвигаються до саги – ACID:

A — Atomicity (атомарність). Сага або виконується повністю, або повністю відкатується, частково застосування транзакцій недопустимо. Для забезпечення цих умов у кожній транзакції саги повинні бути компенсуючі операції. Є два види забезпечення атомарності саги.

Відновлення назад. При виникненні виключної ситуації в одній із транзакцій саги, виконання саги призупиняється і виконується послідовність компенсуючих транзакцій у зворотному порядку, такому що

$$T1 \Rightarrow T2 \Rightarrow T3 [\text{Exception}] \Rightarrow (C3) \Rightarrow C2 \Rightarrow C1 \quad (2.2)$$

Перша виконується компенсуюча транзакція є компенсаційною для тієї, яка викликала вийняток. На практиці це не завжди просто реалізувати, а іноді це і не потрібно — достатньо викликати компенсаційну транзакцію для попередніх транзакцій, наприклад, якщо транзакція виконує транзакцію в базі даних. При виникненні виключної ситуації база даних справляє відкат усіх змін [31].

Відновлення вперед. Для цього нам потрібно ввести поняття Save Point – точки, які дозволяють повторити спробу виконання саги, що дуже схоже на повторну спробу, таку що

$$T1 \Rightarrow \text{Збереження точки 1} \Rightarrow T2 \Rightarrow \text{Збереження точки 2} \Rightarrow T3 [\text{Виняток}] \Rightarrow \text{Збереження точки 2} \Rightarrow T3 \quad (2.3)$$

Можна компенсувати, об'єднати ці два способи забезпечення атомарності, що можна позначити як

$$\begin{aligned}
 T1 \Rightarrow \text{Точка збереження} \Rightarrow T2 \Rightarrow T3 \text{ [Виняток]} \Rightarrow \\
 C3 \Rightarrow C2 \Rightarrow T2 \Rightarrow T3 \Rightarrow \text{Кінець}
 \end{aligned}
 \tag{2.4}$$

Для того, щоб було можливо виконати одну і ту ж транзакцію кілька разів, важливо ввести поняття ключа ідемпотентності вхідного параметра. У такому випадку транзакція ігнорує застосування її операцій, якщо раніше транзакція була успішно виконана цим ключем [16].

Окремий момент забезпечення атомарності виконання саги є стійкістю саги. В ідеальному варіанті необхідно зберегти всі Save Point, але це може сильно збільшити продуктивність. Варіант – зберігати на диску тільки перший Save Point до початку першої транзакції. Це дозволяє в разі аварійної перезавантаження додатків виконати сагу заново. Ключі ідемпотентності гарантують єдине застосування операцій транзакцій.

C – Consistency (консистентність). ACID – це набір інваріантів, які приймає система після завершення саги. Ця вимога поширюється до транзакцій саги. Власне, сумарний ефект від виконання всіх транзакцій має бути одним із інваріантів станів системи. Ця ж вимога стосується і компенсаційних транзакцій.

I – Isolation. (Ізольованість). Паралельно виконувані саги не повинні впливати один на одного. Найчастіше його можна послабити.

D – Durability (стійкість). У разі успішного виконання саги її результат уже не може бути скасовано – ця вимога також ескалується до стійкості транзакцій саги.

Розділимо вищеописану бізнес-транзакцію створення документації на транзакції:

Перша транзакція саги це транзакція в базі даних, що виконує запит DML:

```

UPDATE stock SET quantity=quantity-#{proces.quantity}, reserved =
reserved+#{proces.quantity} WHERE item_id = #{proces.itemId} AND quantity >=
#{proces.quantity};

```

Якщо транзакція створила більше нуля записів, продовжуємо виконання саги. Компенсація - транзакція в базі даних:

```
UPDATE stock SET quantity=quantity+${proces.quantity}, reserved = reserved-${proces.quantity} WHERE item_id = ${proces.itemId}
```

Друга транзакція – виклик стороннього веб-сервісу редагування документу:

```
var trackingNumber= creationWebService.post(proces.itemId,proces.service)
```

Компенсація – ми маємо викликати перевірку внесених характеристик

```
undoDeliveryWebService.post(trackingNumber)
```

Третя транзакція – надсилання нотифікації користувачеві. Компенсація – надсилання нотифікації про скасування створення документу.

Четверта транзакція саги:

```
INSERT INTO creation_journal (proces_id, tracking_number, item_id, quantity) VALUES (${proces.id}, ‘$trackingNumber’, ${proces.item_id}, ${proces.quantity});
```

```
UPDATE stock SET reserved = reserved — ${proces.quantity} WHERE item_id = ${proces.item_id};
```

Компенсація:

```
DELETE FROM creation_journal WHERE trackingNumber = $'trackingNumber';
```

```
UPDATE stock SET reserved = reserved + ${proces.quantity} WHERE item_id = ${proces.item_id};
```

Розглянемо як сага вирішила озвучені раніше проблеми:

Проблема 1. Блокування запису. Тривале блокування стовбця у таблиці документів не відбувається. Інші транзакції можуть паралельно створювати документи при необхідності. Якщо бути точним, то блокування все-таки є між виконанням CREATE запиту і коммітом транзакції в базі даних, але це дуже короткий період [17].

Проблема 2. Довга відкрита транзакція у базі даних. Відсутня.

Проблема 3. Відкат транзакції у базі даних. Відкат останньої транзакції можливий, однак у цьому випадку виконується компенсація третьої, другої транзакції та першої транзакції саги та система система прийде у консистентний стан.

Проблема 4. Виняткова ситуація під час надсилання нотифікації. У цьому випадку виконується ланцюжок компенсуючих транзакцій - скасування створення документу. Системи входить у консистентний стан.

Проблема 5. Припинення роботи процесу між операціями. І тут буде можливість повторити виконання саги від початку. Завдяки ідемпотентності раніше виконані транзакції будуть проігноровані [18].

Розділивши одну бізнес-транзакцію на серію транзакцій, ми визначили компенсуючі дії для кожної транзакції окремо. Компенсація кожної окремої транзакції залежить від інших транзакцій. Ми можемо додавати або прибирати транзакції в сазі не мене їх логіки інших транзакцій, можемо тестувати транзакцію та компенсацію окремо від іншої бізнес-логіки.

2.2 Розробка бізнес-транзакції «Обліку конструкторської документації» за допомогою офарбованих мереж Петрі.

Дослідження бізнес-транзакцій можливе з погляду різних аспектів, для чого можуть застосовуватися різні засоби моделювання. До таких аспектів вивчення, як уже було показано, належать послуги, сервіс-орієнтована архітектура, транзакційне управління, композиції сервісів, пошук сервісів у розподіленій середовищі, різні технології інтеграції, системи паралельних процесів та, нарешті, бізнес-процеси [16].

Для побудови апарату моделювання як базовий інструмент скористаємося теорією пофарбованих сіток Петрі. Для цього дамо короткий

опис теорії комплектів, теорії простих мереж Петрі та теорії пофарбованих мереж Петрі.

Офарбованою мережою Петрі будемо називати таку мережу, що:

$$CPN = (P, T, A, \Sigma, V, C, G, E, I), \quad (2.5)$$

де:  $P$  – кінцева безліч позицій;

$T$  – кінцева безліч переходів таких, що  $P \cap T = \emptyset$ ;

$A$  – кінцева безліч спрямованих дуг, таких, що  $P \cap T = P \cap A = T \cap A = \emptyset$ ;

$\Sigma$  – кінцеве безліч непустих типів, також званих множинами кольорів. Безліч типів визначає значення даних, функцій та операторів (вирази дуг, вартових та функцій ініціалізації);

$V$  – кінцева безліч типізованих змінних таких, що  $\forall v \in V: Type[v] \in \Sigma$   
Під позначенням  $Type$  розуміється функція, що повертає тип змінної;

$C: P \rightarrow \Sigma$  функція «офарбовування», що ставить у відповідність кожній позиції підмножина кольорів, тобто інтуїтивно це означає, що кожна фішка позиції  $p$  повинна мати типи даних;

$G$  – сторожова функція, що ставить у відповідність кожному переходу сторожову умову («вартова»), така, що:  $Type[G(t)] = Bool$ , тобто функція повертає булеве значення а всі змінні вирази мають тип, що належить  $\Sigma$  [17];

$E$  – функція вираження над дугою, що ставить у відповідність кожній дузі  $a \in A$ , вираз такий, що:

$$\forall a \in A: Type[E(a)] = C(p)ms, \quad (2.6)$$

де  $p$  – це позиція з'єднана з дугою  $a$ .

Це означає, що кожен вираз над дугою має повертати комплект над типом суміжної позиції.

—  $I$  – це функція ініціалізації, що ставить у відповідність кожній позиції  $p \in E$  вираз ініціалізації такий  $\forall a \in P: Type[I(p)] = C(p)ms$ ,  
офарбована мережа Петрі безпосередньо дозволяє виділення паралельних дуг,

замінюючи відображуваний ними функціонал комплектами змінних і операціями над дугами, тобто. граф пофарбованої мережі Петрі є простий граф.

Переваги використання пофарбованих мереж Петрі для моделювання динамічних систем були описані Куртом Йенсенем в його тематичних роботах.

- офарбовані мережі Петрі мають добре розвинену семантику, яка однозначно описує поведінку будь-якого екземпляра мережі, дозволяє розробляти симулятори і використовувати формальні методи аналізу;

- офарбовані мережі Петрі, на відміну інших розширень, дуже узагальнені і тому можуть використовувати різні типів систем;

- офарбовані мережі Петрі можуть розширюватися аналізом тимчасових міток та ієрархічною структурою, що дозволяє їм описувати складні динамічні системи;

- офарбовані мережі Петрі об'єднують опис управління та синхронізації системи разом з описом маніпуляцій з даними, що дозволяє моделювати систему невідривно від використовуваних нею даних;

- на відміну від інших технологій моделювання мережевих структур, моделі на основі пофарбованих мереж Петра не зазнають серйозних змін при модифікації системи, що моделюється;

- добре розроблений і доведений математичний апарат забарвлених мереж Петрі дозволяє використовувати широкий набір формальних методів аналізу моделі.

Для моделювання складних структур було запропоновано розширити забарвлені мережі Петрі ієрархією модулів. Моделювання процесів за допомогою мереж Петрі описано в роботах Віллаван дер Аалста (англ. Will.MP van der Aalst), де він описує бізнес-процеси, їх структуру та ієрархію і вводить поняття WF-мереж (Workflow мережа) для моделювання таких процесів.

WF-мережа, це мережа Петрі  $PN = (P, T, F)$ , така, що:

— PN має дві спеціальні позиції: вхідну  $i: \bullet i = \emptyset$  і вихідну:  $o \bullet = \emptyset$ . Ця властивість дозволяє моделювати виконання процесу у WFMS-системі та визначає запуск та закінчення процесу шляхом приміщення в його початкову позицію фішки та досягнення фішки вихідної позиції;

— якщо додати в PN перехід  $t^*$ , який з'єднає  $i$  і  $o$ , то результуюча мережа Петрі буде строго пов'язаним графом. Ця умова вважає, що будь-який перехід лежить на шляху від  $i$  к  $o$ .

Існує реалізація технології моделювання сервісної композиції на основі т.зв. "сервісних" або "відкритих" мереж. Сервісна мережа  $N = (P, Pin, Pout, T, F, \Omega)$  містить:

— мережу Петрі  $PN = (P, T, F)$ ;

— дві непересічних множини  $Pin \subseteq P$ , вхідних позицій і  $Pout \subseteq P$  вхідних позицій таких, що:  $\bullet Pin = Pout \bullet = \emptyset$   $Pin$  і  $Pout$  становлять інтерфейс мережі  $N$  і початкова, ні фінальні маркування не зачіпають інтерфейсні позиції;

— безліч  $\Omega$  різних маркувань, званих фінальними маркуваннями.

Сервісні мережі призначені для моделювання мережевих повідомлень між службами, асинхронної взаємодії та композиції сервісів. При цьому вони розглядають служби в цілому, не дозволяють виділяти їх окремий бізнес-функціонал і не мають можливостей підтримки різних типів фішок, як пофарбовані мережі, тому не можуть використовуватися для моделювання сервісів у середовищі бізнес-транзакцій.

### 2.2.1 Моделювання бізнес-транзакції за допомогою мережі Петрі

Мережі Петрі використовуються для моделювання сервісних композицій і бізнес-процесів. Ідея бізнес-транзакцій з сервісної ідеології дозволяє уявити переходи за окремі сервіси або вкладені бізнес-процеси, тоді

як маркування позиції демонструють певні стани, досяжні бізнес-транзакцією в процесі виконання. Організаційної точки зору сервіс-набір операцій, згрупованих за якою-небудь ознакою.

З огляду на те, що як програмна одиниця може реалізовувати безліч різних операцій, доцільно розглядати як переходів мережі Петрі ці операції. У сервіс-орієнтованому програмуванні існує низка способів виділення та угруповання цих операцій. Наприклад, у стандарті WSDL операції об'єднуються в деякі набори «типи порту», які вже включаються в сервіс. Операція сервісу реалізує окреме завдання з чітко визначеними вхідними та вихідними параметрами. Отже, кожену операцію можна як окремого переходу і враховувати те, що у моделі, реалізованої з допомогою мережі Петрі, кілька переходів можуть належати одному й тому самому сервісу. Як уже зазначалося, контракт служби є універсальною технологією забезпечення слабкої пов'язаності, відокремлюючи її опис виконуваних нею функцій від конкретної реалізації [18].

Визначення 1. Нехай  $CPN = (P, T, A, \Sigma, V, C, G, E, I)$  - це офарбована мережа Петрі, що моделює взаємодію сервісів [19]. Тоді окрему операцію сервісами представимо як:

$$O = (t, P^o, A^o, \Sigma^o, V^o, C^o, G^o, E^o), \quad (2.7)$$

де:  $t \in T$  – окремий перехід мережі Петрі;

$P^o \subseteq P$  – безліч вхідних та вихідних позицій таке, що  $P^o = P_{in} \cup P_{out}$ ;

$P_{in} \subseteq P^o$  безліч вихідних позицій переходу  $t$ ;

$A^o \subseteq P_{in} \times t \cup t \times P_{out}$  – безліч дуг, що з'єднують перехід  $t$  с вхідними та вихідними позиціями. Під  $A_{in}$  і  $A_{out}$  будемо розуміти набір вхідних та вихідних дуг відповідно;

$\Sigma^o \subseteq \Sigma$  – це підмножина безлічі типів даних, що використовуються в фарбованій мережі Петрі;

$V^o \subseteq V$  – підмножина типізованих змінних, що використовуються операцією;

$C^o: P^o \rightarrow \Sigma^o$  – функція «офарбовування», що ставить у відповідність кожній позиції підмножину кольорів;

$G^o$  – сторожова функція, що ставить у відповідність операції сторожову умову, така, що:  $Type[G^o(t)] = Bool$ , тобто функція повертає булеве значення;

$E^o$  – функція вираження над дугою, що ставить у відповідність кожній дузі  $a \in A^o$  вираз такий, що  $\forall a \in A^o: Type[E^o(a)] = C(p)ms$ , де  $p$  – це позиція, поєднана з дугою  $a$ .

Єдина сторожова умова дозволяє здійснювати запуск операції у разі, якщо вхідні змінні будуть задовольняти умовам, що задаються стражами. Функція фарбування  $C: P^o \rightarrow \Sigma$  дозволяє задавати строго типізований вхід та вихід операції, що властиво властивостям контракту служби та завданням, які вона реалізує.

Операція  $O$  є обмеженою, якщо для всіх її вхідних позицій визначено деякий граничний комплект:

$$\forall p \in P_{in}^o, \forall M(p): |M(p)| \ll = m_p, \quad (2.8)$$

де:  $m_p$  – граничний комплект позиції  $p$ . Вимоги обмеженості операції потрібні, наприклад, якщо окрема операція представляється системою масового обслуговування з накопичувачем обмеженої ємності. Мережа є обмеженою, якщо всі її операції обмежені.

Сервіс, як і будь-яка операція, може викликатися тільки одним учасником. Якщо сервіс викликається кількома учасниками, відбувається запуск кількох служб одночасно. Таким чином, для мережі Петрі необхідно встановити обмеження, що передбачає виклик сервісу тільки одним учасником. Для цього необхідно задати умову, щоб усі вхідні позиції операції належали одному учаснику.

### 2.2.2 Конструкція сервісної взаємодії

Методика виділення зі служби набору операцій дозволяє реалізувати моделі сервісного взаємодії мережах Петрі без обмежень, що з складністю служби.

На відміну від представлених у західній літературі підходів у вигляді: «сервісних» або «відкритих» мереж Петера Массутхе, WF-мереж Віллаван дер Аалста і простих мереж Петрі, одержувана модель має наступні перевагами.

Пропонує розглядати сервіс і з точки зору програмної реалізації, тобто дозволяє описувати не сервіси цілком, а окремі сервісні операції, які, можливо, не пов'язані один з одним.

Виділяє ключові аспекти сервіс-орієнтованого програмування: поняття інтерфейсу, незалежної реалізації служби, сервісної композиції, яка може розглядатися як єдиний сервіс більш високого рівня абстракції і також включатися до композиції з іншими службами.

Описується на базі високорівневих мереж Петрі - забарвлених мереж, що дозволяє використовувати всі переваги цієї технології для моделювання динамічних систем. Отримувана модель, на відміну простих мереж Петрі, здатна описувати маніпуляції з різними типами даних.

Не має обмежень, що накладаються на процес виконання на початкові та фінальні позиції моделі, що дозволяє вважати взаємодію завершеним при досягненні фінального маркування, а не деякої фінальної позиції. Правильно і зворотне - ініціювати взаємодію дозволяє початкова маркування, що охоплює, можливо, безліч сервісів, а не деяка початкова позиція.

З іншого боку, модель дозволяє відображати не тільки виключно сервіс-орієнтовані системи, але, що важливо, описувати будь-які багатоетапні бізнес-процеси, які включають низку послідовних операцій.

Введемо поняття учасника. Нехай  $H$  – безліч учасників сервісної взаємодії. Безліч позицій та операцій належать цим учасникам. Приналежність будемо позначати як  $hP$  – де  $h \in H$ , що означає безліч операцій  $P$  належить учаснику  $h$  (вірніше операції входять до склад сервісів, які належать учаснику).

Нехай  $CPN = (P, T, A, \Sigma, V, C, G, E, I)$  – це пофарбована мережа Петрі, що моделює взаємодію сервісів  $H$  – множина учасників. Тоді:

$$\begin{aligned} P &= h_1P \cup h_2P \cup \dots \cup h_nP, \\ T &= h_1T \cup h_2T \cup \dots \cup h_nT, \end{aligned} \quad (2.9)$$

де:  $n$  – кількість учасників,  $P$  і  $T$  – множина переходів та операцій мережі відповідно.

Нехай  $O = (t, P^o, A^o, \Sigma^o, V^o, C^o, G^o, E^o)$  – це окрема операція сервісу мережі  $CPN$ ,  $H$  – безліч учасників. Щоб задовольняти природі викликаних програмних операцій, ця операція повинна бути обмежена наступною умовою:

$$P_{in}^o \subseteq_h P: h \in H, \quad (2.10)$$

Тобто всі вхідні позиції операції мають належати одному учаснику.

Зважаючи на те, що апарат пофарбованих мереж Петрі не дозволяє описувати обчислення над даними поза операціями над дугами, виконувати операцією функції можна лише у вигляді сукупності операцій над вхідними і вихідними дугами.

### 2.2.3 Переваги використання офарбованих мереж при інтеграції сервісів у бізнес-транзакції.

Методика виділення зі служби набору операцій дозволяє реалізувати моделі сервісного взаємодії мережах Петрі без обмежень, що з складністю служби [20].

На відміну від представлених у західній літературі підходів у вигляді: «сервісних» або «відкритих» мереж Петера Массутхе, WF-мереж Віллаван дер Аалста і простих мереж Петрі, одержувана модель має наступні перевагами:

— пропонує розглядати сервіс і з точки зору програмної реалізації, тобто дозволяє описувати не сервіси цілком, а окремі сервісні операції, які, можливо, не пов'язані один з одним;

— виділяє ключові аспекти сервіс-орієнтованого програмування: поняття інтерфейсу, незалежної реалізації служби, сервісної композиції, яка може розглядатися як єдиний сервіс більш високого рівня абстракції і також включатися до композиції з іншими службами [18];

— описується на базі високорівневих забарвлених мереж Петрі, що дозволяє використовувати всі переваги цієї технології для моделювання динамічних систем. Модель, на відміну від простих мереж Петрі, здатна описувати маніпуляції з різними типами даних;

— не має обмежень, що накладаються на процес виконання на початкові та фінальні позиції моделі, що дозволяє вважати взаємодію завершеним при досягненні фінального маркування, а не деякої фінальної позиції. Правильно і зворотне – ініціювати взаємодію дозволяє початкова маркування, що охоплює, можливо, безліч сервісів, а не деяка початкова позиція;

— з іншого боку, модель дозволяє відображати не тільки винятково сервіс-орієнтовані системи, але, що важливо, описувати будь-які багатоетапні бізнес-процеси, які включають ряд послідовних операцій [21].

### **3 ВИКОРИСТАННЯ ТЕОРЕТИЧНИХ РЕЗУЛЬТАТІВ З ВИРІШЕННЯ ЗАДАЧІ ІНТЕГРАЦІЇ ІТ-СЕРВІСІВ В БІЗНЕС-ТРАНЗАКЦІЇ**

#### **3.1 Постановка задачі реалізації інтеграції ІТ-сервісу у бізнес-транзакцію**

Розглянувши декілька методів побудови бізнес-транзакцій та існуючу концепцію інтеграції ІТ-сервісів інформаційних систем у бізнес-транзакцію, ми можемо точно сказати, що існуючі методи інтеграції та побудов інформаційних систем є ситуативними та мають бути спроектовані з урахуванням чітко поставленої задачі бізнес-транзакції та вимог до сервісу, щоб уникнути помилок даних, зайвих витрат, тощо.

Постановка задачі розробка сервісу обліку конструкторської документації ІС машинобудівного підприємства полягає в автоматизації рутинних і трудомістких процесів, пов'язаних з процесами обліку, зберігання та обігу КД на машинобудівному підприємстві. Інформація та документація, що обліковується, обробляється та відображається, доступна працівникам конструкторського бюро.

Як бізнес-транзакцію будемо розглядати сценарій задачі обліку конструкторської документації ІС машинобудівельного підприємства до вимог якої входить автоматизацію процесу обліку КД; скорочення часу на виконання операцій формування і обробки документації; зниження трудомісткості та збільшення швидкості обробки КД; підвищення якості перевірки документації; забезпечення надійного зберігання даних; швидкого доступ до них; ефективний пошук, перегляд та швидке внесення змін до документації; запобігання дублювання даних; забезпечення малої ймовірність введення помилкових даних, низького відсотку дублювання даних, захисту інформації.

Для того, щоб мета задачі розробки та інтеграції ІТ-сервісу у бізнес-транзакцію була досягнена, бізнес-транзакція повинна відповідати наступним критеріям:

- повинна забезпечувати збільшення швидкості отримання вихідних документів;
- повинна забезпечувати виключення дублювання інформації за рахунок ведення єдиної БД;
- не повинна містити помилкових даних; повинна мати достатню швидкодію.

Обмеження на створення ІС виглядають наступним чином:

- вартісні – визначають, що витрати на створення системи не повинні бути більше ніж експлуатація існуючої системи. Вартість розробки повинна окупитися за 1-2 роки експлуатації; часові – розробка задачі обмежена у часі;
- функціональні – задача «Облік конструкторської документації» ІС машинобудівельного підприємства, що розробляється, повинна містити перелік усіх заявлених до реалізації замовником функцій.

3.2 Опис сценарію бізнес-транзакції обліку конструкторської документації.

3.2.1 Опис об'єкту дослідження та постановка основних задач поставлених до бізнес-транзакції.

Об'єктом дослідження є НКМЗ – Новокраматорський машинобудівний завод – завод важкого машинобудування в Україні. Підприємство

зорієнтовано на виготовлення унікальних машин і важкого промислового обладнання для гірничо-металургійного комплексу.

Місією підприємства є задоволення потреб ринку як провідного виробника унікальної техніки шляхом ефективного використання інтелектуальних і технічних можливостей підприємства [3].

Згідно з класифікацією організацій НКМЗ є:

- за формою власності – приватним;
- за принципом побудови і функціонування – формальною організацією;
- за належністю до організаційно-правової форми – приватним акціонерним товариством;
- в залежності від виду діяльності – виробничим підприємством;
- за належністю до галузі виробництва – промисловим;
- за масштабами діяльності – національним;

Діяльність НКМЗ полягає у виконанні наступних основних бізнес-процесів:

- розробка і впровадження у виробництво найбільш прогресивних видів устаткування і технологічного оснащення, засобів автоматизації та механізації;
- розробка ескізних, технічних і робочих проектів виробів, що відповідають технічним завданням, стандартам, нормам техніки безпеки, з використанням стандартизованих деталей і складальних одиниць;
- забезпечення технічною конструкторською документацією (ДК) виробів.

Для організації функціонування основних процесів треба налагодити ряд другорядних процесів, таких як облік та контроль креслень та специфікацій деталей, вузлів, агрегатів, виробів на технологічність; розробка розрахункових матеріалів для обґрунтування потреби підприємства, в обладнанні; систематичне вивчення причин браку і дефектів у виробництві.

Структуру НКМЗ можна спрощено представити наступним чином: управління підприємством – це функція дирекції та організаційно-керуючих відділів, до яких відносяться: генеральний директор заводу, його заступники з виробництва, організаційних та комерційних питань та головний інженер виробництва.

Машинобудівний завод складається з окремих виробничих одиниць – цехів, підрозділів та відділів проектування обладнання. Склад цехів заводу визначається обсягом випуску продукції, характером технологічних процесів, вимог до якості виробів та іншими виробничими факторами.

Склад машинобудівного заводу можна розділити на групи:

- конструкторські відділи проектування обладнання;
- обробні цехи (механічні, термічні, холодного штампування);
- допоміжні цехи (інструментальні, ремонтно-механічні та ін.);
- склади для інструменту, готових деталей, палива, моделей;
- енергетичні підрозділи (електростанція, теплоелектроцентрально);
- санітарно-технічні підрозділи (опалення, водопостачання, вентиляція, технологічна лабораторія, медичний пункт, їдальня).

Серед конструкторських відділів на ЗАТ НКМЗ можна виділити:

- відділ металургійного обладнання – виконує роботи по проектуванню систем розподілу металу (жолоби), фільтрувального обладнання, печей та міксерів різних видів (індукційні, електричні, газові), типового і нестандартного обладнання, будь-яких металоконструкцій для технічних пристроїв, будівель і споруд;
- відділ прокатного обладнання – виконує роботи по проектуванню товстолистові і широкосмугові прокатних станів та обладнання для них, рольгангів, ножиць;
- відділ проектування редукторів – виконує роботи по проектуванню станових, транспортних, спеціальних редукторів;

— відділ електроприводу – виконує роботи по проектуванню гідравлічного, мастильного та пневматичного обладнання, електрообладнання машин і механізмів на базі електроприводів.

Слід зазначити, що організаційна структура управління на підприємстві побудована за лінійно-функціональному типу. Головна перевага сучасної організаційної структури ЗАТ НКМЗ – збереження цілісності єдиного виробничого комплексу з певною самостійністю його структурних підрозділів. Кожне виробництво самостійно відповідає за свій розвиток і забезпечення замовленнями цехів, випуск конкурентоспроможної продукції, дотримання термінів поставки продукції відповідно до укладених договорів, за собівартість, обсяги грошових коштів, що надходять від споживачів продукції [4].

Для формування вимог сценарію приведемо схему організаційної структури розглядаємого підприємства на рисунку 3.1.

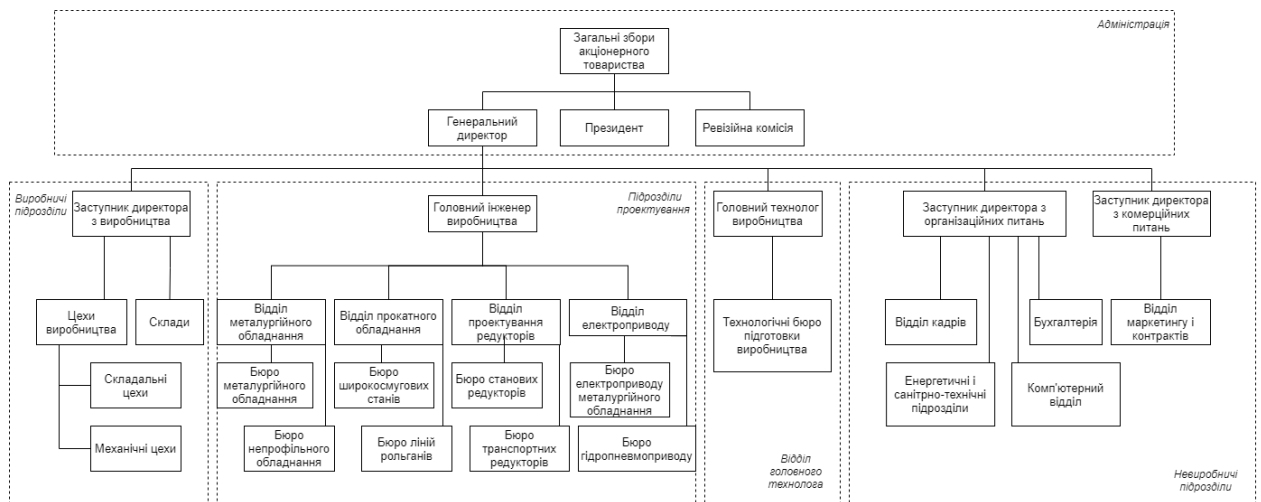


Рисунок 3.1 – Схема організаційної структури Новокраматорського машинобудівного заводу

Ціль діяльності конструкторського бюро – розробка якісної КД у вигляді складальних креслень, специфікацій та окремих креслень, що в подальшому надходять в складальні цехи. Кожне бюро спеціалізується на проектуванні

різного виду обладнання: бюро металургійного обладнання, бюро непрофільного обладнання, бюро широкосмугових станів бюро ліній рольгангів, бюро станових редукторів, бюро транспортних редукторів, бюро електроприводу металургійного обладнання, бюро гідропневмоприводу.

Фінансові цілі бюро спрямовані на зменшення витрат підприємства за рахунок скорочення часу на виконання операцій, пов'язаних з КД. Отримання доходу для подальшої підтримки і розвитку підприємства.

Опанування нової технології електронного документообігу у вигляді бази даних (БД) конструкторської документації з можливістю автоматичного створення звітів про виконану інженерами-конструкторами роботу є інноваційною ціллю діяльності конструкторського бюро.

Залежно від функціонального призначення складальної одиниці її розробкою займається певне бюро відділу. Для кожної складальної одиниці, що випускається підприємством, складається пояснювальна записка з принципами її роботи і технічними умовами. Пояснювальну записку і креслення не обов'язково розробляє один і той же співробітник.

Кожен конструкторський відділ спеціалізується на розробці конструкторської документації для певного виду обладнання. Підлягає виготовленню обладнання для зручності проектування розбивається на складальні одиниці (вузли) – виріб, що складається з декількох деталей, що з'єднуються в процесі його виготовлення між собою в одну загальну конструкцію.

Обов'язками інженер-конструктора є розробка ескізних, технічних і робочих проектів особливо складних, складних і середньої складності виробів, розробка відповідної конструкторської документації, з використанням засобів автоматизації проектування.

Працівники конструкторського бюро – інженера-конструктори в залежності від рівня освіти, кваліфікації та стажу роботи обіймають категорію I, II або III. Для інженера-конструктора I категорії потрібна повна вища професійна (технічна) освіта і стаж роботи на посаді інженера-конструктора II

категорії не менш 3 років. Інженер-конструктор II категорії зобов'язаний мати повну вищу освіту та стаж роботи на посаді інженера-конструктора III категорії або інших інженерно-технічних посадах не менше 3 років. Для інженера-конструктора III категорії потрібна вища професійна (технічна) освіта і досвід роботи зі спеціальності, придбаний у період навчання, чи стаж роботи на інженерно-технічних посадах без кваліфікаційної категорії.

Право на розробку креслень вузлів або деталей надається інженерам-конструкторам певної категорії в залежності від встановленої складності виробу (наприклад, I категорія може займатися розробкою креслення для деталі будь-якої складності). За одним інженером може бути закріплена розробка декількох креслень і над одним кресленням можуть працювати відразу кілька співробітників конструкторського бюро, здійснюючи контроль (наприклад, один розробив, інший перевірів, третій затвердив, четвертий вніс зміни).

В процесі розробки креслення може змінювати свій статус («Розробляється», «Перевіряється», «Стверджується», «Змінюється», тощо), після затвердження креслення начальником бюро, його статус приймає значення «Видано». За одним кресленням може бути виготовлено декілька деталей, що відрізняються номером та використовуються при отриманні підприємством замовлення на виготовлення комплекту однакових деталей для різних виробів, що є складовою частиною різних машин. У кожного креслення є свій тип: складальний, деталі, загального вигляду і т.д.

Креслення розробляються в середовищі Mathcad, призначеному для автоматизованого проектування креслень, мають формат (A1, A2, A3 і т.д.) і можуть мати роздруковану копію, яка зберігається в архіві. Кожне складальне креслення супроводжується розробкою специфікації, яка визначає склад складальної одиниці – має перелік її деталей, їх кількість, розміри, матеріали. Специфікацію складають на окремих аркушах на кожен складальну одиницю. ДСТУ 3321:2003 встановлює форму і порядок заповнення специфікацій конструкторських документів на виробу всіх галузей промисловості [22].

Розробка функціональної задачі «Облік конструкторської документації» ІС машинобудівельного підприємства необхідна для забезпечення ефективного функціонування підприємства важкого машинобудування, тому що вона спрямована на автоматизацію рутинних і трудомістких процесів, пов'язаних з його діяльністю.

Основними роботами задачі «Облік конструкторської документації» ІС машинобудівельного підприємства є:

- внесення даних про креслення та специфікації;
- облік креслень;
- розробка специфікації виробу;
- формування сповіщення про внесення змін до документу;
- присвоєння архівного номеру сформованому документу;
- формування журналу обліку КД.

Схема потоків робіт (рисунок 1.3) задачі «Облік конструкторської документації» ІС машинобудівельного підприємства містить шість робіт. Перша робота відповідає за внесення даних про креслення та специфікацію, необхідних для розробки виробу.

Друга робота відповідає за облік креслень виробів на основі списку видів та типів креслень виробів, які вже зберігаються до БД ІС підприємства. Нові креслення згідно з технічного завдання (ТЗ) на їх розробку так само додаються інженером-конструктором до вже існуючих у БД.

Третя робота відповідає за розробку специфікації виробу, що містить інформацію про склад та кількість деталей, а також матеріалів, які входять до складу виробу.

Четверта робота відповідає за формування сповіщення про внесення змін до документа. Одночасно виконати другу, третю та четверту роботи неможливо. Результати другої, третьої чи четвертої роботи є вхідною інформацією для п'ятої роботи. Вона відповідає за присвоєння архівного номеру занесеного до системи та пізніше роздрукованого документу – кресленню чи специфікації виробу.

На основі результатів п'ятої роботи виконується шоста робота, що відповідає за формування журналу обліку документації.

В результаті виконання задачі «Облік конструкторської документації» ІС машинобудівельного підприємства утворюються такі документи:

- журнал обліку документації;
- специфікація виробу;
- креслення виробу;
- сповіщення про внесення змін до документу.

Виходячи з отриманих вимог можемо побудувати схему бізнес-транзакції для реалізації обліку конструкторської документації на рисунку 3.2.

Роботи, пов'язані з формуванням і корегуванням КД та занесенням розроблених документів до БД здійснює інженер-конструктор, журнал обліку документації начальник конструкторського бюро використовує для аналізу результатів виконаної роботи [23].

Задача «Облік конструкторської документації» ІС машинобудівельного підприємства пов'язана з задачею обліку працівників підприємства. Відомості про працівників – особиста інформація та вхідні дані користувачів для доступу до особистого кабінету вносяться до БД системи адміністратором ІС – працівником комп'ютерного відділу.

Дані, що використовуються в задачі «Облік конструкторської документації» ІС машинобудівельного підприємства, припускають введення інформації за допомогою клавіатури, а також використання даних з БД [24]. Введення даних здійснюється інженер-конструктором.

Вихідна інформація представлена екранними формами, роздрукованими документами. Одержувач документів – начальник конструкторського бюро та складальні цехи.

Умови припинення вирішення задачі «Облік конструкторської документації» ІС машинобудівельного підприємства автоматизованим способом:

- несанкціонований доступ до особистих даних співробітників або розробленої КД;
- несправність технічних засобів забезпечення задачі «Облік конструкторської документації» ІС машинобудівельного підприємства;
- аварійне відключення електроживлення на підприємстві або у конструкторському бюро;
- відсутність доступу до мережі і БД через апаратну несправність модему, проблеми, пов'язані з інтернет-з'єднанням або з сервером БД.

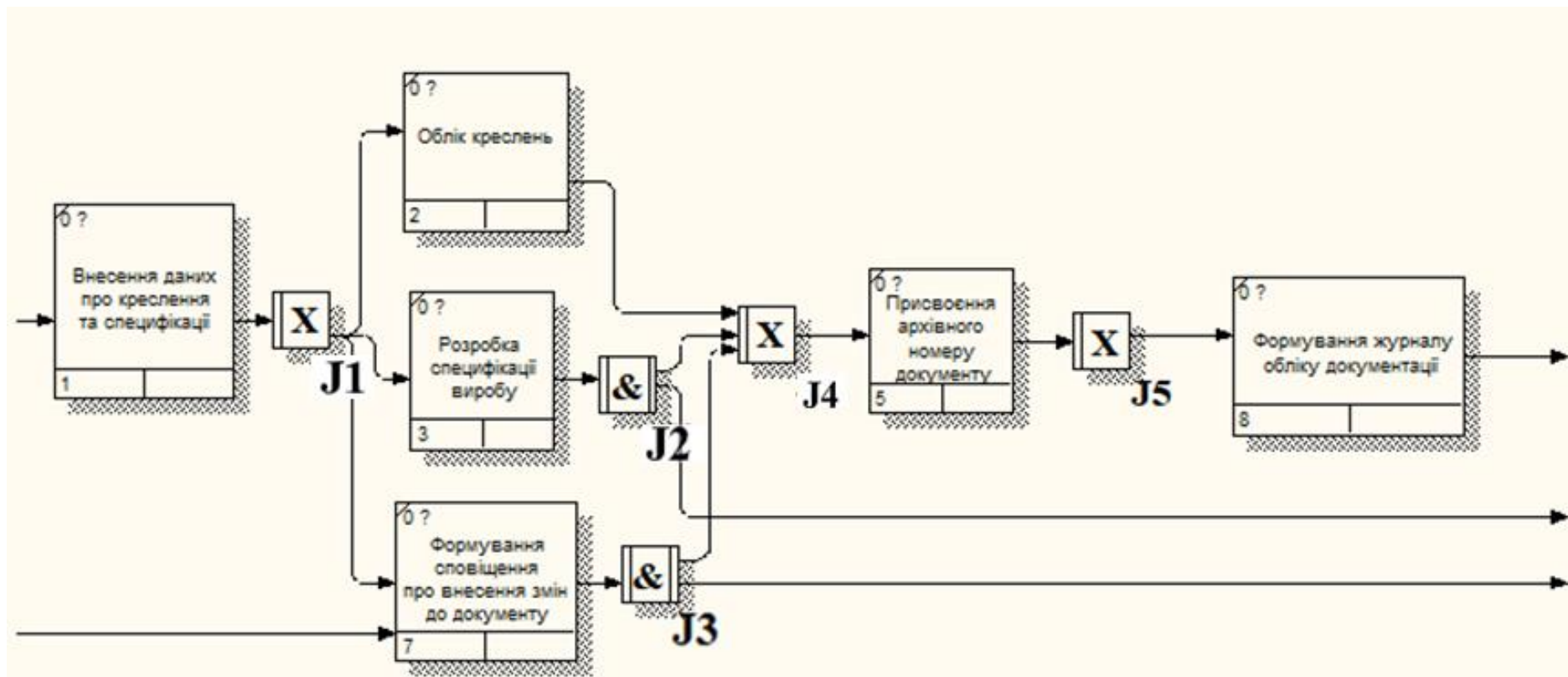


Рисунок 3.2 – Схема бізнес-транзакції задачі обліку конструкторської документації

Схема функціональної структури задачі «Облік конструкторської документації» ІС машинобудівельного підприємства представлена на контекстній діаграмі (рисунок 3.3) з виділенням основних функцій на діаграмі декомпозицій першого рівня (рисунок 3.4).

Контекстна діаграма потоків даних містить загальний блок, який визначає головну задачу – «Облік конструкторської документації» ІС машинобудівельного підприємства.

На вхід задачі подаються інформація про креслення та специфікацію виробу, графічне зображення креслення та інформація про зміни до документу.

На виході задачі отримуємо розроблені документи: специфікація виробу, журнал обліку документації та сповіщення про внесення змін до документу.

Задача «Облік конструкторської документації» ІС машинобудівельного підприємства призначена для автоматизації процесу обліку, зберігання та обігу КД на машинобудівальному підприємстві. Даний облік необхідний для виконання наступних завдань:

- підтримка і ведення БД для ідентифікації, зручного пошуку, перегляду та швидкого внесення змін до КД;
- контроль креслень та специфікацій виробів на технологічність;
- спрощення створення звітів про виконану роботу, за її результатами нарахування премій, присвоєння категорій співробітникам.

Обов'язками інженера-конструктора є розробка відповідної КД, використовуючи засоби автоматизації проектування, її супровід, перевірка, затвердження, та формування на їх основі звіту про результати виконаної роботи.

Запропоновані механізми автоматизації дозволяють зменшити витрати підприємства за рахунок скорочення часу на виконання операцій, знизити трудомісткість обробки КД, підвищити якість перевірки КД, забезпечити надійне зберігання даних, швидкий доступ до них, ефективний пошук, перегляд та швидке внесення змін до документації, запобігання дублювання, малу ймовірність введення помилкових даних.

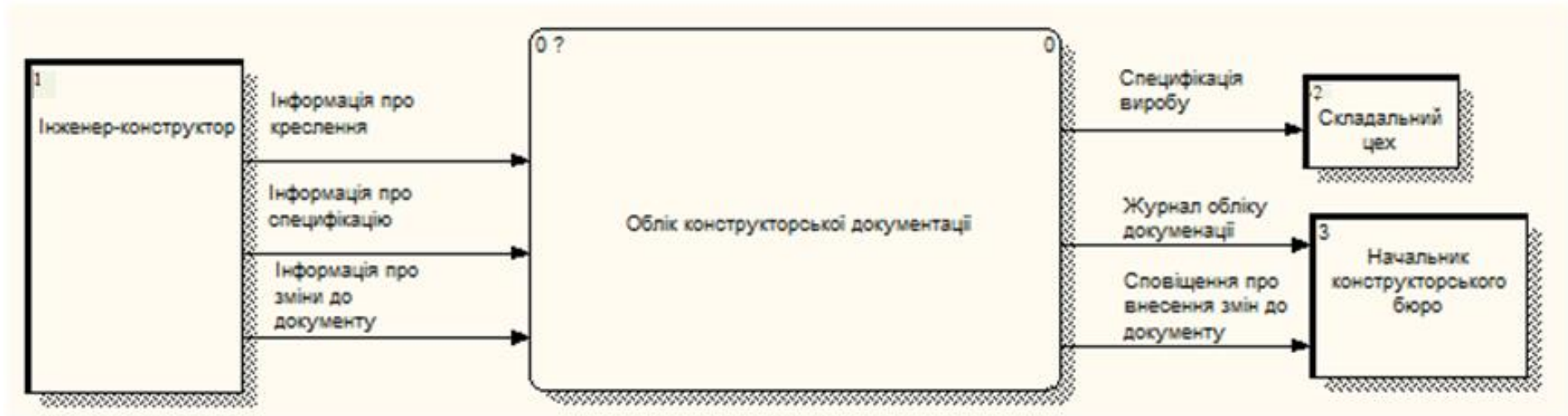


Рисунок 3.3 – Схема функціональної структури задачі «Облік конструкторської документації» ІС машинобудівельного підприємства

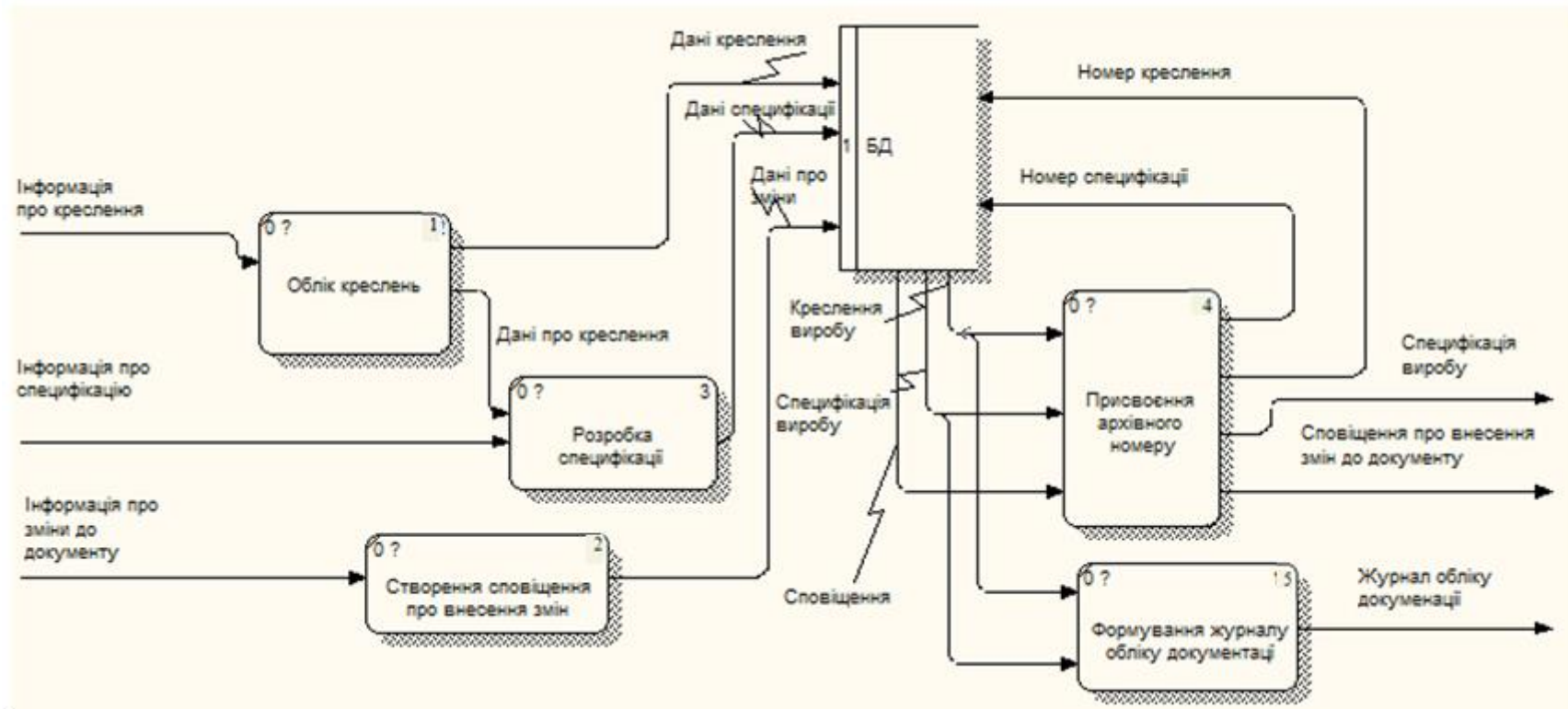


Рисунок 3.4 – Діаграма декомпозицій першого рівня

Роботи, пов'язані з формуванням і корегуванням КД та занесенням розроблених документів до БД здійснює інженер-конструктор, журнал обліку документації начальник конструкторського бюро використовує для аналізу результатів виконаної роботи [23].

Задача «Облік конструкторської документації» ІС машинобудівельного підприємства пов'язана з задачею обліку працівників підприємства. Відомості про працівників – особиста інформація та вхідні дані користувачів для доступу до особистого кабінету вносяться до БД системи адміністратором ІС – працівником комп'ютерного відділу.

Дані, що використовуються в задачі «Облік конструкторської документації» ІС машинобудівельного підприємства, припускають введення інформації за допомогою клавіатури, а також використання даних з БД [24]. Введення даних здійснюється інженер-конструктором.

Розроблений документ «Опис постановки задачі «Облік конструкторської документації» ІС машинобудівельного підприємства» приведений у додатку А.

Вихідна інформація представлена екранними формами, роздрукованими документами. Одержувач документів – начальник конструкторського бюро та складальні цехи.

Умови припинення вирішення задачі «Облік конструкторської документації» ІС машинобудівельного підприємства автоматизованим способом:

- несанкціонований доступ до особистих даних співробітників або розробленої КД;
- несправність технічних засобів забезпечення задачі «Облік конструкторської документації» ІС машинобудівельного підприємства;
- аварійне відключення електроживлення на підприємстві або у конструкторському бюро;
- відсутність доступу до мережі і БД через апаратну несправність модему, проблеми, пов'язані з інтернет-з'єднанням або з сервером БД.

### 3.3 Побудова сутностей задачі

Для представлення семантичної інформації про об'єкт автоматизації традиційно використовується модель "сутність-зв'язок" (entity-relationship model). Базовими елементами ER-діаграм є сутності, зв'язки між ними і їх атрибути [25].

Відомості про типи сутностей і зв'язків задачі «Облік конструкторської документації» ІС машинобудівельного підприємства представлені в таблицях 3.1 і 3.2 відповідно.

Таблиця 3.1 – Відомості про типи сутностей

Назва сутності	Опис	Псевдо-німи	Особливості використання
1	2	3	4
Співробітник	Інформація про співробітників	С	Обов'язкові для заповнення поля: «ID_співробітника», «Прізвище», «Ім'я», «Дата народження», «Дата зарахування», «Адреса».
Користувач	Інформація про користувачів ІС	К	Всі поля обов'язкові для заповнення
Тип користувача	Перелік типів користувачів	ТП	Всі поля обов'язкові для заповнення
Бюро	Інформація про бюро	Б	Всі поля обов'язкові для заповнення
Посада	Перелік посад співробітників	П	Всі поля обов'язкові для заповнення
Складальна одиниця	Інформація про складальні одиниці	СО	Всі поля обов'язкові для заповнення, окрім поля «Примітки». До складу СО може входити декілька деталей
Деталь	Інформація про деталі	Д	Всі поля обов'язкові для заповнення, окрім поля «Примітки»
Тип креслення	Перелік типів креслень	ТК	Всі поля обов'язкові для заповнення

Кінець таблиці 3.1

1	2	3	4
Креслення	Інформація про креслення	К	Всі поля обов'язкові для заповнення За одним кресленням може бути виготовлено декілька деталей
Специфікація	Інформація про специфікації СО	СПЦ	Всі поля обов'язкові для заповнення
Тип специфікації	Перелік типів специфікацій	ТС	Всі поля обов'язкові для заповнення
Сповіщення	Сповіщення на внесення змін до документу	СПВ	Всі поля обов'язкові для заповнення

Таблиця 3.2 – Відомості про типи зв'язків

Тип сутності	Тип зв'язку	Тип сутності	Кардинальність
Співробітник	Працює у	Бюро	М:1
Співробітник	Займає	Посада	М:1
Співробітник	Має привілеї такого користувача	Користувач	М:1
Співробітник	Складає	Сповіщення	1:М
Користувач	Має	Тип користувача	М:1
Складальна одиниця	Входить до	Специфікація	М:1
Деталь	Входить до	Специфікація	М:1
Креслення	Має	Тип креслення	М:1
Специфікація	Має	Тип специфікації	М:1

### 3.3.1 Опис атрибутів сутностей і їх доменів

Атрибут сутності – це характеристика, яка є деякою властивістю сутності. Найменування атрибута повинно бути виражено іменником в однині, іноді можливий більш детальний опис двома словами. Домен атрибута – це набір усіх допустимих значень, які може містити цей атрибут. Відомості про атрибути і домени атрибутів представлені в таблицях 3.3 і 3.4 відповідно.

Таблиця 3.3 – Відомості про атрибути

Тип сутності	Атрибут	Опис	Тип даних, довжина	Обмеження	Допустимість NULL
1	2	3	4	5	6
Співробітник	ID_Співробітника	Унікальний ідентифікатор співробітника в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Прізвище	Прізвище співробітника	Символьний, до 30 с.		Ні
	Ім'я	Ім'я співробітника	Символьний, до 30 с.		Ні
	По батькові	По батькові співробітника	Символьний, до 30 с.		Так
	Дата народження	Дата народження співробітника	Дата/час, короткий формат дати		Ні
	Дата зарахування	Дата зарахування на роботу	Дата/час, короткий формат дати		Ні
	Адреса	Чинна адреса, за якою проживаю співробітник	Символьний, до 50 с.		Так
	Телефон	Мобільний телефон співробітника	Символьний, до 20 с.		Ні
	ID_Посади	Унікальний ідентифікатор посади у системі	Числовий, до 15 с.	Зовнішній ключ	Так
	ID_Бюро	Унікальний ідентифікатор бюро у системі	Числовий, до 15 с.	Зовнішній ключ	Так
Бюро	ID_Бюро	Унікальний ідентифікатор бюро в ІС	Числовий, до 15 с.	Первинний ключ	Ні
	Найменування	Назва бюро	Символьний, до 30 с.		Ні

Продовження таблиці 3.3

1	2	3	4	5	6
Посада	ID_Посади	Унікальний ідентифікатор посади співробітника в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Найменування	Назва посади (інженер-конструктор, головний конструктор, і т.ін.)	Символьний, до 30 с.		Ні
Складальна одиниця	ID_Складальної одиниці	Унікальний ідентифікатор складальної одиниці в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Умовне позначення	Умовне позначення складальної одиниці у відповідності до діючого ДСТУ оформлення КД	Символьний, до 20 с.		Ні
	Найменування	Найменування складальної одиниці	Символьний, до 30 с.		Ні
	Кількість	Кількість складальних одиниць у кінцевому виробі	Числовий, до 5 с.		Ні
	Примітки	Маса складальної одиниці в кг	Числовий, до 5 с.		Так
Деталь	ID_Деталі	Унікальний ідентифікатор деталі в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Умовне позначення	Умовне позначення деталі у відповідності до діючого ДСТУ	Символьний, до 20 с.		Ні
	Найменування	Назва деталі	Символьний, до 100 с.		Ні
	Кількість	Кількість деталей, що входить до складу складальної одиниці	Числовий, до 5 с.		Ні
	Примітки	Маса деталі в кг	Числовий, до 5 с.		Так

Продовження таблиці 3.3

1	2	3	4	5	6
Креслення	ID_Креслення	Унікальний ідентифікатор креслення в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Умовне позначення	Умовне позначення креслення у відповідності до діючого ДСТУ оформлення КД	Символьний, до 20 с.		Ні
	Найменування		Символьний, до 30 с.		Ні
	ID_Типу креслення	Унікальний ідентифікатор типу креслення в системі	Числовий, до 15 с.	Зовнішній ключ	Ні
	ID_Співробітника	Унікальний ідентифікатор того, хто розробив креслення	Числовий, до 15 с.	Зовнішній ключ	Ні
Тип креслення	ID_Типу креслення	Унікальний ідентифікатор типу креслення в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Найменування	Назва креслення (складальний, деталі, загального вигляду)	Символьний, до 30 с.		Ні
Тип специфікації	ID_Типу специфікації	Унікальний ідентифікатор типу специфікації в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Найменування	Назва типу специфікації	Символьний, до 30 с.		Ні
Специфікація	ID_Специфікації	Унікальний ідентифікатор специфікації в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Найменування	Назва специфікації	Символьний, до 30 с.		Ні

Продовження таблиці 3.3

1	2	3	4	5	6
Специфікація	Умовне позначення	Умовне позначення специфікації у відповідності до діючого ДСТУ	Символьний, до 20 с.		Ні
	ID_Типу специфікації	Унікальний ідентифікатор типу специфікації в системі	Числовий, до 15 с.	Первинний ключ	Ні
	ID_Креслення	Унікальний ідентифікатор креслення в системі	Числовий, до 15 с.	Первинний ключ	Ні
Сповіщення	ID_Сповіщення	Унікальний ідентифікатор документації в системі	Числовий, до 15 с.	Первинний ключ	Ні
	ID_Співробітника	Унікальний ідентифікатор того, хто розробив креслення	Числовий, до 15 с.	Зовнішній ключ	Ні
	Умовне позначення	Умовне позначення сповіщення про зміни до документу	Символьний, до 20 с.		Ні
	Найменування	Назва документа	Символьний, до 30 с.		Ні
	Причина	Причина, з якою необхідно внести зміни до документу	Символьний, до 20 с.		Ні
	Зміст	Зміст внесених змін до документу	Символьний, до 200 с.		Ні
	ID_Креслення	Унікальний ідентифікатор креслення в системі	Числовий, до 15 с.	Зовнішній ключ	Ні

Кінець таблиці 3.3

1	2	3	4	5	6
Користувач	ID_ Користувача	Унікальний ідентифікатор користувача в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Логін	Логін користувача для авторизації	Числовий, до 6 с.		Ні
	Пароль	Пароль користувача для авторизації	Символьний, до 5 с.		Ні
Тип Користувача	ID_ Типу Користувача	Унікальний ідентифікатор типу користувача в системі	Числовий, до 15 с.	Первинний ключ	Ні
	Найменування	Назва типу користувача (адміністратор, співробітник)	Символьний, до 10 с.		Ні

Таблиця 3.4 – Відомості про домени атрибутів

Імя домену	Характеристика домену	Приклад допустимих значень
Номер телефону	Рядок змінної довжини, до 20 с.	+38(095)-836-29-01
Адреса	Рядок змінної довжини, до 100 с.	Бул. Двірцева, буд.53 кв.103
Умовне позначення креслення	Рядок змінної довжини, до 20 с.	1С.290.00.00 СК
Умовне позначення специфікації	Рядок змінної довжини, до 20 с.	1С.290.00.00
Маса	Рядок змінної довжини, до 20 с.	10 кг
Формат	Рядок змінної довжини, до 2 с.	A4
Логін	Рядок змінної довжини, до 6 с.	381641
Пароль	Рядок змінної довжини, до 5 с.	Qwer1

### 3.4 Опис схеми даних задачі

Для побудови логічної та фізичної схем даних задачі «Облік конструкторської документації» ІС машинобудівельного підприємства використовується модуль ERwin Data Model.

Логічна та фізична схеми даних задачі «Облік конструкторської документації» ІС машинобудівельного підприємства представлені у вигляді ER-діаграми відповідно на рисунках 3.5 та 3.6.

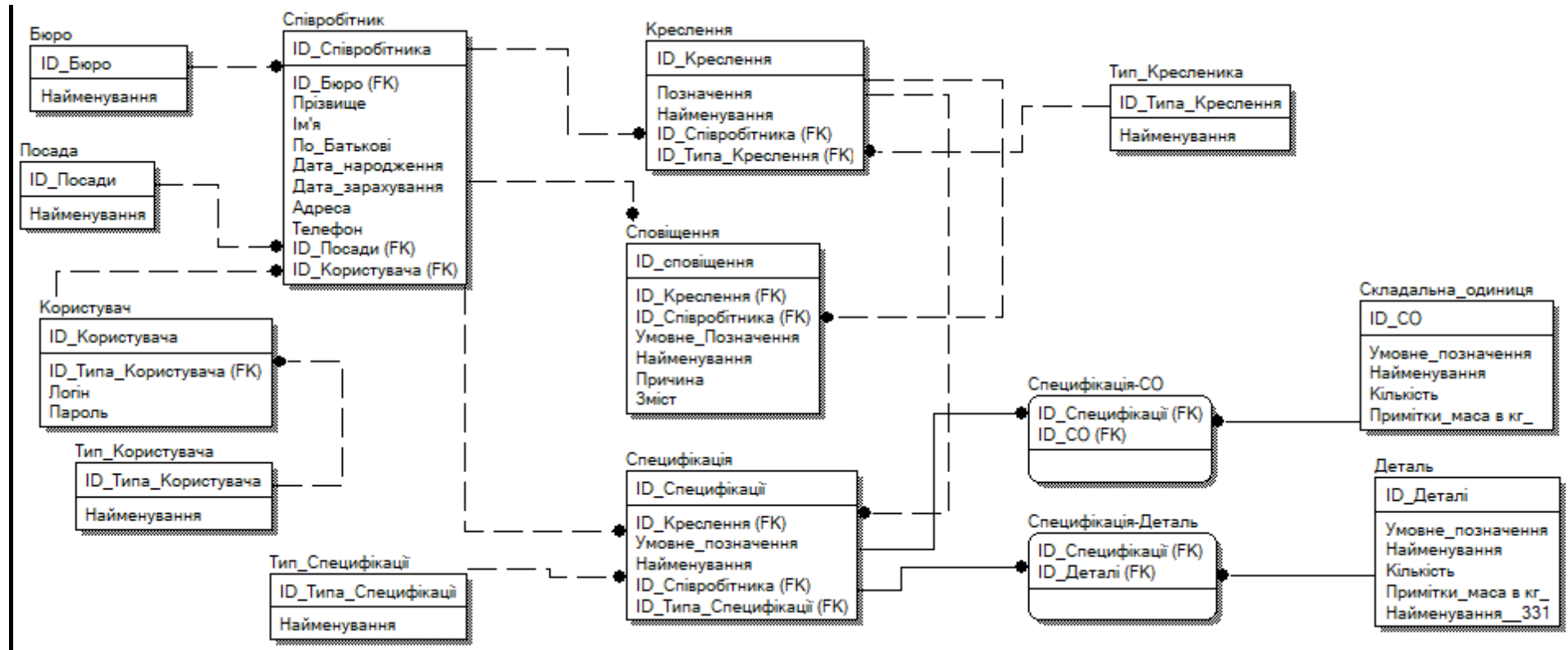


Рисунок 3.5 – Логічна схема даних задачі «Облік конструкторської документації» ІС машинобудівельного підприємства у вигляді ER-діаграми

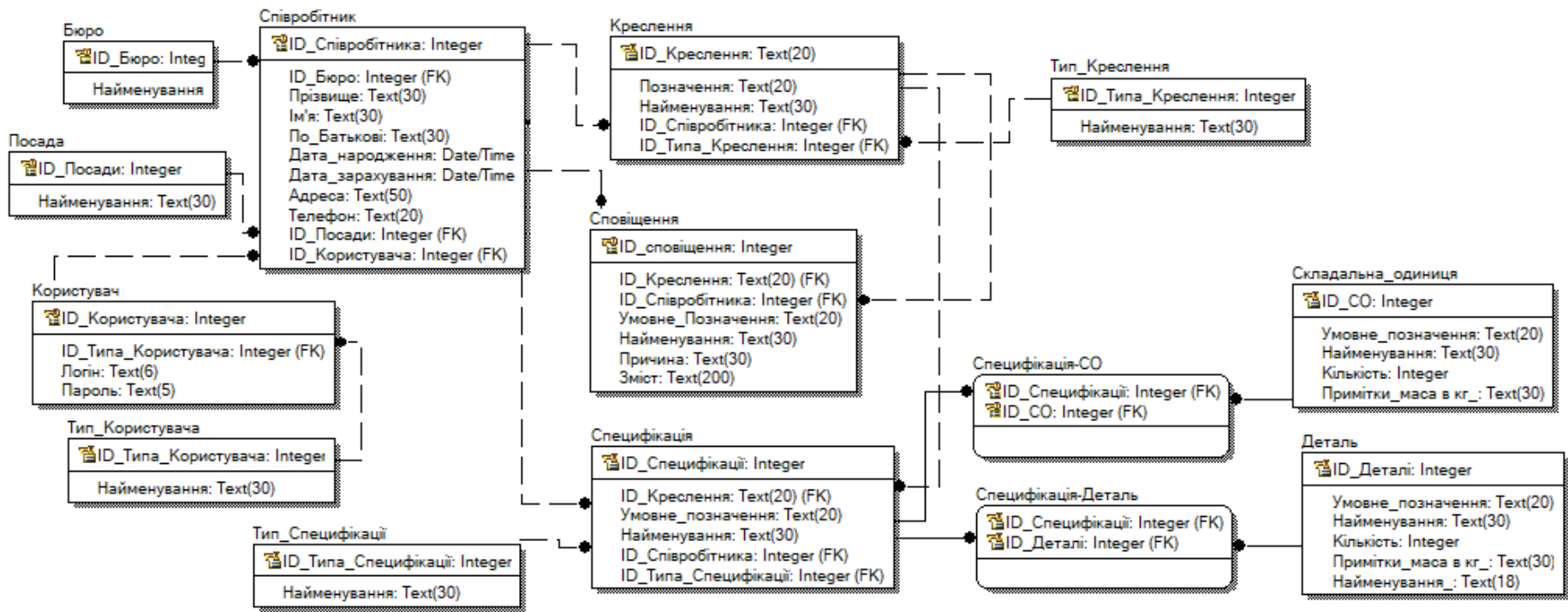


Рисунок 3.6 – Фізична схема даних задачі «Облік конструкторської документації» ІС машинобудівельного підприємства у вигляді ER-діаграми

Була проведена нормалізація, внаслідок чого схема даних задачі «Облік конструкторської документації» ІС машинобудівельного підприємства задовольняє третій нормальній формі, перелік відношень і атрибутів мінімальний, первинний ключ неподільний, кожен не ключовий стовпець незалежний один від одного, а залежить тільки від повного первинного ключа.

Для створення і роботи з БД використовувалася система управління БД PostgreSQL. PostgreSQL – це вільно поширювана об'єктно-реляційна система управління БД (ORDBMS). На відміну від СУБД MySQL, розробка якої спочатку орієнтувалася на web, розробка PostgreSQL орієнтується на використання в складних додатках. Методи доступу, які є загальними для MySQL і PostgreSQL включають в себе ADO.NET, JDBC і ODBC. ADO.NET представляє з себе набір інтерфейсів додатків (API), який використовуються для доступу до даних на основі XML. JDBC є API для мови Java, а ODBC – це стандартний API для доступу до БД PostgreSQL.

PostgreSQL має відкритий вихідний код, що забезпечує максимальну відкритість і доступність. СУБД також є крос-платформним продуктом і працює не тільки в широкому діапазоні діалектів UNIX, але і на платформі Microsoft Windows. Для роботи з PostgreSQL існує декілька інтерфейсів і бібліотек взаємодії з іншими мовами програмування: Java (JDBC), Perl, Python, Ruby, C, C ++, PHP, Lisp, Scheme і Qt [26].

**3.5 Висновки практичного рішення інтеграції ІТ-сервісів в бізнес-транзакції реалізованого власним методом.**

Розглянувши методи реалізації поставленої задачі у другому розділі було розроблено ІТ-сервіс на основі комбінації приведених методів (2.1., 2.2).

В результаті чого розроблено сценарій бізнес-транзакції на основі взаємодії ІТ-сервісу побудованого за допомогою нових методів web розробки

та взаємодією зі сценарієм бізнес-транзакції завдяки використанню сутностей БД з чіткою нормалізацією, що дало можливість запобігти можливим виникненням помилок бізнес-транзакції описаним у пункті(2.1.1), а саме:

- блокування запису;
- довго відкритої транзакції у базі даних;
- відкату транзакції у базі даних;
- виняткової ситуації під час надсилання нотифікації;
- припинення роботи процесу між операціями.

Завдяки чітко побудованому сценарію бізнес-транзакції та використанням комерційного рішення щодо інтеграції ІТ-сервісу, підприємство отримало готовий продукт, зроблений на основі нових технологій у web індустрії.

## 4 ОПИС ОТРИМАНИХ ТЕОРЕТИЧНИХ ТА ПРАКТИЧНИХ ДОСЛІДЖЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ

4.1 Обґрунтування необхідності використання інтегрованих сервісів.

4.1.1 Опис технічних характеристик необхідних для використання ІТ-сервісів.

Технічне забезпечення являє собою комплекс технічних засобів (КТЗ), що забезпечують збір, обробку та зберігання інформації для користувачів. Технічні засоби служать основою побудови ІС. Потужність засобів значною мірою визначає склад задач, що розв'язуються в даній предметній галузі. До технічного забезпечення ІС належать комп'ютерна техніка, засоби комунікації та оргтехніка.

Технічне забезпечення сучасних ІС – це комплекс різних видів техніки: обчислювальна техніка, периферійні пристрої, засоби автоматичного зчитування даних, офісне обладнання, комунікаційне обладнання, засоби передачі та обміну даними, комунікаційне обладнання, мережеве обладнання, засоби мультимедіа тощо [27].

Вимоги до КТЗ: висока ефективність функціонування; надійність; захист від несанкціонованого доступу; допустимі вартісні характеристики; мінімізація витрат на придбання, впровадження, експлуатацію та подальший супровід і технічне обслуговування.

Для функціонування системи, призначеної для автоматизації задачі «Облік конструкторської документації» ІС машинобудівельного підприємства КТЗ має включати:

- персональний комп'ютер;
- клавіатуру;
- комп'ютерну мишу;

- сервер;
- модем;
- комутатор кімнати;
- комутатор поверху;
- маршрутизатор;
- принтер;
- факс;
- джерело безперебійного живлення (ДБЖ).

Розглянемо інтеграцію теоретичних і практичних досліджень до ЗАТ НКМЗ. Інженерний корпус складається з 5 поверхів, кожен поверх – конструкторський відділ, який може містити у собі до 20 кімнат – конструкторських бюро.

Конструкторське бюро займає кімнату з можливістю розміщення до 20 робочих місць. Для більш наглядного представлення схеми мережі LAN розглянемо лише ділянку кімнати з чотирма робочими місцями для інженерів-конструкторів та начальника бюро, де на кожне робоче місце виділяється 2,5 кв.м. Основним компонентом мережі LAN є кімнатний комутатор – комутатор конструкторського бюро з портом (1) для з'єднання з магістраллю поверху типу 100Base-T і з портами (2-8) для підключення робочих станцій, одного принтера, факсу та серверу). Для підключення робочих станцій інженерів-конструкторів досить технології типу Fast Ethernet 100Base-T. Швидкість передачі інформації в створеній локальній мережі – 6Мгб / с. Схема підключення комплексу технічних засобів для конструкторського бюро представлена на рисунку 4.1.

Для того щоб організувати робочі місця користувачів системи необхідні наступні мінімальні параметри персональних комп'ютерів:

- тактова частота процесора не менше 800 МГц;
- об'єм ОЗП не менш 1 Гб;
- мінімум 512 Мб вільної пам'яті на локальному диску;
- операційна системи сімейства Windows не нижче Windows 7;

- підключення до комп'ютера принтера;
- засоби введення інформації: клавіатура і миша.

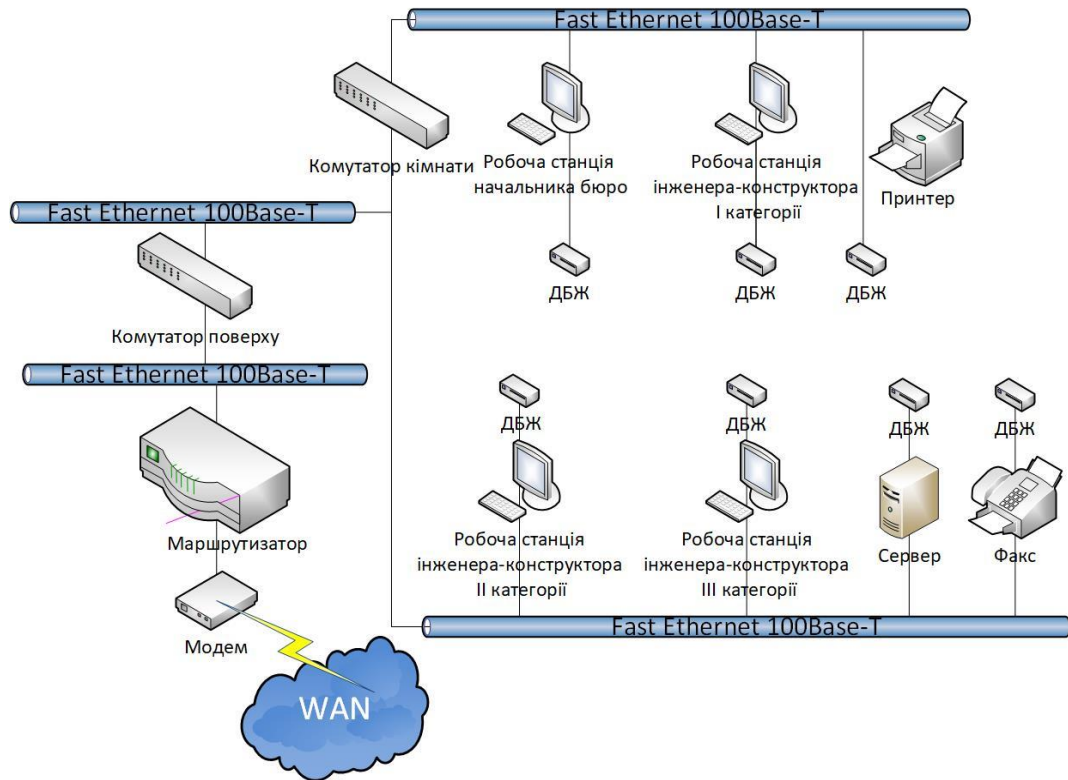


Рисунок 4.1 – Схема підключення комплексу технічних засобів для конструкторського бюро

Сервер, який буде зберігати інформацію про інженерів-конструкторів та результати розробки конструкторської документації, потребує наявності наступних параметрів:

- мінімум два процесора з тактовою частотою не менше 1 ГГц;
- мінімум 4 Гб вільного місця на локальному диску;
- мінімум 2 Гб оперативної пам'яті для нормального функціонування ОС і запущеної на ній СУБД.

Також необхідно забезпечити використання джерела ДБЖ, яке в разі незапланованого збою в режимі роботи мережі або проблем з постачання

електрики забезпечать резерв часу для коректного закриття додатків і виходу з системи.

Специфікація необхідного обладнання для технічного забезпечення задачі «Облік конструкторської документації» ІС машинобудівельного підприємства представлена в таблиці 4.1

Таблиця 4.1 – Специфікація обладнання для технічного забезпечення задачі «Облік конструкторської документації»

Найменування	Характеристика
1	2
Сервер	Процесор: DualCore Intel Core i3-3240 Частота процесора: 3400 MHz Материнська плата: Asus H61M-K, Обсяг HDD:500 ГБ Обсяг ОЗП: 4 ГБ Тип ОЗП: DDR3
Системний блок	Everest 1020_8426 Home&Office 1020 Серія процесору – Intel Pentium Gold Кількість ядер – 2 шт. Частота процесора – 3,7 ГГц ОЗП – DDR3, 4 ГБ Мережевий адаптер – 100 Мбіт / с
Монітор	LG 19M38A-B Діагональ екрану – 20 " Дозвіл екрану – 1600x900 Співвідношення сторін – 16: 9 Час реакції – 5 мс Споживання енергії (в роботі) – 20 Вт Споживання енергії (очікування) – 0,3 Вт

Продовження таблиці 4.1

1	2
Клавіатура	Logitech Wireless Combo MK270 Rus EER Інтерфейс підключення: USB Формат: повнорозмірна Розкладка клавіатури: українська, російська, англійська Розміри клавіатури: 460 x 218 x 40 мм
Миша	USB A4Tech XL-750BK USB Black Тип сенсору – лазерний Кількість кнопок – 2 + 1 шт. З'єднання з ПК – провідне Час відгуку сенсора – 1 мс
Принтер	Canon 1476C006AA LBP-653Cdw Формат – А4 Обсяг пам'яті – 32 Мб Максимальна роздільна здатність при друку 2400 x 600 dpi Максимальна швидкість друку: 20 стор. / Хв.
Факс	Panasonic KX-FT984 Black (KX-FT984UA-B) Формат паперу: форматна Друк: термоперенос Швидкість модему: 9.6 Кбіт / с Телефон: провідний Місткість: 20 аркушів Швидкий набір номерів (номерів в пам'яті): 100 шт.
ДБЖ	CyberPower UT450E 450VA Вихідна потужність: 450 ВА / 225 Вт Діапазон вхідного навантаження: 165 - 290 В Час роботи при повному навантаженні: 8 хвилин (90 Вт) Захист ліній передачі даних

Кінець таблиці 4.1

1	2
Комутатор для конструкторського бюро (для кімнати)	<p>D-Link DGS-1210-28</p> <p>Внутрішня пропускна здатність: 1.6 Гбіт / сек</p> <p>Пропускна здатність (пакетів в секунду): 1.19 Mpps</p> <p>Порти: 8 портів RJ-45 для 10 / 100Base-TX</p> <p>Типи кабелю: 10Base-T - UTP / STP, 100Base-TX - UTP / STP</p> <p>Загальний буфер пам'яті: 448 К</p> <p>Максимальний розмір пакета: 1536 bytes</p>
Комутатор для конструкторського відділу (для поверху)	<p>D-Link DES-1210-28/ME</p> <p>Внутрішня пропускна здатність: 12.8 Гбіт / сек</p> <p>Пропускна здатність (64-байтних пакетів в секунду): 9,5 Mpps</p> <p>Буфер пакетів: 384 К</p> <p>Форм-фактор: Настільний / стічний</p> <p>Таблиця MAC-адрес: 8К записів на пристрій</p>
Маршрутизатор	<p>M4100-12GF (GSM7212F)</p> <p>Порти: 12 x RJ45 10/100/1000 Мбіт</p> <p>Внутрішня пропускна здатність: 24 Гбіт / сек</p> <p>Пропускна здатність (пакетів в секунду): 17,857 Mpps</p> <p>Кількість VLAN 1024 (802.1Q) одночасно</p> <p>Таблиця MAC-адрес: 16К записів на пристрій</p>
Модем	<p>D-Link DSL-224</p> <p>Інтерфейси: DSL; RJ-11 LAN 10/100BASE-TX</p> <p>Стандарти: VDSL/ADSL: VDSL2: ITU G.993.2; ADSL: Multi-mode, Типи підключення WAN: PPPoA, PPPoE, IPv6 PPPoE</p> <p>Сетевые функции: DHCP-сервер/relay; расширенная настройка встроенного DHCP-сервера</p>

#### 4.1.2 Вимоги до взаємодії користувачів задачі.

Під час вирішення задачі «Облік конструкторської документації» ІС машинобудівельного підприємства особлива увага приділяється візуальному забезпеченні системи. Проектування та розробка графічного інтерфейсу користувача відбувається згідно з правилами проектування програмних інтерфейсів для зручної взаємодії користувачів задачі.

Інтерфейс ПЗ повинен відповідати наступним критеріям:

- повністю відповідати завданням користувача;
- не містити надлишкових функцій, не потрібних для виконання безпосередніх операцій з обліку КД;
- за рахунок своєї простоти зменшувати тривалість виконання операцій та час на прийняття рішень користувачем;
- підвищувати здатність користувача фокусуватися на своїх безпосередніх цілях та задачах;
- за рахунок своєї зрозумілості та логічності зменшувати кількість помилок користувача через «людський чинник»;
- відповідати рівню навиків роботи з комп'ютером користувача.

На сприйняття користувачем інтерфейсу також впливають фактори розмітки сторінки (візуального вікна програми): візуальна ієрархія елементів інтерфейсу, угруповання, позиціонування і вирівнювання елементів на сторінці, використання шаблонів «динамічних дисплеїв».

Основними компонентами інтерфейсу системи для вирішення задачі «Облік конструкторської документації» ІС машинобудівельного підприємства є кнопки, текстові вставки, списки, таблиці та інші елементи управління, представлені технологією JavaFX.

4.1.3 Методичні рекомендації щодо застосування сервісу інтегрованого під час практичного проектування.

Адмініструванням, тестуванням та впровадженням системи, що вирішує задачу «Облік конструкторської документації» ІС машинобудівельного підприємства займаються співробітники комп'ютерного відділу, які відповідають за технічну підтримку системи: контролюють та підтримують функціонування системи в режимі роботи без перебоїв.

Для забезпечення роботи ІС, вся необхідна конструкторська документація та звіти за результатами роботи інженерів-конструкторів повинні зберігатися в електронному вигляді в локальній мережі інженерного корпусу.

Кожен користувач системи повинен мати власний логін та пароль для авторизації. Кожній групі користувачів призначається роль, що визначає їх рівень доступу до даних. У системі існує всього три типи групи користувачів, що розрізняються правами доступу: адміністратор ІС, начальник конструкторського бюро та інженер-конструктор.

БД задачі повинна зберігати актуальні дані про співробітників конструкторських бюро та розроблену ними чи під їх керівництвом документацію.

На етапі налаштування системи перевіряється коректність звітних форм, розподілення прав доступу та наявність усіх необхідних довідників. На етапі тестування в систему заносяться тестові дані та проводиться попереднє випробування системи з метою виявлення помилок. Після етапу дослідницької експлуатації для порівняння результатів роботи нової системи з отриманими вручну результатами, впровадження системи завершується та починається промислова експлуатація системи. Це означає, що системи цілком придатна для використання та відповідає усім висунутим вимогам.

4.2 Підсумкова характеристика практичного використання ІТ-сервісу інтегрованого в бізнес-транзакцію.

Розроблений ІТ-сервіс інтегрований за до бізнес-транзакції за допомогою власних рішень, в кінцевому варіанті має здебільшого переваги ніж недоліки, оскільки поставлена задача має бути вирішена з чітко поставленими вимогами, та мати за собою можливість малозатратної інтеграції на підприємство, можна судити що доцільність обраного підходу інтеграції ІТ-сервісу на результатах досліджень має більш вигідніші для обраного типу підприємств наслідки.

Зручність інтерфейсу інтегрованого ІТ-сервісу (приклади представлені у «Додатку Б») в свою чергу є невід'ємним критерієм якості розробленого ІТ-сервісу.

Розробка ІТ-сервісу для виконання задачі була виконана з використання мови програмування Java.

Java – об'єктно-орієнтована мова програмування, розроблена компанією SunMicrosystems. Програми Java компілюються в спеціальний байт-код, тому вони можуть працювати на будь-якій віртуальній Java-машині (JVM) незалежно від комп'ютерної архітектури [28].

Перевагою подібного способу виконання програм є повна незалежність байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина.

Іншою важливою особливістю технології Java є гнучка система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання.

Усередині Java існують кілька основних сімейств технологій:

- Java SE – Java Standard Edition, основне видання Java, містить компілятори, API, JRE (Java Runtime Environment);
- Java EE – Java Enterprise Edition, являє собою набір специфікацій для створення ПЗ рівня підприємства;
- Java ME – Java Micro Edition, створена для використання в пристроях, обмежених за обчислювальною потужністю, наприклад, в мобільних телефонах або вбудованих системах;
- JavaFX – технологія призначена для створення графічних інтерфейсів корпоративних додатків і бізнесу. FXML – це формат файлу, який використовує JavaFX для створення макета екранів [29].

У якості середовища розробки ІТ-сервісу задачі було обрано JetBrains IntelliJ IDEA – середовище розробки на мові Java. IntelliJ IDEA, високотехнологічний комплекс тісно інтегрованих інструментів програмування, що включає інтелектуальний редактор вихідних текстів з розвиненими засобами автоматизації, інструменти рефакторинга коду, механізми інтеграції з середовищем тестування і системами управління версіями, унікальний інструмент оптимізації та перевірки коду Code Inspection, а також візуальний конструктор графічних інтерфейсів.

IDEA дозволяє спростити процес пошуку потрібних класи або файлів, дозволяє створювати у вихідних текстах нумеровані закладки для переходу до необхідних фрагментів коду, швидко переміщатися до місця внесення останніх змін. Інтелектуальна функція Find Usages дозволяє знайти в проєкті всі посилання і виклики заданого класу, методу або змінною.

Всі вище перераховані технології розробки на відміну від класичних методів запропонованих раніше мають розповсюдженість на ринку ІТ, що робить розроблений сервіс не тільки інноваційним а й легко обслуговуваним у майбутньому, що приведе за собою зменшення витрат на подальше вдосконалення системи у разі необхідності.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи були досліджені методи інтеграції ІТ-сервісів в бізнес-транзакції, а саме метод «САГА» та метод інтеграції та побудови бізнес-транзакцій за допомогою мереж Петрі.

В результаті теоретичних досліджень описаних методів, кожним із них було сконструйовано моделі бізнес-транзакцій та розроблено концептуальне рішення їх інтеграції. Завдяки отриманим даним при теоретичних дослідженнях поставленої задачі інтеграції ІТ-сервісів в бізнес-транзакцію в практичній частині кваліфікаційної роботи було розроблено сценарій бізнес-транзакції на основі комбінації сучасних методів моделювання бізнес-транзакцій для використання її у галузі обліку конструкторської документації на підприємствах, а також реалізовано ІТ-сервіс побудований на основі сучасних web технологій.

Вирішення задачі інтеграції ІТ-сервісу ґрунтується на основі результатів передпроектного аналізу об'єкта автоматизації та дослідження розглянутої предметної галузі. При цьому були вивчені проблеми та особливості функціонування конструкторського документообігу на обраному для впровадження отриманих результатів дослідження підприємстві. Згідно з цим були обрані відповідні рішення з розв'язку поставленої функціональної задачі та її забезпечення.

Розроблено рішення з інформаційного забезпечення поставленої задачі в галузі використання побудованої бізнес-транзакції, описано сутності, їх атрибути та зв'язки задачі. Було розроблено логічну та фізичну схеми БД задачі. Також була створена схема роботи задачі, за допомогою якої була виконана програмна реалізація задачі об'єктно-орієнтованою мовою програмування Java та створена схема підключення комплексу технічних засобів. Графічний матеріал атестаційної роботи наведений у додатку Г.

Запропоновані метод інтеграції ІТ-сервісів в бізнес-транзакції дозволяють зменшити витрати підприємства, скоротити час на виконання операцій, знизити трудомісткість на обробку КД, підвищити якість перевірки КД, забезпечити надійне зберігання даних, швидкий і ефективний пошук, перегляд та швидке внесення змін до документації, запобігання дублювання, малу ймовірність введення помилкових даних.

Розглядаючи отримані результати досліджень слід зазначити необхідність продовження досліджень методів інтеграції ІТ-сервісів для подальшого відточення розробки більш нових технологій. Хоча відповідна тематика має вже перевірені методи її реалізації, слід акцентувати увагу на тому, що не кожен з цих методів доцільно використовувати для всіх типів поставленої задачі, оскільки розглянута у дослідженні галузь має строгу необхідність дотримання чітких норм та правил при вирішенні поставленої задачі, можна зробити висновки, що методологія інтеграції ІТ-сервісів в бізнес-транзакції потребує використання комбінацій запропонованих на даний час методів, а отже в майбутньому потребує вдосконалення існуючих рішень для більш зручного та універсального вирішення поставлених задач.

Дана кваліфікаційна робота була виконана у відповідності до державних стандартів оформлення звітів у сфері науки і техніки [22]. Перелік посилань оформлено з використанням державних стандартів оформлення бібліографічних посилань [23].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ**

1. Стейр Р М, Рейнольдс Г. Основи бізнес-інформаційних систем. Thomson Learning; 2008, 118-129;
2. Laudon К С, Laudon J Р. Інформаційні системи управління: управління цифровою фірмою. Видання 12, Прентіс Хол; 2012 рік;
3. Haugen В., Fletcher Т. Multi-Party Electronic Business Transactions URL: <http://logisticalsoftware.com/MultiPartyBusinessTransactions.PDF>
4. Papazoglou М.Р. Web Services and Business Transactions // World Wide Web: Internet and Web Information Systems. 2003. No. 6. P. 49–91.
5. Little М. Transactions and Web Services // Communications of the ACM. 2003. No. 10. P. 49- 54
6. Papazoglou М.Р., Kratz В. Web services technology in support of business transactions URL: <http://arno.uvt.nl/show.cgi?fid=69154>.
7. Бізнес-транзакцій (business transaction) в процесах автоматизації компаній URL: <http://4pll.com/301015.html>.
8. Свистунов А.Н. Построение распределенных систем на Java URL: <http://www.intuit.ru/department/pl/distrsysjava/14/>.
9. Sun С., Aiello М. Requirements and Evaluation of Protocols and Tools for Transaction Management in Service Centric Systems [електронний ресурс]. URL: <http://www.cs.rug.nl/~aiellom/publications/refs07.pdf>.
10. Wang Y., Weihai Y. Adaptable Transaction Processing in the Web Services Domain URL: <http://www.nik.no/2004/bidrag/Wang.pdf>
11. Schmit В. Towards Transactional Web Services URL: <http://www.infosys.tuwien.ac.at/Staff/sd/papers/Towards%20Transactional%20Web%20Services.pdf>.
12. Doreen Т. An analysis of transactions in service-centric systems [електронний ресурс]. URL: <http://www.cs.rug.nl/~aiellom/tesi/tuheirwe.pdf>.

13. Фаулер, М. Архітектура корпоративних програмних програм / Мартін Фаулер. -М. : Видавничий дім «Вільямс», 2006. -544с.
14. Пітерсон, Д. Теорія мереж Петрі та моделювання систем / Дж. Пітерсон. -М. : Мир, 1984. -264с.
15. Котов, В. Є. Мережі Петрі/В. Є. Котов. -М. : Наука. Головна редакція фізико-математичної літератури, 1984. -160с.
16. van der Aalst, W. Application of Petri Nets до Workflow Management / W.M.P. van der Aalst // Journal of Circuits, Systems and Computers. -1998.-Vol. 8, No1. -P.21-66.
17. Hartonas, C. South-East European Network on Formal Methods / Chrysafis Hartonas // A CombinedOpen Petri Net і Process Algebraic approach to Message-Passing Services. -2007
18. Jensen, K. An Introduction до Theoretical Aspects of Coloured Petri Nets / Kurt Jensen // A Decade of Concurrency, Lecture Notes in Computer Science/ J.W. de Bakker, W.-P. de Roever, G. Rozenberg. -Springer-Verlag, 1994.- P. 230-272.
19. Jensen, K. Coloured Petri Nets і CPN інструменти для моделювання і validation of concurrent systems / Kurt Jensen, Lars Michael Kristensen, Lisa Wells // Міжнародні Journal on Software Tools for Technology Transfer (STTT). -2007. - Vol. 9, No 3. -P.213-254
20. ДСТУ ГОСТ 2.601:2006 Єдина система конструкторської документації (ЄСКД). Експлуатаційні документи. – К.: Держспоживстандарт України, 2007. – 31 с.
21. Скударь Г. М. Ощадливе виробництво. Раціональна організація робочих місць: навч. посіб. – Краматорськ: НКМЗ, 2009. – 20 с.
22. Сагайда П. І. Розробка та організація баз даних у системах автоматизації проектування та управління / П. І. Сагайда. – Краматорськ: ДДМА, 2013. – 160 с.
23. Гайна Г. Основи проектування баз даних / Г. Гайна. – К: Кондор, 2018. – 204 с.

24. Uorsli D. PostgreSQL. Dlya professionalov / D. Uorsli, D. Drejk. – Spb: Piter, 2003. – 496 s.
25. Технічне забезпечення інформаційних систем. URL: [https://studopedia.com.ua/1\\_301539\\_tehnichne-zabezpechennya-informatsiynih-sistem.html](https://studopedia.com.ua/1_301539_tehnichne-zabezpechennya-informatsiynih-sistem.html) (дата звернення: 27.05.2019).
26. Schildt G. Java. Polnoe rukovodstvo + code / G. Schildt. – M.: VD Vil'yams, 2012. – 1104 s.
27. Topley K. JavaFX Developer's Guide / K. Topley. – Eastbourne: Gardners Books, 2010. – 1152 c.
28. Hector Garcia Molina, Kenneth Salem, “Sagas” — Princeton University, 1987 г. URL:<ftp://ftp.cs.princeton.edu/reports/1987/070.pdf>
29. Philip A. Bernstein, Nathan Goodman, Multiversion Concurrency Control - Theory and Algorithms, 1982 г.
30. Томас Кайт, Кун Дарл Oracle для професіоналів, 2015 г.
31. Eventuate Tram Saga, Sagas framework for Java URL:<https://github.com/eventuate-tram/eventuate-tram-sagas>
32. Cinema Framework. Фреймворк для построения оркестрируемых sag. URL:<https://github.com/codedoneteam/cinema>