

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ТА
ЗАСОБІВ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS
AND TOOLS FOR AUTOMATING BUSINESS PROCESSES

Виконав: студент 2 курсу, групи КНТ-121м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Куцак С.О.

(прізвище та ініціали)

Керівник Гладкова О.М.

(прізвище та ініціали)

Рецензент Зеленьова І.Я.

(прізвище та ініціали)

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

Факультет комп'ютерних наук та технологій
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

Куцака Сергія Олександровича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів та засобів автоматизації бізнес-процесів. Research and Software Implementation of Methods and Tools for Automating Business Processes

керівник проєкту (роботи) Гладкова Ольга Миколаївна, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “04” листопада 2022 року №363

2. Строк подання студентом проєкту (роботи) 1 грудня 2022 року

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Дослідження особливостей розробки автоматизованої системи управління підприємством. 2. Аналіз та вибір системи управління бізнес-процесами. 3. Програмна реалізація підсистеми автоматизації бізнес-процесів. 4. Методика використання розробленої підсистеми для автоматизації бізнес-процесу «Оформлення онлайн-замовлення».

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Гладкова О.М., доцент		
Нормоконтроль	Белова А.В., асистент		

7. Дата видачі завдання “5” вересня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2–3 тижні	Розділ 1
3	Дослідження методів та засобів вирішення задачі	4–5 тижні	Розділ 1
4	Моделювання та моніторинг бізнес-процесу в обраному середовищі	6–7 тиждень	Розділ 1
5	Розробка програмного забезпечення	8–10 тижні	Розділ 2
6	Розробка методики використання розробленого програмного забезпечення	11–12 тиждень	Розділ 3
7	Оформлення пояснювальної записки та документів до неї.	13–14 тижні	ПЗ, Додатки
8	Нормоконтроль та рецензування.	15 тиждень	
9	Захист роботи.	16 тиждень	

Студент

_____ Куцак С. О.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

_____ Гладкова О.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
103 с., 44 рис., 5 табл., 2 додатки, 27 джерел.

АВТОМАТИЗОВАНА СИСТЕМА УПРАВЛІННЯ ПІДПРИЄМСТВОМ,
БІЗНЕС-ПРОЦЕС, МОДЕЛЮВАННЯ, МОНІТОРИНГ, ІНТЕРНЕТ-МАГАЗИН,
ОНЛАЙН ЗАМОВЛЕННЯ.

Об'єкт дослідження – процес автоматизації бізнес-процесів.

Предмет дослідження – методи та засоби автоматизованого управління бізнес-процесами.

Мета роботи – дослідження та програмна реалізація підсистеми автоматизації для моделювання та моніторингу бізнес-процесів в середовищі BPM-системи.

Матеріали, методи та технічні засоби: середовища Intalio | Designer та Intalio | Server, мова опису визначень та реалізацій процесів XPDL, мова формального опису бізнес-процесів та протоколів їх взаємодії між собою BPEL, персональний комп'ютер з процесором Intel Core 2 Duo під управлінням операційної системи Microsoft Windows 10.

Результати. Розроблено підсистему автоматизації бізнес-процесу «Оформлення online-замовлення», що дозволяє користувачеві в режимі реального часу швидко замовити в Інтернет-магазині продукцію, що його цікавить, а адміністратору надає інформацію про стан бізнес-процесу.

Висновки. Сучасні системи управління бізнес-процесами надають потужні інструменти моделювання та моніторингу, а також інструментальні засоби для програмної реалізації додаткових функцій з метою підвищення ефективності автоматизованих систем управління підприємством.

Галузь використання – практика автоматизованого управління підприємством.

ABSTRACT

Explanatory note to the master thesis: 103 pages, 44 figures, 5 tables, 2 appendixes, 27 sources.

ENTERPRISE AUTOMATED MANAGEMENT SYSTEM, BUSINESS PROCESS, SIMULATION, MONITORING, INTERNET STORE, ONLINE ORDERING.

The object of study is the process of automating business processes.

The subject of the study are methods and tools of automated management of business processes.

The goals of the work are research and software implementation of the automation subsystem for modeling and monitoring business processes in the environment of the BPM-system.

Materials, methods and technical tools: Intalio | Designer and Intalio | Server environments, XPDL language for describing definitions and implementations of processes, BPEL language for formally describing business processes and their interaction protocols, a personal computer with an Intel Core 2 Duo processor running the Microsoft Windows 10 operating system.

The results. A business process automation subsystem "Online order processing" has been developed, which allows the user to quickly order the products he is interested in from the online store in real time, and it provides the administrator with information about the state of the business process.

Conclusions. Modern business process management systems provide powerful modeling and monitoring tools, as well as tools for software implementation of additional functions in order to increase the efficiency of automated enterprise management systems.

The field of application is the practice of enterprise automated management.

ЗМІСТ

	С.
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК.....	8
ВСТУП.....	9
1 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РОЗРОБКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ПІДПРИЄМСТВОМ	10
1.1 Бізнес-процеси як основа АСУП.....	10
1.2 Оптимізація бізнес-процесів.....	12
1.3 Аналіз особливостей технології Workflow.....	16
1.4 Постановка задач магістерської роботи	19
2 АНАЛІЗ ТА ВИБІР СИСТЕМИ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ	20
2.1 Особливості та переваги BPM систем	20
2.2 Огляд структури BPMS	22
2.3 Порівняльний аналіз та вибір BPM-системи.....	24
2.4 Структура робочого середовища	28
2.5 Висновки до розділу 2	30
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ	32
3.1 Опис бізнес-процесу та структура підсистеми автоматизації.....	32
3.2 Моделювання процесу «Оформлення online-замовлення».....	34
3.2.1 Використання BPMN-нотації	35
3.2.2 Створення форм	37
3.2.3 Редагування діаграми бізнес-процесу та налагодження зв'язків	38
3.3 Програмне створення форм з використанням мови XPDL.....	40
3.4 Моніторинг бізнес-процесу.....	44
3.4.1 Запуск процесу «Оформлення online-замовлення»	46
3.4.2 Заповнення форм процесу «Оформлення online-замовлення».....	46
3.4.3 Адміністрування процесів	46
3.5 Розробка підсистеми автоматизації бізнес-процесу «Оформлення online-замовлення»	47

3.6 Висновки до розділу 3	52
4 МЕТОДИКА ВИКОРИСТАННЯ РОЗРОБЛЕНОЇ ПІДСИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСУ «ОФОРМЛЕННЯ ONLINE- ЗАМОВЛЕННЯ».....	53
4.1 Призначення підсистеми та інструкція з використання	53
4.2 Повідомлення про помилки	59
4.3 Висновки до розділу 4	61
ВИСНОВКИ	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	63
ДОДАТОК А Текст програми	66
ДОДАТОК Б Слайди презентації.....	96

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

BPEL – Business Process Execution Language;
BPM – Business Process Management;
BPMN – Business Process Modeling Notation;
BPMS – Business Process Management System/Software;
CMS – Content Management System;
CRM – Customer Relationship Management;
DMN - Decision Model and Notation;
ERP – Enterprise Resource Planning System;
REST - Representational State Transfer;
XML – Extensible Markup Language;
XPDL – XML Process Definition Language;
АСУП – автоматизована система управління підприємством;
БП – бізнес-процес;
ЖЦ – життєвий цикл;
ІМ – Інтернет-магазин;
ІС – інформаційна система;
ІТ – інформаційна технологія;
ПЗ – програмне забезпечення.

ВСТУП

На сьогоднішній день керівники підприємств та компаній, усвідомлюють, що для того, щоб бути конкурентоспроможними, підвищити продуктивність і зробити бізнес більш ефективним, необхідно здійснити кроки, спрямовані на оптимізацію бізнес-процесів (БП). Щоб визначити, що саме необхідно оптимізувати, насамперед потрібно мати чітке розуміння процесів, залучених до конкретного бізнесу [1].

Для підтримки процесів управління бізнесом сьогодні на ринку програмного забезпечення пропонується спеціалізований клас продуктів – системи управління БП (Business Process Management, BPM) [2].

Найбільш відомими BPM-системами є: ARIS Express, JBoss jBPM, RUNA WFE, BizAgi, Intalio|BPMS, Camunda, IBM WebSphere Business Modeler, Fox Manager BPA.

Кожна з цих систем має свої переваги та недоліки, обмеження та вимоги до вхідної інформації, але жодна з систем не враховує особливості ведення бізнесу в мережі Інтернет. Але електронна торгівля розвивається стрімкими темпами, що було зокрема спричинено пандемією COVID-19. Тому, тема роботи, що пов'язана з розробкою підсистеми автоматизації БП для Інтернет-магазину є актуальною.

Метою роботи є дослідження та програмна реалізація підсистеми автоматизації для моделювання та моніторингу бізнес-процесів в середовищі BPM-системи.

1 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РОЗРОБКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ПІДПРИЄМСТВОМ

1.1 Бізнес-процеси як основа АСУП

БП - це ряд певних завдань, що піддаються виміру, виконуються людьми і системами, спрямовані на досягнення заздалегідь запланованого результату [3], [4].

БП притаманні такі важливі риси:

- мають внутрішніх та зовнішніх користувачів;
- діють усередині підрозділів компанії та між ними, а також між різними організаціями;
- засновані на способі виконання робіт, властивому тій чи іншій організації.

БП мають три основні елементи: місце, об'єкт, завдання:

- місце, де відбуваються процеси;
- об'єкти - можуть бути фізичними та інформаційними, власне процеси є результатом дій із об'єктами;
- завдання, тобто робота з метою впливу на об'єкти [5].

Таким чином, БП є набором взаємопов'язаних бізнес-процедур (функцій або дій, які формують результат, що має цінність для споживача), в результаті яких виробляється певна група продуктів і послуг. Усі БП існують для виконання функцій підприємства і мають відповідати встановленій у ньому ієрархії цілей.

Задача проєктування та впровадження автоматизованої системи управління підприємством (АСУП), що заснована на БП, є надзвичайно важливою в рамках стратегії бізнесу.

Для опису стратегії діяльності підприємства створюється стратегічна мапа (strategy map), що представляє собою набір причинно-наслідкових зв'язків на декількох рівнях (рис. 1.1) [6].

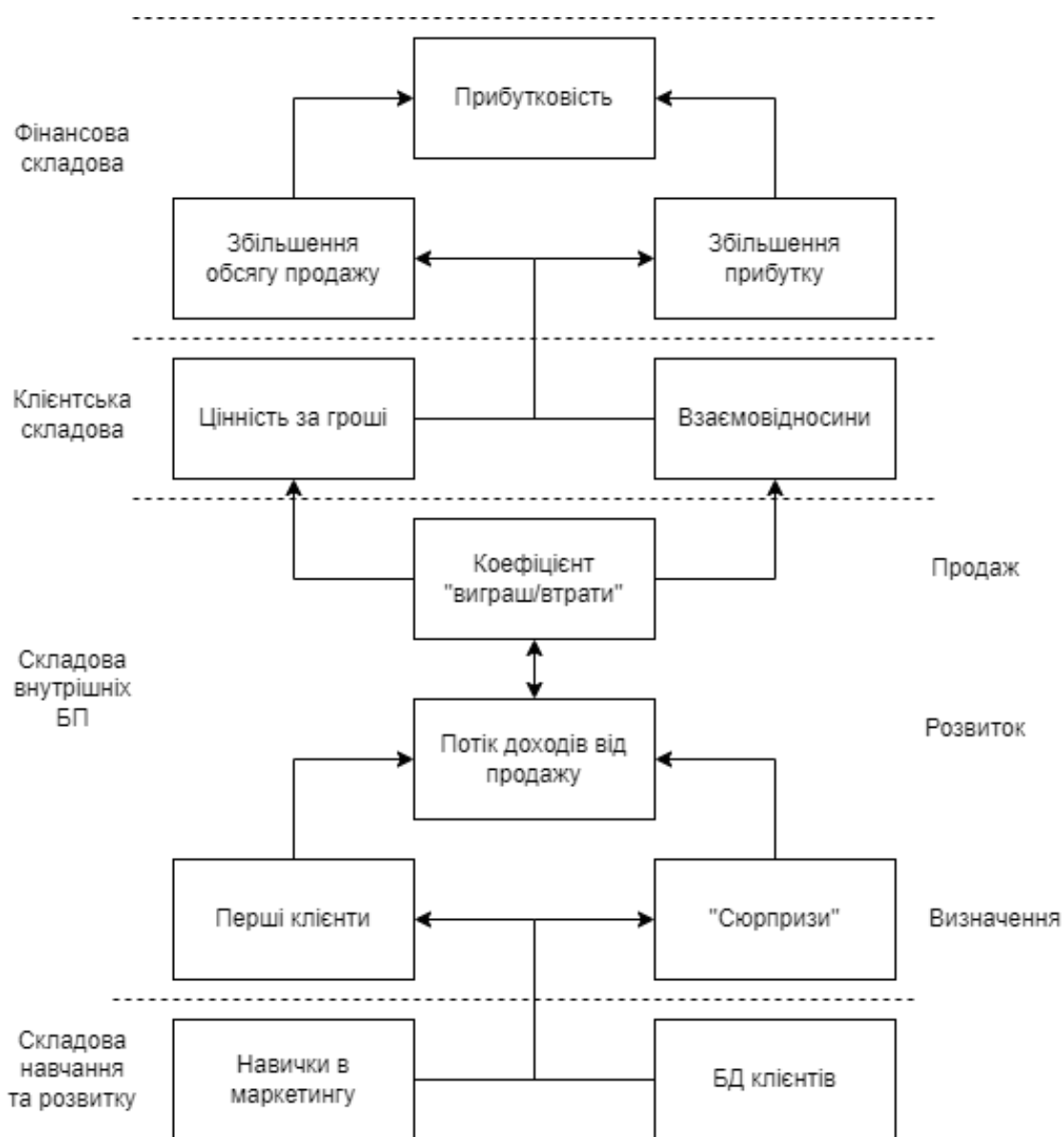


Рисунок 1.1 – Стратегічна мапа діяльності підприємства [6]

Рівень 1. Проекція фінансів компанії. Будується на основі аналізу мети власника щодо підвищення ринкової вартості підприємства.

Рівень 2. Проекція клієнтів. Показує як організація прагне виглядати в очах клієнтів, тобто - це конкурентна пропозиція компанії.

Рівень 3. Проекція внутрішніх БП. Визначає ключові внутрішні процеси, у яких організація має перевершити конкурентів. Проекцію внутрішніх процесів не варто базувати на існуючих у компанії структурних одиницях (відділ маркетингу, фінансовий департамент чи відділ дистрибуції), швидше

вона повинна визначати, як організувати взаємодію різних підрозділів, щоб реалізувати стратегію.

Рівень 4. Проекція навчання та розвитку. Передбачає аналіз потреб ринку, підвищення кваліфікації персоналу та ведення БД клієнтів.

Причинно-наслідкові співвідношення на стратегічній мапі дають організаціям можливість ефективно забезпечувати підтримку виконання стратегії на всіх рівнях організації. Мапа чітко виділяє основні напрямки, в яких слід діяти для досягнення головної мети, зазначеної власником, а також виділити основні елементи аналізу, зокрема асортимент товарів та послуг, що пропонуються компанією. Дуже важливим є розуміння принципів розподілу зон відповідальності для підприємства. Існує два основні варіанти розподілу – функціональний та процесний. Моделювання БП у першому випадку має здійснюватися з максимальною ретельністю у випадках, коли процеси переходять із відповідальності одного підрозділу до відповідальності іншого. При процесній організації більшість БП залишаються в рамках одного підрозділу та, крім того, мають більш логічну структуру [6].

Використання АСУП підвищує ефективність бізнесу. Особливо помітний результат автоматизація дає там, де регламентовано усі БП. Наявність чіткої системи БП дозволить підійти до впровадження АСУП з точки зору максимальної продуктивності.

1.2 Оптимізація бізнес-процесів

БП, що відбуваються в рамках організації, можна поділити на дві великі групи: виробничі та адміністративні. Оптимізація виробничих БП спрямована на вдосконалення основних процесів, що пов'язані з виробництвом товарів та наданням послуг. Оптимізація адміністративних БП спрямована на вдосконалення всіх допоміжних процесів, наприклад, розробка нового продукту, обробка замовлень тощо [7]-[9].

Концепція оптимізації БП ґрунтується на чотирьох підходах (рис. 1.2).

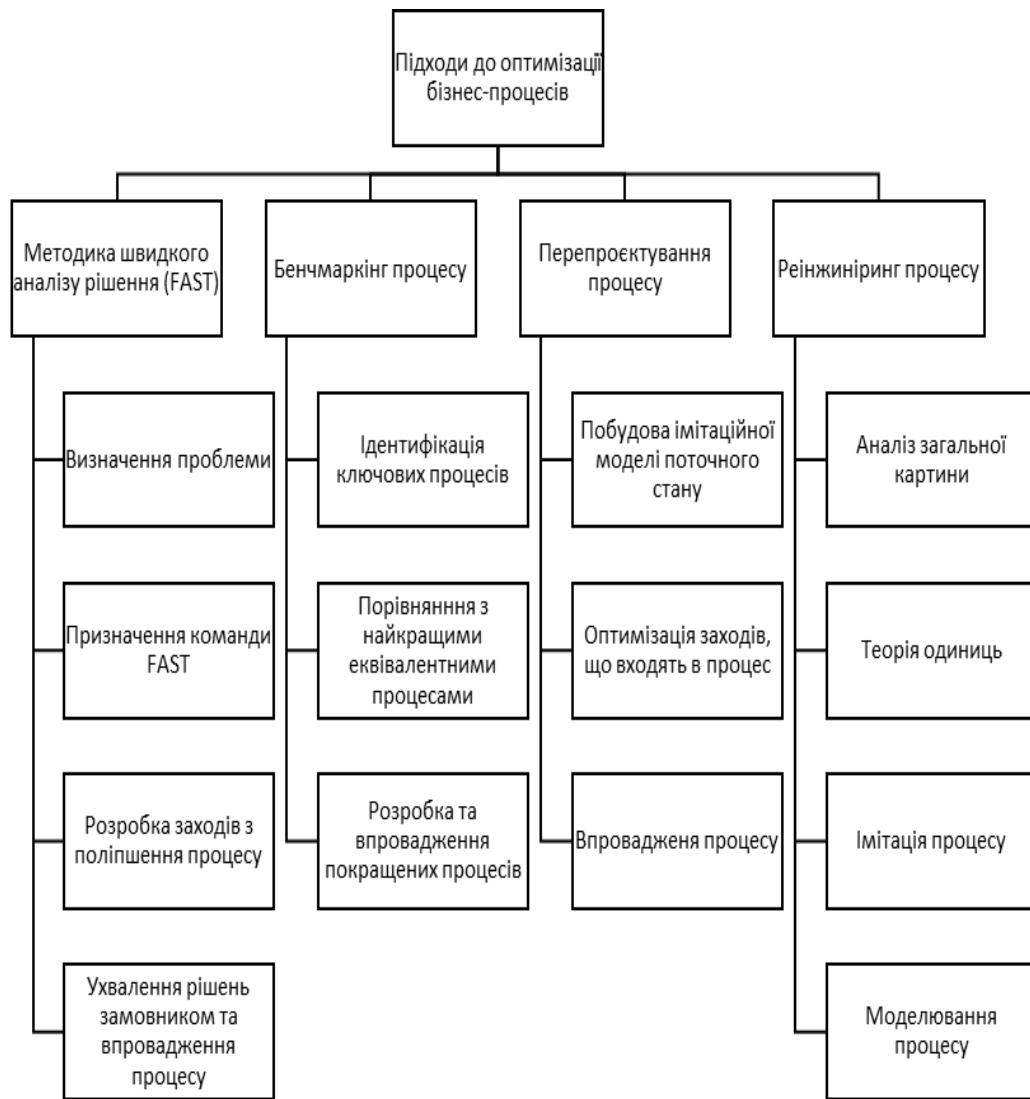


Рисунок 1.2 – Підходи до оптимізації БП [10]

Підходи до оптимізації БП:

- а) методика швидкого аналізу рішення (FAST);
- б) бенчмаркінг процесу;
- в) перепроєктування процесу;
- г) реінжиніринг процесу [10].

Методика швидкого аналізу рішення може застосовуватися до заходів будь-якого рівня. Типовими покращеннями при застосуванні FAST-підходу є зниження витрат, тривалості циклу та рівня помилок. Виявлення можливостей для покращення та схвалення їх впровадження здійснюється за декілька днів, тому даний підхід і отримав свою назву FAST [10].

Бенчмаркінг БП передбачає, що ключові процеси ідентифікуються, усвідомлюються і порівнюються з кращими еквівалентними процесами для визначення небажаних розбіжностей. Команда, що проводить бенчмаркінг, використовує отриману інформацію для розробки та впровадження покращених процесів, що поєднують у собі кращі риси процесів «еталонних» організацій, і при цьому часто створюючи процеси, які виявляються кращими, ніж будь-який із вивчених ними раніше [10].

Підхід до перепроєктування процесу зосереджує зусилля команди на вдосконаленні існуючого процесу. При цьому будується імітаційна модель поточного стану. Відповідно до концепції перепроєктування, команда не створює нових шляхів застосування інформаційної технології (ІТ), але використовує переваги вже випробуваних методик. Часто порівняльний аналіз процесів проводиться паралельно з перепроєктуванням заходів, щоб гарантувати, що перепроєктований процес буде не гіршим або кращим за відповідний зразок [10].

Реінжиніринг процесу – найбільш радикальний з усіх чотирьох підходів до покращення БП, оскільки його успіх ґрунтується на інноваціях та творчих здібностях команди щодо покращення процесу. Цей підхід забезпечує максимальні поліпшення, проте залишається найбільш дорогим, тривалим з усіх підходів до поліпшення БП і з ним пов'язана також найбільша ступінь ризику [10]. Одним з завдань реінжинірингу є застосування методу теорії одиниць (рис. 1.3). Основними блоками алгоритму даного методу є наступні.

Блок 1. Початок – визначення необхідних дій у рамках процесу.

Блок 2. Мінімальний одиничний ресурс – встановлення мінімальної кількості елементів, яку хочемо оптимізувати.

Блок 3. Виключення парадигм – відкидаються парадигми оптимізації тривалості циклу.

Блок 4. Інструмент ІТ – обираються необхідні інструменти ІТ для БП.

Блок 5. Інструмент процесу – обираються інструменти оптимізації процесу.



Рисунок 1.3 – Схема алгоритму методу «Теорія одиниць»

Блок 6. Інструмент персоналу – підбирається потрібний персонал.

Блок 7. Організаційні інструменти – вибираються організаційні інструменти для БП.

Блок 8. Відповідає уявленню про процес – перевіряється одержаний процес. Якщо отримуємо прийнятну відповідь, то рухаємося далі і переходимо до блоку 10. Якщо ні, то переходимо до блоку 9 і цикл триває доти, доки не буде досягнута мета.

Блок 9. Ресурси плюс один – додавання ресурсу задля досягнення мети.

Блок 10 Початок заходів щодо моделювання – підготовка процесу до моделювання.

Блок 11. Кінець. Перехід до моделювання процесу [10].

1.3 Аналіз особливостей технології Workflow

Для ефективної реалізації БП розроблено методи та інструментальні засоби опису, проектування, аналізу та оцінки БП, концепції та правила їх реорганізації, а також ІТ підтримки. Технологія Workflow є однією з популярних технологій управління БП [11].

Продукти класу Workflow – це програмні системи, що забезпечують повну чи часткову координацію виконання виробничих операцій (завдань, робіт, функцій), що становлять структуровані БП підприємства.

Відношення між базовими поняттями технології Workflow відображені на концептуальній інформаційній моделі, яка представлена на рисунку 1.4.

Ця концептуальна модель та/або її різні модифікації є основою програмної реалізації будь-якої системи класу Workflow, що забезпечує рішення трьох наступних завдань:

- розробка опису БП;
- управління виконанням БП;
- інтеграція використовуваних у процесі застосунків [11].

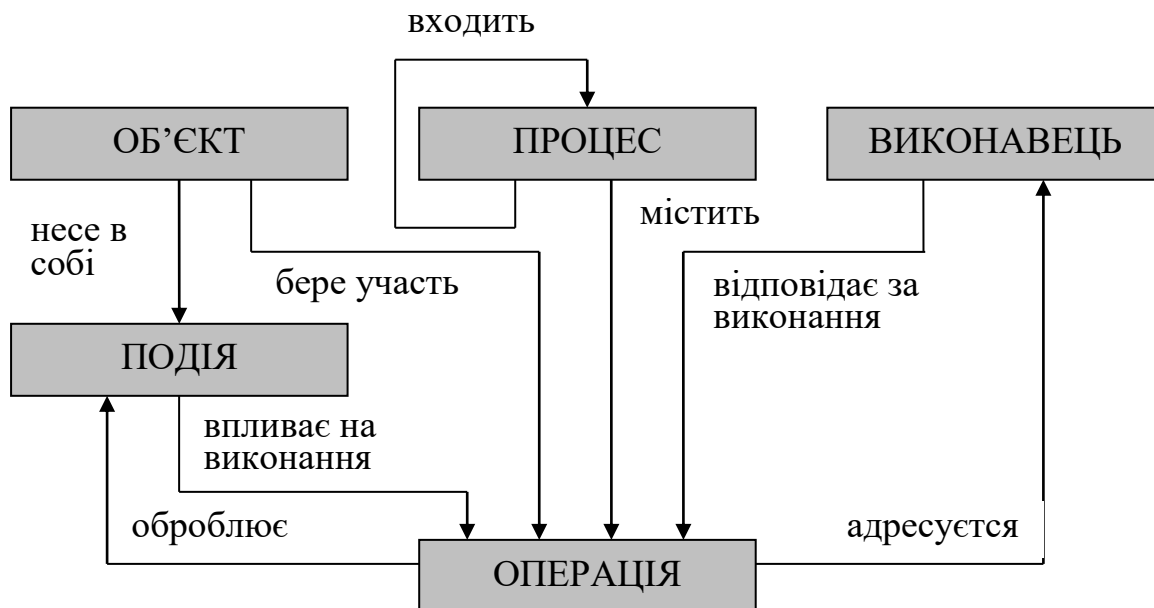


Рисунок 1.4 – Концептуальна інформаційна модель технології Workflow [11]

Відповідно до цих завдань у складі системи можна виділити типові компоненти (рис. 1.5) і проаналізувати зв'язки між ними.

Інструментальні засоби опису процесу призначені для формування формального опису процесу у вигляді впорядкованої множини операцій, правил їх виконання, пов'язаних з ними об'єктів, виконавців та подій. Отриманий опис або специфікація процесу використовується для контролю та управління виконанням процесу на основі даних, що надходять. Як ці дані виступають введена користувачем інформація, а також результати виконання окремих операцій, дані від прикладних систем, архівів і баз.

У свою чергу засоби управління виконанням процесу надають користувачеві та/або прикладній системі інформацію, необхідну для кожної операції, а виявлені на етапі виконання невідповідності є підставою для перегляду специфікації процесу [11].

Найважливішою особливістю технології Workflow є підтримка управління процесами, що містять як автоматизовані – виконувані засобами інформаційних систем (ІС), так і неавтоматизовані – виконувані вручну операції.

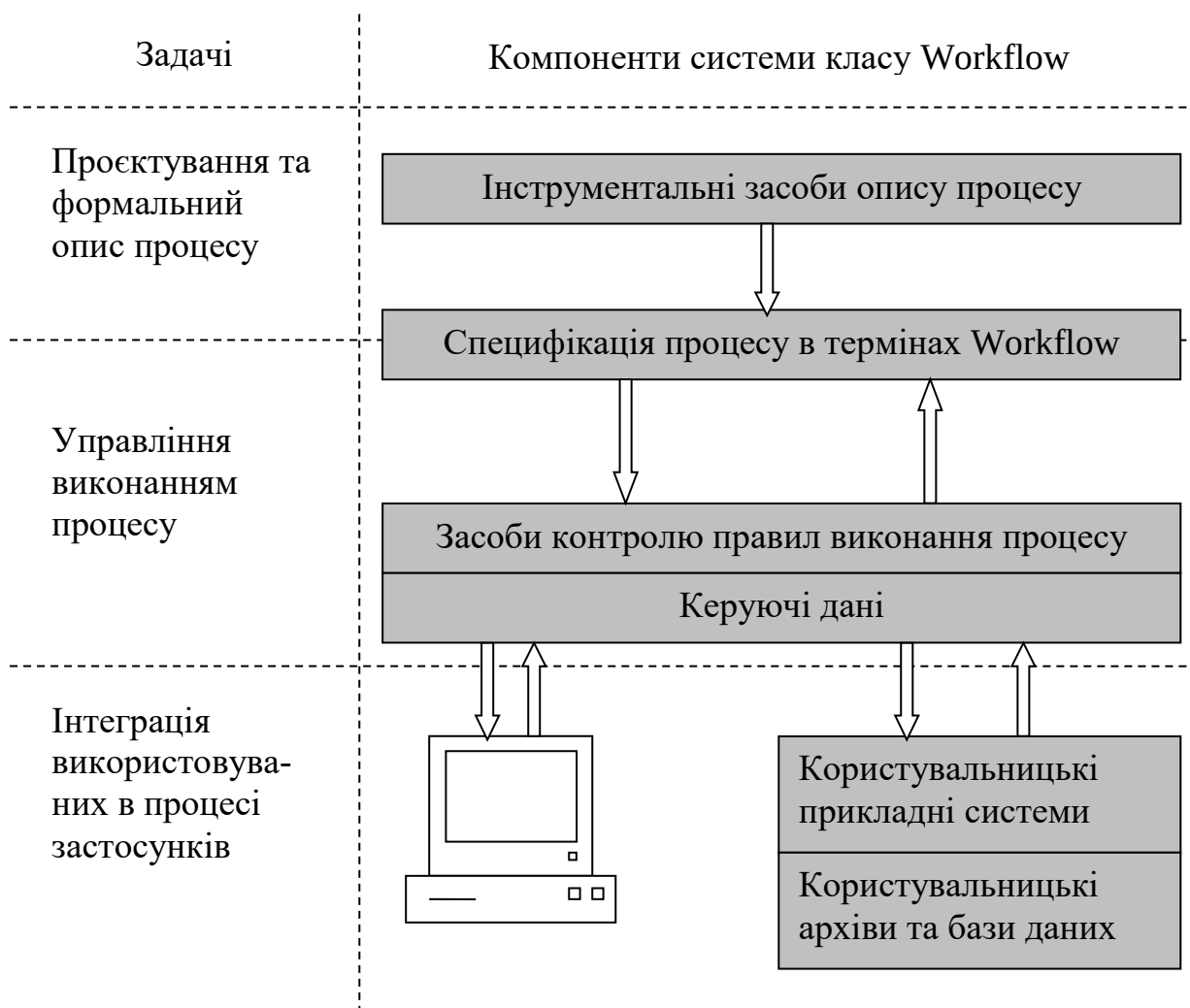


Рисунок 1.5 – Задачі та компоненти системи класу Workflow [11]

Завдяки цій особливості будь-який БП підприємства може бути представлений у вигляді процесу Workflow, якщо, звичайно, цей процес:

- виділено;
- структуровано;
- виконується за правилами, які можна сформулювати;
- періодично повторюється.

З точки зору організації бізнес-системи основним завданням технології Workflow є відокремлення правил виконання БП від прикладних систем та систем управління базами даних, що забезпечує принципово більшу гнучкість та адаптованість ІС. Іншими словами, технологія Workflow надає можливість

оперативної модифікації правил виконання БП без перебудови прикладного програмного забезпечення (ПЗ) та зміни структури корпоративної бази даних.

Іншим важливим напрямом використання технології Workflow є інтеграція різних застосунків та даних навколо БП. У цьому плані Workflow можна як розглядати як певний крок у розвитку архітектури відкритих систем [11].

1.4 Постановка задач магістерської роботи

Як показали проведені дослідження, задача проєктування та застосування АСУП базується на створенні системи чітко регламентованих та оптимізованих БП. Для ефективної реалізації БП розроблено різноманітні методи та інструментальні засоби опису, проєктування, аналізу та оцінки БП, концепції та правила їх реорганізації, а також інформаційні технології підтримки.

Метою роботи є дослідження та програмна реалізація підсистеми автоматизації для моделювання та моніторингу бізнес-процесів в середовищі ВРМ-системи.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати та обрати ВРМ-систему для управління бізнес-процесами;
- розробити опис та діаграму бізнес-процесу «Оформлення online-замовлення»;
- розробити структуру підсистеми автоматизації бізнес-процесів;
- програмно реалізувати модулі моделювання та моніторингу бізнес-процесу;
- виконати розробку методики використання розробленої підсистеми.

2 АНАЛІЗ ТА ВИБІР СИСТЕМИ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ

2.1 Особливості та переваги BPM систем

Традиційний спосіб автоматизації БП – розробка або придбання готового прикладного ПЗ. Однак на практиці прикладні програми автоматизують лише частину кроків БП, а головне – внесення навіть невеликих змін в схему БП означає необхідність перепрограмування та значні витрати часу. В результаті прикладні програми не встигають оновлюватися в тому темпі, який диктують змінювані умови бізнесу та потреби підприємства. Тому, BPM-системи з'явилися як рішення саме цієї конкретної проблеми.

BPM система (BPMS) — це інструмент для реалізації методології управління для вдосконалення БП організації шляхом ідентифікації, моделювання, автоматизації, аналізу та вимірювання ефективності.

Ключовою перевагою BPMS є те, що її користувачі мають можливість брати активну участь у вдосконаленні БП за допомогою простих та інтуїтивно зрозумілих інструментів. Усі ці переваги можливі завдяки трьом компонентам: засобам побудови процесів, їх виконання (платформа) та моніторингу (бібліотека процесів) [12].

BPMS допомагає компаніям розробляти, моделювати, виконувати, автоматизувати та покращувати набір дій і завдань, які після виконання досягають організаційної мети. У сукупності ці дії та завдання, які можуть виконуватися людьми та обладнанням і зазвичай охоплюють бізнес-відділи, складають те, що називається БП. BPMS — це технологічний продукт, який підтримує управління БП, дисципліну, спрямовану на вдосконалення БП [13].

BPMS, яку іноді називають пакетом керування БП, — це набір технологій, які включають наступне:

- інструменти інтелектуального аналізу процесів для виявлення, представлення та аналізу завдань, які керують БП;
- нотація моделювання БП (Business Process Modeling Notation, BPMN) для діаграм БП;

- механізми робочого процесу для автоматизації потоку завдань, які завершують БП;
- механізми бізнес-правил, які дозволяють кінцевим користувачам змінювати бізнес-правила, не звертаючись за допомогою до програміста;
- інструменти моделювання та тестування для спостереження за тим, як поведуться процеси, без попереднього кодування [13].

Успішне впровадження BPMS допомагає підприємствам досягти наступного:

- прискорити час виходу на ринок;
- покращити показники вартості, продуктивності та якості;
- покращити рівень обслуговування та задоволеності клієнтів;
- спростити БП;
- керувати ризиками та відповідати нормативним вимогам;
- швидше запроваджувати нові конструкції процесів;
- скоротити витрати і збільшити доходи [13].

Інструменти BPMS формують робоче середовище, яке дозволяє досягти цілей та переваг BPM. Ці інструменти допомагають компаніям реалізовувати свої бізнес-стратегії, координуючи зміни та вдосконалення БП між відділами та навіть із зовнішніми партнерами. Вони забезпечують інтеграцію корпоративних систем і корпоративних програм, таких як планування ресурсів підприємства (Enterprise Resource Planning System, ERP) і управління взаємовідносинами з клієнтами (Customer Relationship Management, CRM).

Інструменти BPMS збирають дані та проводять аналітику, щоб допомогти компаніям визначити, які частини БП можна автоматизувати або змінити [13].

При правильному розгортанні інструменти BPMS також підвищують продуктивність і моральний дух співробітників, автоматизуючи ручні завдання, усуваючи вузькі місця процесу та надаючи їм кращу видимість того, як їх робота вписується в більший процес.

BPMS вважається однією з технологій, яка допомагає компаніям реалізувати свої ініціативи з цифрової трансформації, надаючи інструменти, які керують оптимізацією існуючих БП і створенням нових [13].

Стандартні функції пакетів BPMS включають наступне:

- механізм для моделювання процесів і вебзастосунків;
- збір даних і бізнес-аналітика для забезпечення розумних змін процесів;
- система управління контентом (Content Management system, CMS) для безпечного зберігання файлів;
- інструменти для співпраці, включаючи програмне забезпечення для соціальної співпраці;
- хмарне або локальне розгортання [13].

Набори функцій продуктів BPMS швидко розвиваються, оскільки постачальники намагаються реагувати на потреби своїх клієнтів, швидко автоматизувати як складні, так і прості процеси, вносити зміни в режимі реального часу та використовувати новітні технології штучного інтелекту для отримання конкурентної переваги [13].

2.2 Огляд структури BPMS

Типова BPMS складається із стандартного набору компонентів, відповідно до стадій життєвого циклу (ЖЦ) БП: проєктування, виконання, моніторинг [14]. Узагальнену структурну схему BPMS наведено на рисунку 2.1.

Проєктування - це розробка схеми БП бізнес-аналітиком, в процесі чого він малює кроки, описує бізнес-логіку, визначає групи користувачів і перелік реквізитів, що вводяться на кожному кроці. Для цього до складу BPM-системи входять такі інструменти: графічний дизайнер для створення схеми БП та репозиторій для її зберігання та організації спільного доступу [14].

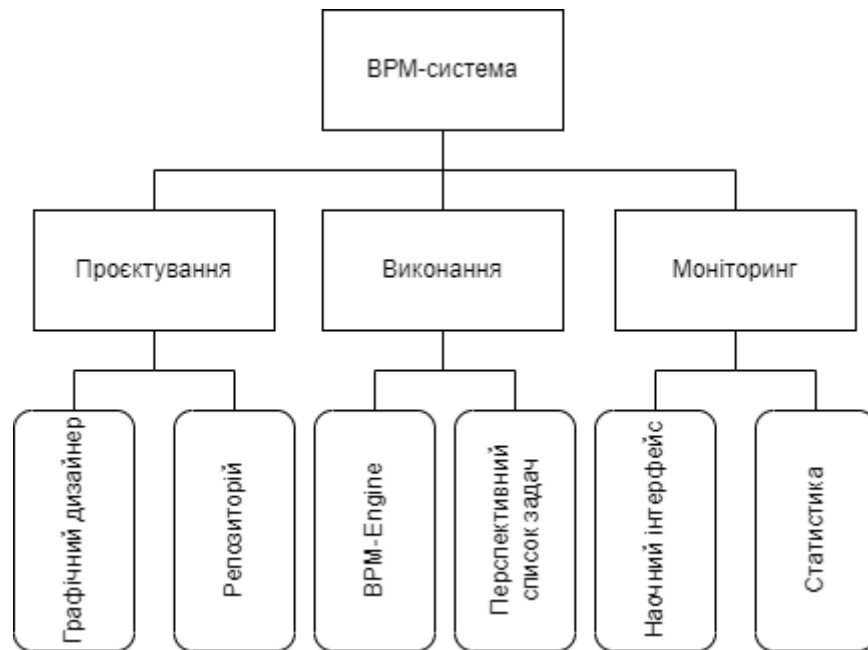


Рисунок 2.1 – Узагальнена структурна схема BPMS [14]

Результат зберігається на сервері, після чого процес може бути ініційований. При необхідності в схему можна вносити зміни. Альтернативно, схема БП може розроблятися в якомусь із традиційних засобів моделювання БП та переноситися до BPM-системи за допомогою імпорту-експорту [14].

Ядром BPM-системи виступає BPM Engine, який виконує бізнес-правила, стартує екземпляри БП, відстежує зміну їхніх станів, зберігає значення реквізитів. Готова схема БП у вигляді XML-файлу завантажується в «двигун», а потім починають стартувати екземпляри БП. «Двигун» зберігає інформацію про екземпляри БП: ким і коли він запущений, на якому кроці зараз знаходиться і хто відповідає за його виконання. БП може запускатися вручну або програмно [14].

Ядро BPM-систем надає також інтерфейси для інтеграції із зовнішніми програмами – спеціальні адаптери, web-сервіси, драйвери для доступу до реляційних баз даних або інших джерел даних. Використання цих інтерфейсів залежить від типу БП.

Тип перший - БП, який BPM-система може виконати повністю автоматично, шляхом запуску спеціалізованих програм. Тип другий – БП, який

передбачає як стикування зі спеціалізованими застосунками, так і участь живих людей. Тип третій – БП, кроки яких автоматизувати неможливо чи надто складно. У такій ситуації BPM-система надішле користувачеві повідомлення про те, що йому доручено певне завдання, і чекатиме від нього підтвердження виконання [14].

Персональний список завдань - це перелік кроків запущених екземплярів БП, призначених даному конкретному користувачеві або групі користувачів, до якої він належить. Завдяки такій організації роботи виконавець бачить перелік призначених йому завдань, і коли він бере чергове завдання собі на виконання, потрібна програма запускається автоматично [14].

BPM-система виконує моніторинг БП двома способами: на основі динамічно формованого графічного зображення та на основі накопичення статистичних даних про параметри виконання екземплярів БП:

- інтенсивність - кількість екземплярів на тиждень чи місяць;
- тривалість - час від запуску до завершення;
- навантаження на окремих фахівців - кількість та тривалість виконаних завдань [14].

Дизайнер, «двигун» та монітор є стандартними компонентами BPM-системи та забезпечують основні задачі: розробка схеми БП, виконання та аналіз. Розвинені BPM-системи можуть включати додаткові модулі.

2.3 Порівняльний аналіз та вибір BPM-системи

Системи від провідних постачальників підтримують повний ЖЦ БП (моделювання, виконання, моніторинг) і дозволяють як інтегрувати зовнішні системи та програми, так і автоматизувати діяльності людей-учасників процесу.

В роботі виконано аналіз популярних BPM-систем: ARIS Express, JBoss jBPM, RUNA WFE, BizAgi, Intalio|BPMS, Camunda, IBM WebSphere Business Modeler, Fox Manager BPA.

ARIS Express — це безкоштовний інструмент моделювання для аналізу та управління БП. Він підтримує різні нотації моделювання, такі як BPMN 2, керовані подіями ланцюги процесів, організаційні діаграми, ландшафти процесів, дошки тощо. Інструмент надається як безкоштовна програма на вебсторінці спільноти ARIS. ARIS Express належить до сімейства засобів моделювання ARIS (ARchitecture of Integrated Information Systems) і включає не тільки інструменти моделювання БП та публікації моделей, але і засоби розробки системи збалансованих показників, що інтегруються між собою, оцінки та оптимізації вартості БП, інструменти, що спрощують впровадження ERP-систем, і навіть інструменти контролю над виконанням БП [15].

jBPM — це набір інструментів для створення бізнес-застосунків для автоматизації БП і рішень. jBPM надає різноманітні можливості, які спрощують і перетворюють бізнес-логіку на активи для повторного використання, такі як кейси, процеси, таблиці рішень тощо. jBPM можна використовувати як окремий сервіс або вбудований в сервіс користувача. Він не вимагає використання будь-яких фреймворків, його можна успішно використовувати в різних програмах. jBPM зазвичай використовується для створення бізнес-застосунків. Бізнес-програму можна визначити як спеціальне рішення, яке розв'язує конкретну бізнес-проблему. Для реалізації бізнес-логіки він використовує можливості з різних інфраструктур, таких як БП, бізнес-правила та обмеження планування, а також стійкість, обмін повідомленнями, транзакції тощо [16].

RunaWFE — це безкоштовна BPMS з відкритим кодом, яка застосовується для оптимізації та автоматизації БП (для Windows, Linux, Mac OS). Це проста в установці та використанні, повнофункціональна платформа бізнес-застосунків. Вона включає в себе різні інструменти для роботи з даними (введення/відображення/виведення), сповіщень клієнтів про завдання та інші компоненти, які забезпечують зручність для кінцевого користувача. Виконавцями в системі можуть бути як люди, так і спеціальні комп'ютерні програми - боти, які знімають з людей багато рутинних завдань. Завдяки безкоштовному вебінтерфейсу RunaWFE, користувачі можуть отримувати,

фільтрувати, виконувати та делегувати завдання екземплярів БП, запускати нові екземпляри БП, переглядати властивості запущених екземплярів БП, розгортати нові визначення БП: створювати/вилучати акторів і групи, включати/виключати акторів до/з груп, надавати дозволи на системні об'єкти акторам і групам, примусово зупиняти екземпляри БП, додавати/змінювати правила заміни акторів. RunaWFE Free Workflow System — це програмне забезпечення для повної організації робочого процесу [17].

BizAgi — це BPM-система, спрямована на моделювання, виконання, автоматизацію та аналіз БП. Мобільний застосунок BizAgi дозволяє виконувати та контролювати автоматизовані БП з мобільного телефону. Система Bizagi включає 3 модуля для повноцінного налаштування процесів:

- Modeler — повнофункціональна середовище моделювання процесів у нотації BPMN (Business Process Model and Notation), відмінний інструмент побудови БП, що допомагає не лише створити, а й публікувати результати роботи у різних форматах, включаючи MS Word та інтерактивний HTML;

- Studio — середовище розробки БП;

- Engine — середовище виконання процесів, яке доступне користувачам у будь-якому браузері з будь-якого пристрою [18].

Intalio | BPMS – безкоштовна відкрита система для побудови та аналізу БП, що має найбільшу кількість установок у всьому світі. Система поєднує в собі безкоштовність і відкритість з можливістю придбання підтримки високого рівня, що є важливим фактором при виборі корпоративного ПЗ. Користувачами цієї програми управління БП є такі компанії, як BNP PARIBAS, Deutsche Bank, Santander та багато інших. Компанія Citeck є офіційним партнером компанії Intalio і надає послуги з впровадження та підтримки системи Intalio | BPMS. Інноваційна програма управління БП Intalio | BPMS дозволяє організувати ефективну взаємодію між бізнес-користувачами та ІТ-спеціалістами, поліпшити використання існуючих і підвищити ефективність розробки нових ІС. Intalio BPMS надає можливість моделювання, тестування, моделювання, виконання, контролю та управління БП [19].

Camunda — це безкоштовна та відкрита платформа для автоматизації робочого процесу та прийняття рішень. Платформа Camunda розроблена на Java, поставляється з інструментами для створення робочих процесів і моделей рішень, керування розгорнутими моделями у виробництві та надання користувачам можливості виконувати призначені їм завдання робочого процесу. Desktop Modeler ліцензується відповідно до ліцензії MIT, а інші компоненти — відповідно до пропріетарних ліцензій. Платформа забезпечує механізм робочого процесу, сумісний зі стандартом моделі БП та нотації BPMN, а також механізм прийняття рішень, сумісний зі стандартом моделі прийняття рішень та нотації Decision Model and Notation (DMN), який можна вбудовувати в програми Java та інші мови через Representational state transfer (REST) [20].

IBM WebSphere Business Modeler є програмним засобом, орієнтованим на моделювання, імітацію та аналіз БП. Ключові характеристики IBM WebSphere Business Modeler такі:

- дозволяє сформувати перелік показників, прив'язати їх до елементів БП та шляхом імітації моделі спрогнозувати їх значення. Таким чином, відстежується досягнення стратегічних та тактичних цілей компанії;
- дозволяє описувати БП за допомогою діаграм стандарту BPMN. Інформація про організацію може накопичуватися у вигляді структурованих довідників, між довідниками можуть встановлюватися взаємозв'язки;
- за допомогою інструментарію Crystal Report у системі можуть створюватися будь-які види звітності по об'єктах моделі та регламентної звітності, які можуть бути вивантажені у Word, Excel, PDF та інші формати;
- система підтримує понад 40 видів аналізу як статичного (аналізується структура моделі), так і динамічного (аналізується модель під час та після імітації);
- можливості збору та контролю значень показників дозволяють використовувати систему не тільки як систему проєктування, але і як систему виконання;

- моделі можуть бути опубліковані так, що стануть доступними команді розробників для ознайомлення та аналізу;

- система легко інтегрується з іншими продуктами розробки IBM [21].

Fox Manager BPA – безкоштовна програма для проєктування бізнесу. Нотація, що використовується, близька до Basic Flow Chart, яка багатьом добре знайома своєю простотою побудови БП. Програма автоматично будує процеси верхнього рівня, відображаючи взаємодії категорій та БП у вигляді наочної діаграми. Аналітичні функції програми дозволяють вчасно помітити та усунути помилки, допущені при побудові побудованої процесної моделі, а саме виділити процеси, за які ніхто не відповідає, знайти посилання на неіснуючі документи, посади, постачальників чи БП [22].

2.4 Структура робочого середовища

Структурну схему робочого середовища Intalio | BPMS представлено на рисунку 2.2 [23].

Intalio | Designer є інтегрованим середовищем розробки БП. Оскільки дана система пропонує нульовий код розвитку і надає можливість при натисканні однієї кнопки розгорнути ЖЦ продукту, що розробляється, не обов'язково вдаватися до послуг експерта проєктування БП. Intalio | Designer надає зручну систему допомоги та порад, яка допоможе зробити роботу в даному програмному продукті більш продуктивною. Важливою особливістю Intalio Designer є можливість звернення до функціональних процесів графічно. Intalio | Designer є інструментом для моделювання БП за допомогою нотації BPMN або за допомогою мови BPEL (Business Process Execution Language). У загальному вигляді конфігурація BPEL-проєкту виглядає так:

- BPEL-візуальний редактор (Intalio | Designer);
- сервер управління БП (Intalio | Server) [23].

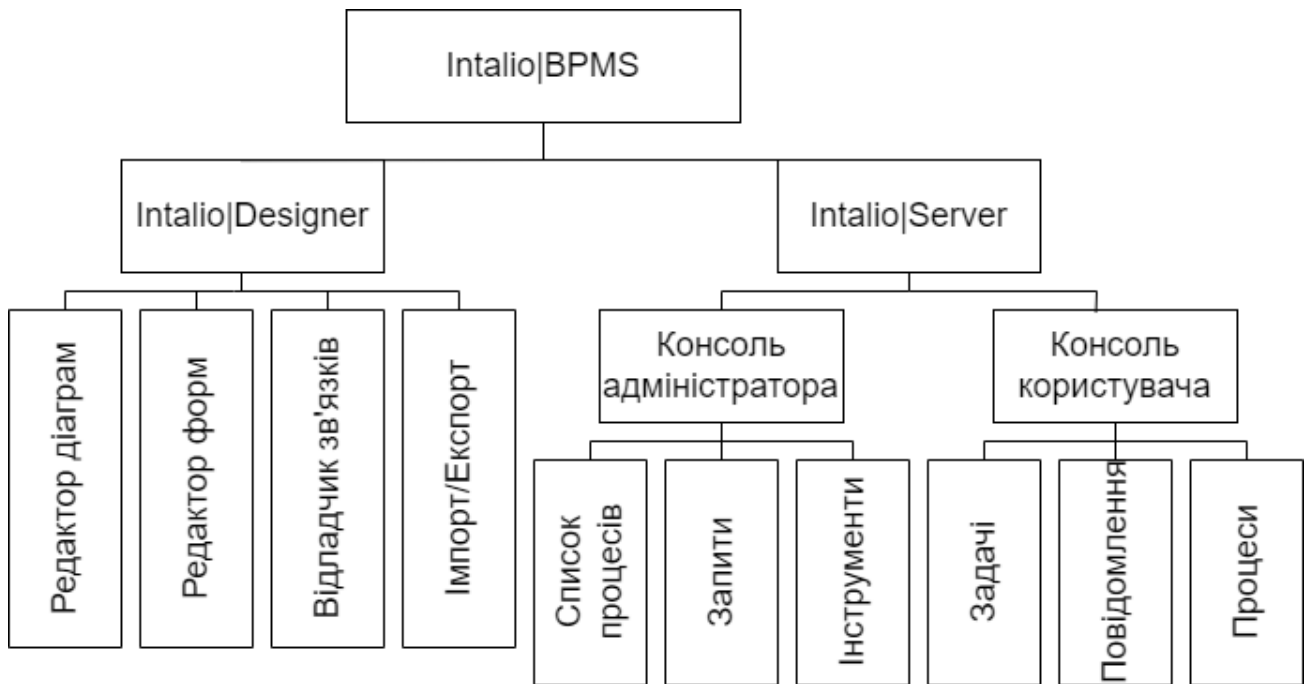


Рисунок 2.2 – Структурна схема Intalio | BPMS [23]

У сукупності вебсервіси та мова BPEL for Web Services дозволяють суттєво підвищити економічну ефективність інтеграції застосунків. З появою засобів на основі BPEL, зокрема Oracle BPEL Process Manager, з'явилася можливість застосування інтеграції в широких масштабах [23].

Web-сервіси у поєднанні з BPEL можуть зіграти таку ж роль щодо інтеграції застосунків, як стандарт SQL доступу до структурованих даних, або HTTP і HTML, що забезпечують доступ до контенту та застосунків. У перспективі вебсервіси дозволять трансформувати Internet у розподілену платформу, забезпечивши просту надійну взаємодію гетерогенних систем.

Intalio | Server структурований за принципом нової технології Java Business Integration (JBI), він може бути розгорнутий практично на будь-якому сервері застосунків J2EE. Intalio | Server є окремою частиною набору інструментів Intalio. Вона може бути використана для виконання процесів, розроблених у Designer. Server може безшовно інтегруватися з Designer, отже процеси змодельовані Designer можуть бути експортовані безпосередньо в Server. Потім Server виконує розроблений процес. Server може бути використаний також без Designer для виконання БП, змодельованих іншими

інструментами. Ця модульність дозволяє більш тонко налаштувати використовувані інструменти, а також дає більше можливостей для інтеграції або випробування сервера в існуючому середовищі, наприклад, в якості заміни або резервування системи [23].

Intalio | Server є високопродуктивним "двигуном" процесу, який може підтримувати найскладніші БП.

Intalio | BPMS Server має такі можливості:

- підтримка середовища виконання Java для виконання сервлетів;
- підтримка специфікації Enterprise JavaBeans;
- наявність високопродуктивних програмних адаптерів для великої кількості поширених серверних систем, що забезпечує скорочення витрат на створення програмного коду, необхідного для з'єднання динамічних вебсторінок з реальними даними бізнес-підрозділів;
- робота в якості серверу застосунків з метою управління сеансами та станами;
- можливість публікації та використання вебслужб, які дозволяють компаніям пов'язувати програми між собою, надавати доступ до бізнес-функцій для ширшого спектру замовників та партнерів, а також ефективніше взаємодіяти з ринками та динамічно створювати нові бізнес-моделі [23].

2.5 Висновки до розділу 2

Дослідження показали, що серед розглянутих BPM-систем оптимальним вибором є система Intalio | BPMS, оскільки вона відповідає всім сучасним вимогам до BPMS та реалізує три основні ідеї:

- підтримка нових галузевих стандартів BPM;
- пропозиція нульового коду розвитку підходу;
- подання розгорнутого ЖЦ продукту, що розробляється при натисканні однієї кнопки.

Таким чином, Intalio | BPMS визначає стан процесів за BPM-методологією та продовжує встановлювати стандарти для управління БП підприємства. Ця система також надає своєчасну підтримку новітніх стандартів у своїй галузі. Ще однією перевагою цієї системи є те, що вона доступна безкоштовно. Тому в рамках даної роботи доцільно використати саме цю систему.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

3.1 Опис бізнес-процесу та структура підсистеми автоматизації

Більшість компаній, враховуючи постійний розвиток ІТ, а також популярність та доступність мережі Інтернет, все частіше веде електронний бізнес. Інтернет можна сміливо назвати його невід'ємним інструментом, він є найсучаснішим торговим каналом, найбільш оптимальним способом організації торгівлі [24].

Інтернет-магазин (ІМ) є електронним ілюстрованим каталогом товарів з можливістю вибору та замовлення з доставкою. Розрахунки за товари в ІМ проводяться за допомогою електронних платіжних коштів – кредитних карток та так званих «віртуальних» грошей. Що дає ІМ продавцю (власнику):

- прямий прибуток через ІМ;
- ефектне подання товарів через електронний каталог;
- 24-годинна торгова точка;
- непряме збільшення продажів поза ІМ;
- розширення ринку збуту.

Чого чекає від ІМ покупець:

- високоякісні товари за низькими цінами;
- великий асортимент;
- швидке виконання замовлення;
- комфортний процес купівлі;
- персональний сервіс;
- можливість запланувати час та місце отримання товару [26].

Ідея автоматизації БП «Оформлення online-замовлення» полягає в тому, що шляхом заповнення мінімальної кількості форм клієнт надає всі необхідні дані (у тому числі адресу доставки та свої банківські реквізити), чим скорочує

час замовлення до декількох хвилин, а адміністратор може здійснювати моніторинг цього процесу в режимі реального часу.

Розроблювана підсистема буде функціонувати в середовищі системи Intalio | BPMS та включатиме два модуля (рис.3.1):

- модуль моделювання БП;
- модуль моніторингу БП.

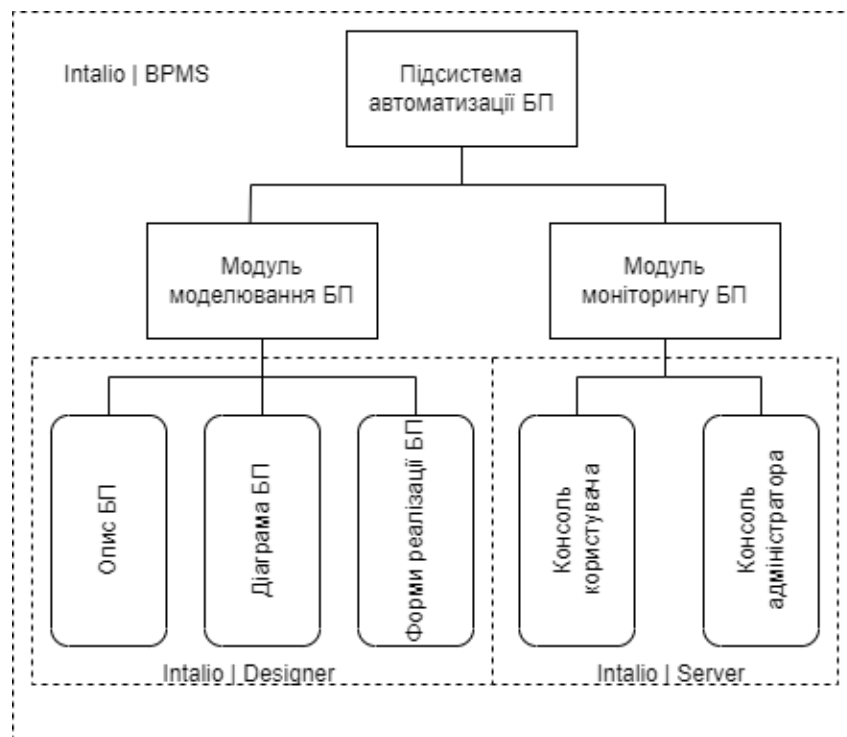


Рисунок 3.1 – Структура підсистеми автоматизації БП

Модуль моделювання БП дозволить створити опис БП, побудувати його діаграму та створити форми реалізації БП. Робота модулю здійснюється шляхом взаємодії з редактором форм Intalio | Designer та екранними формами, що містять відповідні елементи управління та розташовані на панелі Workflow Form Editor Palette. Вхідні дані для роботи модуля містяться у трьох формах:

- форма вибору продукції, яка вимагає від користувача вибрати товар, який він хоче замовити;
- форма адреси доставки, яка вимагає від користувача ввести його адресу, за якою він зможе отримати своє замовлення;

– форма оплати, яка вимагає від користувача ввести дані, про його кредитну картку, якою він розплатуватиметься за придбаний товар.

Вихідні дані - це сформоване замовлення користувача.

Модуль моніторингу БП дозволяє автоматично запустити процес «Оформлення online-замовлення» та контролювати його виконання як користувачем, так і адміністратором системи. Це дозволить скоротити часові витрати та підвищити якість послуг, що надаються. Робота модулю здійснюється в середовищі Intalio | Server за допомогою взаємодії з головним меню та екранними формами, що містять відповідні елементи управління, такі як, кнопки, списки, прапорці, поля введення і т.д.

3.2 Моделювання процесу «Оформлення online-замовлення»

Схему алгоритму моделювання БП наведено на рисунку 3.2.



Рисунок 3.2 – Схема алгоритму моделювання БП

Блок 1. Початок – постановка цілей процесу.

Блок 2. Вибір елементів BPMN-нотації – зіставлення нотації з реальними етапами процесу, що моделюється.

Блок 3. Створення форм – моделювання форм, необхідних для коректного виконання процесу.

Блок 4. Редагування діаграми – наповнення діаграми створеними раніше блоками (дії, потоки, форми і т.д.).

Блок 5. Налаштування зв'язків – налаштування зв'язків між діями.

Блок 6. Кінець – запуск команди Deploy.

3.2.1 Використання BPMN-нотації

Палітру BPMN-нотації в Intalio | Designer показано на рисунку 3.3.

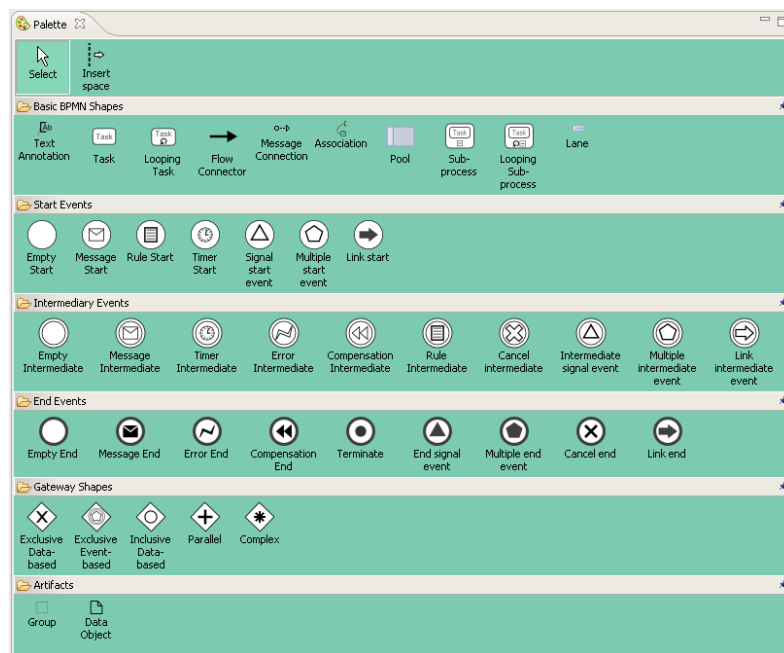
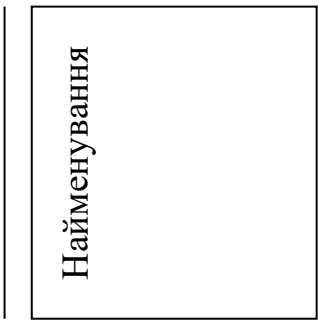


Рисунок 3.3 – Палітра BPMN-нотації

Для моделювання БП «Оформлення online-замовлення» використовуються деякі з елементів BPMN-нотації. Перелік цих елементи наведено у таблиці 3.1.

Таблиця 3.1 – Використані елементи BPMN-нотації

Елемент	Опис	Нотація
Дія	Дія – термін, характерний для позначення роботи, виконуваної компанією. Дія може бути елементарною та неелементарною (складеною). Типи дій, що є частиною моделі процесу: Процес, Підпроцес та Завдання. Завдання та підпроцеси – закруглені прямокутники. Процеси - або безмежні, або містяться в межах області.	
Область	Область представляє собою учасника у процесі. Він також відіграє роль «доріжки» та графічного контейнера для поділу сукупності дій з інших областей, як правило, у контексті ситуацій «бізнес для бізнесу».	
Послідовний потік	Послідовний потік показує порядок, згідно з яким виконуватимуться дії процесу.	
Потік повідомлень	Потік повідомлень символізує потік повідомлень між двома учасниками, готовими до їх надсилання та отримання. У BPMN дві окремі області на діаграмі символізуватимуть двох учасників (наприклад, бізнес-об'єкти або бізнес ролі).	

3.2.2 Створення форм

В Intalio | Designer форми створюються за допомогою панелі Workflow Form Editor Palette (рис. 3.4).

У процесі моделювання було створено три необхідні форми:

- SelectItem.xform - ця форма призначена для того, щоб користувач вибрав необхідний йому товар для замовлення;
- Address.xform – ця форма призначена для введення адреси доставки вибраного товару;
- Payment.xform – ця форма призначена для запиту користувачу надати інформацію про його кредитну картку.

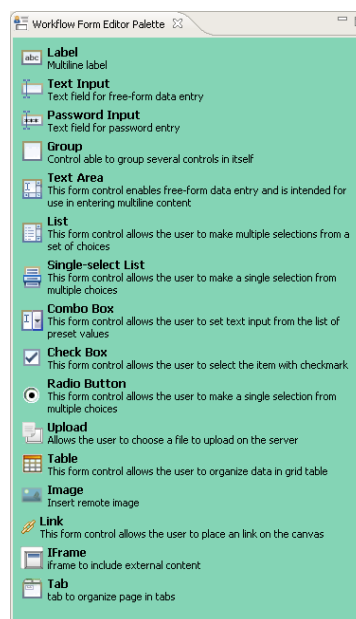
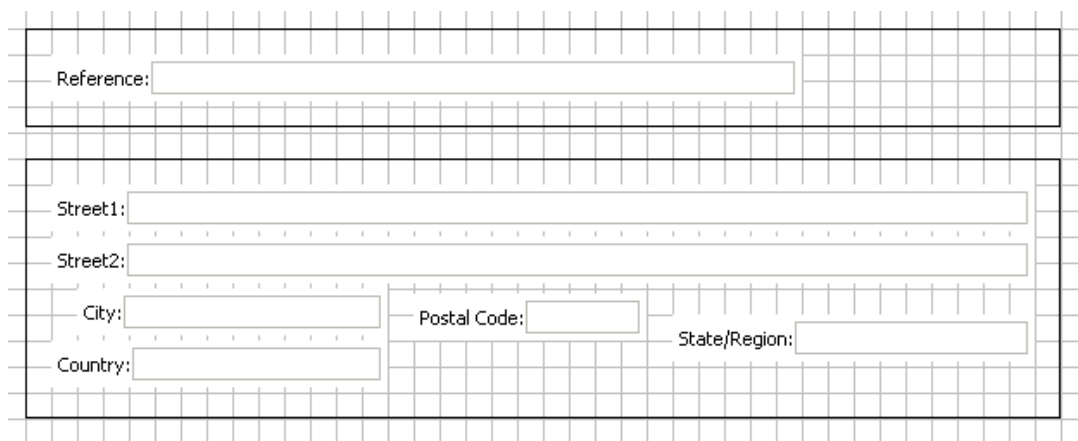


Рисунок 3.4 – Палітра Workflow Form Editor Palette

Інтерфейс створених форм наведено на рис. 3.5 – рис. 3.7.



Рисунок 3.5 – Інтерфейс SelectItem.xform



Reference:

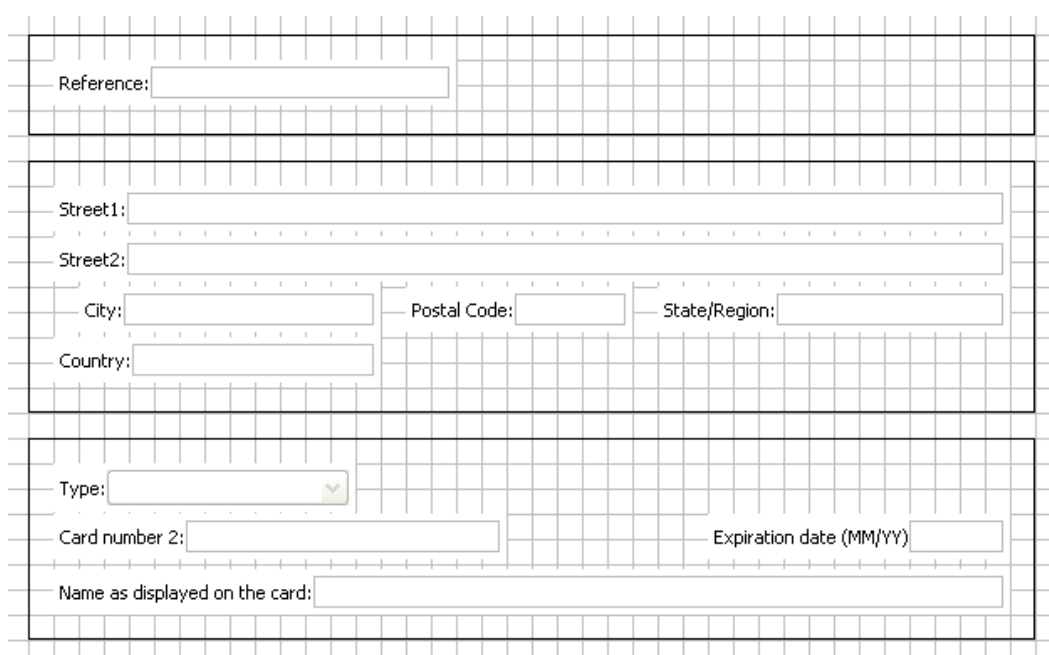
Street1:

Street2:

City: Postal Code: State/Region:

Country:

Рисунок 3.6 – Інтерфейс Address.xform



Reference:

Street1:

Street2:

City: Postal Code: State/Region:

Country:

Type:

Card number 2: Expiration date (MM/YY)

Name as displayed on the card:

Рисунок 3.7 – Інтерфейс Payment.xform

3.2.3 Редагування діаграми бізнес-процесу та налагодження зв'язків

На діаграмі Store.bpm було створено два пула “User” и “Store”. На пулах розміщено дії та створені раніше форми та задано потоки. В результаті було отримано діаграму БП, наведену рисунку 3.8.

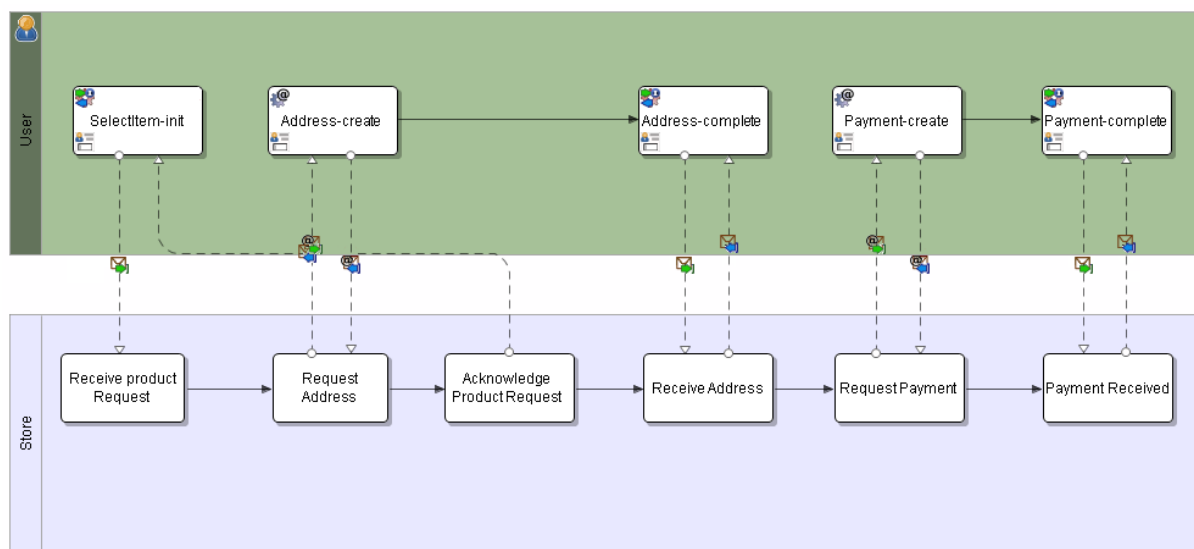


Рисунок 3.8 – Діаграма БП

Заключним етапом моделювання є налагодження зв'язків. На рис. 3.9 – 3.10 показано зв'язки, які встановлюються на діях Request Address та Acknowledge Product Request за допомогою з'єднання потрібних ідентифікаторів.

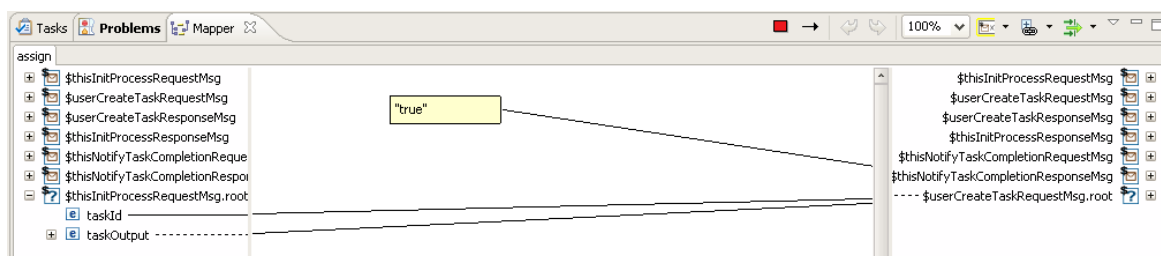


Рисунок 3.9 – Зв'язки дії Request Address

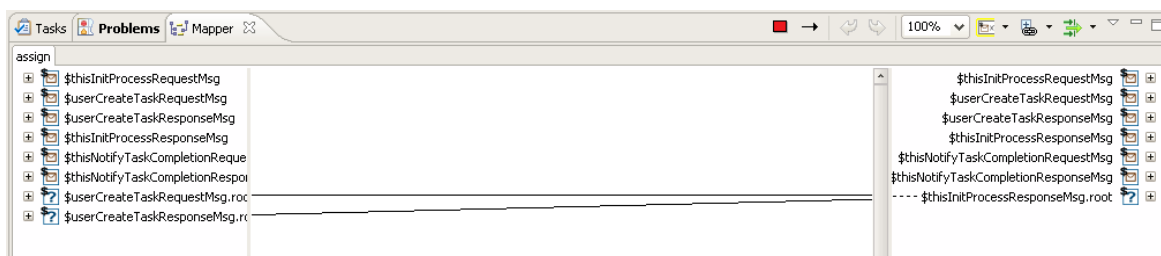


Рисунок 3.10 – Зв'язки дії Acknowledge Product Request

Після завершення процесу моделювання необхідно “розгорнути” процес за допомогою команди “Deploy”. У момент виконання цієї команди результати передаються Intalio | Server для подальшої обробки, виконання та моніторингу.

3.3 Програмне створення форм з використанням мови XPDЛ

У Intalio | Designer форми створюються за допомогою редактора форм з використанням панелі Workflow Form Editor Palette. Але, в рамках цієї роботи форми розроблялися програмно з використанням мови XPDЛ (XML Process Definition Language). Тексти програм наведено в Додатку А.1-А.4.

До характерних особливостей мови XPDЛ можна віднести наступні властивості.

Властивість 1. З позицій опису XPDЛ робочий процес є спрямованим графом, вузлами якого є «дії», пов'язані між собою переходами. Переходи можуть бути умовними, причому умова перевіряється на етапі виконання конкретної «дії».

Властивість 2. У мові XPDЛ існує можливість виділення «блоків», можливість об'єднання «дій» у блок зі своїми окремими умовними чи безумовними точками входу та виходу. Також є можливість визначати вкладені підпроцеси всередині батьківського процесу, які представляють собою повноцінні потоки робіт.

Властивість 3. Специфікація підтримує можливість експорту деяких блоків опису одного процесу в опис іншого з можливістю перевизначення частини імпортованого опису, що унеможливорює дублювання ідентичних фрагментів опису в декількох процесах.

Властивість 4. XPDЛ є розширюваним стандартом. Він дозволяє визначати набір елементів та атрибутів, специфічних для конкретної сфери його застосування.

Властивість 5. Елементи опису процесів XPDЛ мають великий набір атрибутів, що визначають процес виконання процесу. До них можна віднести

умовні вирази для переходів, часові рамки, завдання багатьох виконавців «дій» і т.д. [26]. Результатом розробки є код, який передається до Intalio | Server за допомогою команди розгортання модельованого БП у “Deploy”.

У документі XPDЛ визначається вебсервіс за допомогою основних елементів, зазначених у таблиці 3.2.

Таблиця 3.2 – Основні елементи XPDЛ

Елемент	Що визначає
<portType>	Методи, що надаються вебсервісом
<message>	Повідомлення, що використовуються вебсервісом
<types>	Типи даних, які використовуються вебсервісом
<binding>	Протоколи зв'язку, які використовуються вебсервісом

Основна структура документа XPDЛ виглядає наступним чином.

```

<definitions>
  <types>
    визначення типів.....
  </types>

  <message>
    визначення повідомлення....
  </message>

  <portType>
    визначення порта.....
  </portType>

  <binding>
    визначення зв'язків.....
  </binding>

</definitions>

```

Документ XPDЛ може також містити інші елементи, наприклад, елементи розширення та елемент `service`, який дозволяє об'єднати разом в одному окремому документі XPDЛ визначення декількох вебсервісів.

Елемент `<portType>` є найважливішим елементом у XPDЛ, оскільки він визначає сам вебсервіс, надані їм операції та повідомлення. Цей елемент можна порівняти з бібліотекою функцій (модулем, класом) традиційною мовою програмування.

Елемент `<message>` визначає елементи даних операції. Кожне повідомлення може містити одну або кілька частин. Ці частини можна порівняти з параметрами функцій, що викликаються в традиційній мові програмування.

Елемент `<types>` визначає тип даних, які використовуються вебсервісом. Для максимальної платформи-незалежності XPDЛ використовує синтаксис XML Schema для визначення типів даних.

Елемент `<binding>` визначає формат повідомлення та деталі протоколу для кожного порту.

Інтерфейс розроблених форм наведено на рис. 3.11 – рис. 3.13.

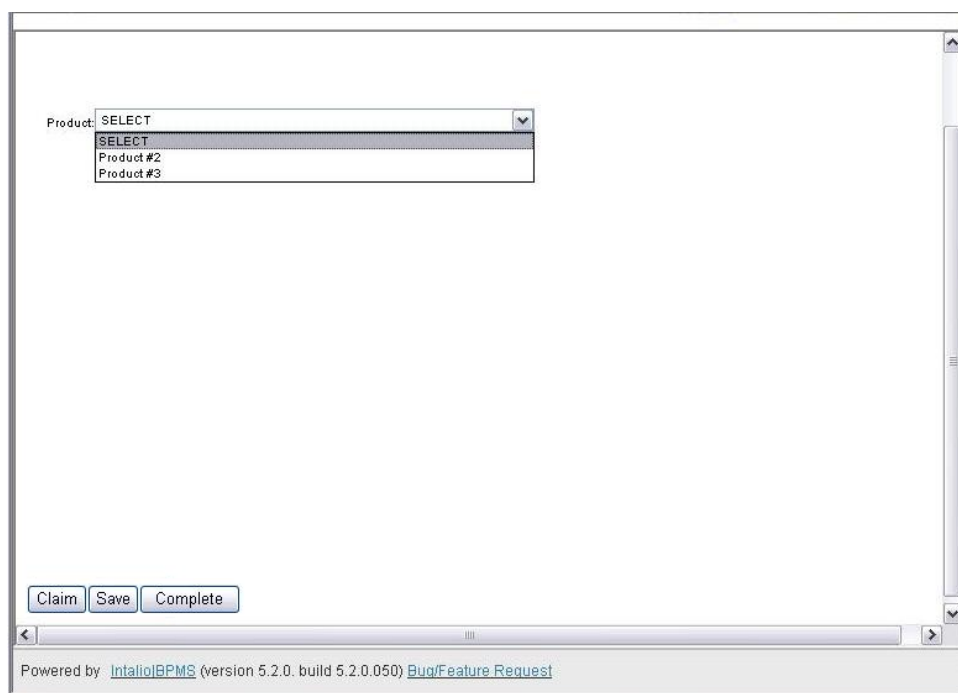


Рисунок 3.11 – Інтерфейс форми «Вибір товару» (SelectItem.xform)

Ordered Items
Reference: P3

Shipping Address
Street1: S-91
Street2: Parkin Street
City: Redwood Postal Code: 160020 State/Region: Green Park
Country: USA

Credit Card
Type: MasterCard
Card number 2: 36147-426574687 Expiration date (MM/YY)
Name as displayed on the card: Charles James

Claim Save Complete

Powered by [Intalio/BPMS](#) (version 5.2.0, build 5.2.0.050) [Bug/Feature Request](#)

Рисунок 3.12 – Інтерфейс форми «Адреса доставки» (Address.xform)

Ordered Items
Reference: P3

Shipping Address
Street1: S-91
Street2: Parkin Street
City: Redwood Postal Code: 160020 State/Region: Green Park
Country: USA

Credit Card
Type: MasterCard
Card number 2: 36147-426574687 Expiration date (MM/YY)
Name as displayed on the card: Charles James

Chained_Execution

Claim Save Complete

Powered by [Intalio/BPMS](#) (version 5.2.0, build 5.2.0.050) [Bug/Feature Request](#)

Рисунок 3.13 – Інтерфейс форми «Оплата» (Payment.xform)

Після завершення моделювання необхідно розгорнути процес за допомогою команди Deploy.

3.4 Моніторинг бізнес-процесу

Після того як Intalio | Designer розгорнув процес, дані передаються в Intalio | Server, де БП виконується відповідно до алгоритму, представленого на рисунку 3.14.

Блок 1. Початок – вхід у середовище Intalio Server.

Блок 2. Вибір консолі. Якщо обирається консоль користувача, то переходимо до Блоку 3, якщо ні, то переходимо до Блоку 8.

Блок 3 Введення паролю користувача – вводиться логін та пароль користувача.

Блок 4. Запуск процесу «Оформлення online-замовлення» – стартує діаграма, яка була передана із системи моделювання Intalio | Designer.

Блок 5. Обробка форми «Вибір товару» – заповнення форми (вибираються необхідні товари для замовлення зі списку).

Блок 6. Обробка форми «Адреса доставки» – вводяться дані про місце, куди необхідно доставити товар.

Блок 7. Обробка форми «Оплата товару» – необхідно вибрати спосіб оплати здійснюваної покупки.

Блок 8. Вибір консолі адміна – вибирається адміністраторська консоль.

Блок 9. Введення пароля адміністратора – вводиться логін та пароль адміністратора.

Блок 10. Моніторинг процесу – переглядаються процеси, їх стан виконання, а також стадії процесу.

Блок 11. Кінець – передача даних про виконане замовлення.

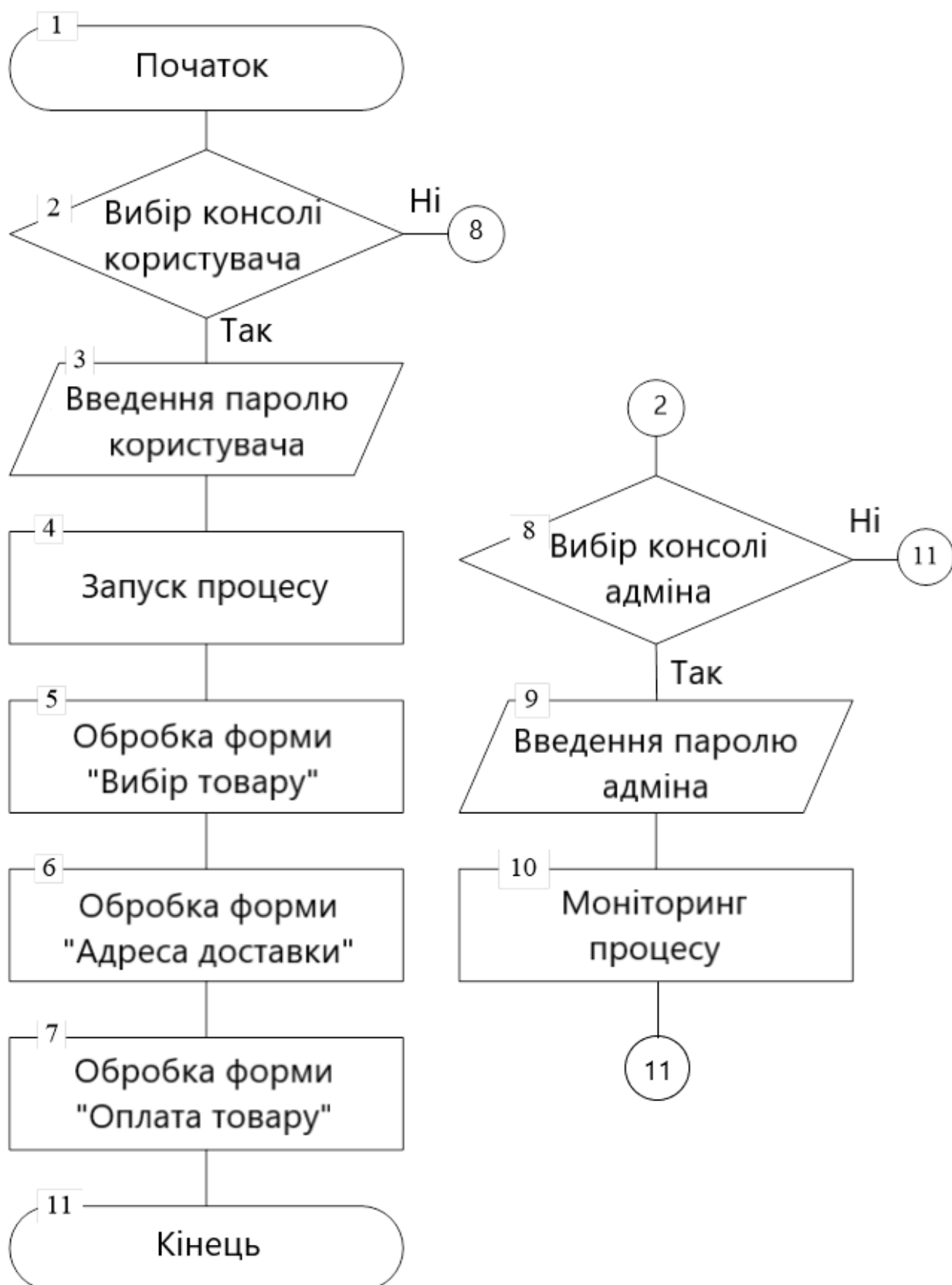


Рисунок 3.14 – Схема алгоритму моніторингу БП

3.4.1 Запуск процесу «Оформлення online-замовлення»

Для того щоб стартувати процес необхідно було написати програму, яка буде автоматично запускати форми для заповнення інформації, перевіряти поля на наявність інформації, а також автоматично переходити з однієї форми на іншу. Це дозволяє наочно побачити всю інформацію та заощадить час, необхідний на оформлення замовлення. Програма написана мовою BPEL. Текст програми наведено Додатку А.5.

3.4.2 Заповнення форм процесу «Оформлення online-замовлення»

Через декілька секунд після того, як процес запущено, з'явиться форма «Вибір товару» (рис. 3.11). Із запропонованого списку товарів необхідно вибрати найменування товару для замовлення. Після цього натиснути кнопку "Complete". Ця форма автоматично замінюється на наступну форму «Адреса доставки» (рис. 3.12) у якій необхідно заповнити поля, що вказують місце доставки товару. При цьому, верхнє поле про найменування товару перенеслося автоматично на цю форму. Після того, як переконались, що всі поля форми заповнені, можна натиснути кнопку «Complete». Знову за кілька секунд з'явиться третя форма «Оплата товару» (рис. 3.13) із вже заповненими полями про товар та адресу. У цій формі необхідно вказати всю необхідну інформацію про кредитну картку, після чого натиснути кнопку «Complete». Це остання форма. Вона веде процес «Оформлення online-замовлення» до завершення. Список завдань користувача оновлюється.

3.4.3 Адміністрування процесів

У консолі адміністратора на вкладці «Process» можна побачити точний час і дату, коли був запущений процес, і переконатися, що процес «Оформлення online-замовлення» було завершено. Також там можна

переглянути користувача, який працював з цим процесом. Звідси можна видаляти процеси чи зупиняти їх у певний час. Насправді можливо контролювати процес у будь-який час. На вкладці «Instances» вибрати потрібний процес, натиснути на кнопку «Resume» і візуалізувати діаграму (рис. 3.15), де буде видно, на якій стадії знаходиться даний процес. Тут можна переглядати час і дату початку процесу і коли він був активний.

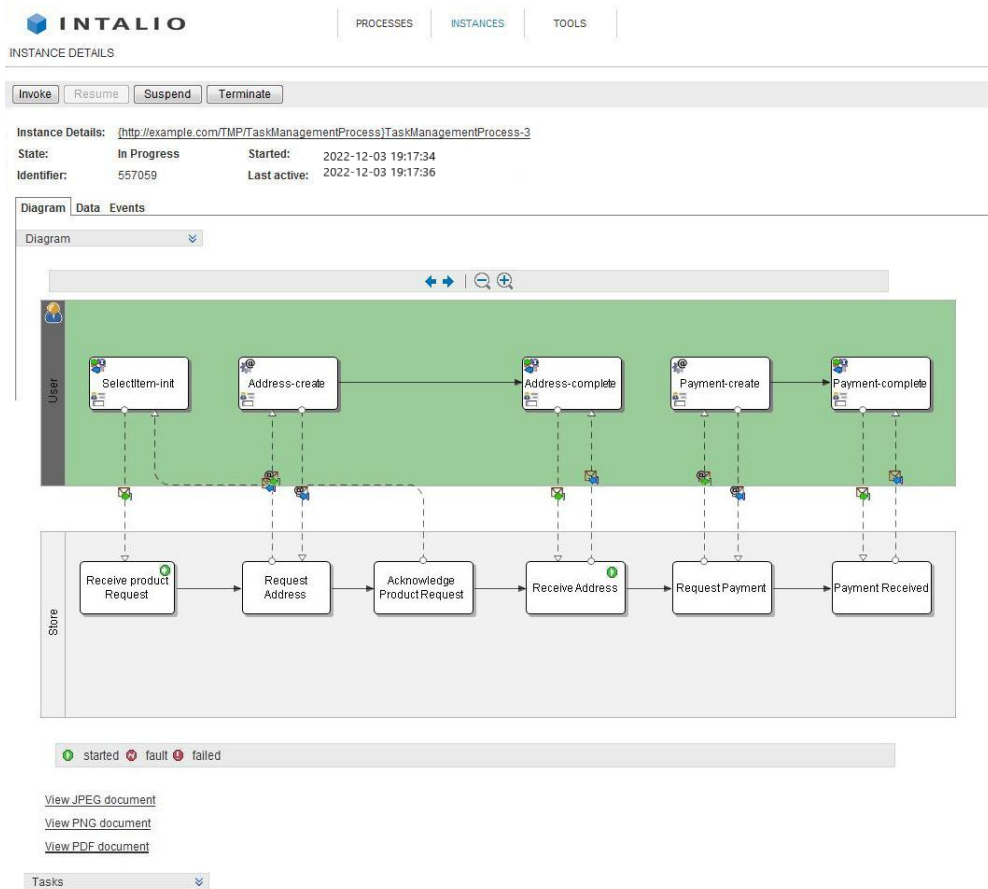


Рисунок 3.15 – Моніторинг бізнес-процесу

3.5 Розробка підсистеми автоматизації бізнес-процесу «Оформлення online-замовлення»

Програма автоматизації бізнес-процесу «Оформлення online-замовлення» дозволяє користувачеві в режимі реального часу замовити

продукцію, що його цікавить. Крім того, створений продукт надає адміністратору інформацію про стан процесу.

Програмний продукт розроблено з використанням мови BPEL [27].

Мова BPEL дозволяє задавати БП і те, як вони пов'язані з вебсервісами, а також як БП використовують вебсервіси для досягнення поставлених завдань і які вебсервіси надаються БП.

Властивості мови BPEL:

- описує логіку БП;
- підтримує багаторазові модулі виконання, що довго виконуються, які є також переривчастими;
- продовжує роботу, яка зупинена через відмову згорнути роботу, яка буде відновлена;
- направляє повідомлення, що надходять до правильного місця в межах правильного процесу;
- визначає набір дій, які мають відбутися у певний час та у визначеному порядку;
- надсилає повідомлення вебсервісам [27].

БП визначається в BPEL як XML-файл, кореневим елементом якого є елемент `<process>`. У кореневий елемент можуть бути в загальному випадку вкладені елементи, представлені в таблиці 3.3.

Таблиця 3.3 – Основні елементи BPEL

Елемент	Що визначає
<code><partners></code>	Опис партнерів (елемент в який вкладено один або кілька елементів <code><partner></code>)
<code><partnerLinks></code>	Опис зв'язків між партнерами (елемент в який вкладено один або кілька елементів <code><partnerLink></code>);
<code><variables></code>	Визначення змінних (елемент, в який вкладено один або кілька елементів <code><variable></code> або контейнерів <code><containers></code>)
<code><correlationSets></code>	Кореляційні набори (елемент, в який вкладено один або кілька елементів <code><correlationSet></code>)

Продовження таблиці 3.3

Елемент	Що визначає
<faultHandlers>	Обробники помилок
<eventHandlers>	Обробники подій
<compensationHandler>	Обробник компенсації

Оскільки BPEL є мовою, яка будує з низки вебсервісів новий сервіс, БП здебільшого складається зі звернень до інших сервісів та отримання звернень від клієнтів. Сервіси, що викликають БП BPEL, та послуги, що викликаються БП BPEL, називаються партнерами. Відносини між партнерами у взаємодії, зазвичай, діалогові, потребують двосторонньої взаємодії сервісів [27].

Тип зв'язку між партнерами характеризує відносини між двома сервісами шляхом визначення ролей, які кожен із них грає в діалозі, та завданням типу порту, що надається кожним сервісом для отримання повідомлень у контексті діалогу:

```
<partnerLinkType name="ім'я_типу_зв'язку">
  <role name="ім'я_ролі">
    <portType="ім'я_типу_порта">
  </role>
</partnerLinkType>
```

Сервіс, з яким БП взаємодіє, моделюється як зв'язок із партнером. Кожен зв'язок із партнером характеризується типом зв'язку:

```
<partnerLink name="ім'я_зв'язку"
  partnerLinkType name="ім'я_типу_зв'язку">
  myRole name="ім'я_ролі">
  partnerRole name="ім'я_ролі" />
```

Кожному зв'язку привласнюється ім'я, що використовується для всіх взаємодій сервісу через цей зв'язок. Роль самого БП є атрибутом myRole, а роль партнера - атрибутом partnerRole. У виродженому випадку зв'язок має лише

одну роль. У таких сценаріях один сервіс висловлює готовність зв'язуватися з будь-яким іншим сервісом без вимог до партнера [27].

Визначення змінної (елемент `<variable>`) містить атрибут, що визначає повне кваліфіковане ім'я змінної, і один із атрибутів визначення її типу: `type` – для простого типу XML Schema, `element` – для складного типу XML Schema, `messageType` – для типу повідомлення WSDL [27].

Елемент привласнення має вигляд:

```
<copy>
  <from .../>
  <to .../>
</copy>
```

У визначенні джерела (елемент `<from>`) може задаватися ім'я змінної (якщо тип змінної - повідомлення WSDL, то може вибиратися частина повідомлення) або ім'я копійованого зв'язку партнера та ролі, або властивість змінної, або вираз, або літерал. У визначенні приймача (елемент `<to>`) може задаватися ім'я змінної, ім'я партнера або властивість змінної [27].

Примітивні дії, які безпосередньо відповідають за обмін повідомленнями – `<recv>` (отримання повідомлення), `<reply>` (відправлення повідомлення) та `<invoke>` (виклик деякого вебсервісу з відправкою йому повідомлення та отриманням повідомлення у відповідь. Наприклад, нижче наведено форму елемента `<invoke>`, а форми двох інших елементів подібні до наведеної:

```
<invoke partnerLink=" зв'язок"
  portType="тип_порта" operation="операція"
  inputVariable="змінна"?
  outputVariable="змінна"?>
  <correlations>?
    <correlation set="кореляційний_набір"
      initiate="yes|no"?
      pattern="in|out|out-in"/>+
  </correlations>
```

```

        <catch faultName="помилка"
faultVariable="змінна"?>*
        дія
    </catch>
    <catchAll>?
        дія
    </catchAll>
    <compensationHandler>?
        дія
    </compensationHandler>
</invoke>

```

У наведеній вище формі символ "?" означає "нуль або один", символ "*" - "нуль або більше", символ "+" - "один або більше", символ "|" - "або".

Обов'язковими для всіх трьох дій є атрибути `partnerLink`, `portType` та `operation`, що визначають взаємодію з партнером. Атрибути `inputVariable` та `outputVariable` у дії `<invoke>` визначають змінні для вхідного та вихідного повідомлень, для двох інших дій задається лише один атрибут – `variable`.

Вкладений елемент `<correlations>` задає у всіх трьох діях кореляційні набори для співвідношення повідомлення з певним екземпляром сервісу. Вкладені елементи `<catch>` та `<catchAll>`, а також `<compensationHandler>` застосовуються лише у дії `<invoke>`.

Крім того, у дії `<receive>` може задаватися також атрибут `createInstance`, що визначає створення нового екземпляра процесу, а в дії `<replay>` - атрибут `faultName`, якщо процес надсилає повідомлення про помилку.

Кореляційний набір (елемент `<correlationSet>` в описі процесу) визначається перерахуванням властивостей, що його становлять. Кожен екземпляр БП має свої екземпляри всіх певних у ньому кореляційних наборів. Дії, пов'язані з прийомом та відправленням повідомлень, посилаються на кореляційний набір, чим досягається прив'язка конкретного повідомлення до конкретного екземпляра процесу [27].

Обробники помилок задаються елементом `<faultHandlers>`, який вкрито елементи `<catch>`. Кожен елемент `<catch>` визначає дії щодо певної

помилки, ім'я якої задається його аргументом `faultName`. Елемент `<catchAll>` обробляє будь-які помилки [27].

Компенсація може викликатись явно чи неявно. Явний виклик компенсації здійснюється дією `<compensate>`, аргументом якої є ім'я області, дії якої компенсуються. Явний виклик компенсації можливий лише з обробників помилок та компенсації. Неявно компенсація викликається для всіх областей, вкладених у ту область, дії якої компенсуються.

З процесом або будь-якою областю дії може бути пов'язаний набір обробників подій, які виконуються паралельно при скоєнні відповідної події. Події можуть бути повідомленнями, що надходять ззовні, або сигналом, що надходить у заданий користувачем час. Обробники подій визначаються елементом `<eventHandlers>`. Вміст цього елемента - такий самий, як і елемента `<pick>`, але елемент вибору очікує на прихід одного з повідомлень, а обробники подій обробляють повідомлення, що надходять асинхронно [27].

Текст програми представлено у Додатку А.5.

3.6 Висновки до розділу 3

В цьому розділі, відповідно до мети та завдань роботи виконано опис досліджуваного БП «Оформлення online-замовлення», моделювання БП з використанням елементів BPMN-нотації в середовищі `Intalio | Designer` та моніторинг БП в середовищі `Intalio | Server`.

Розробку форм виконано за допомогою редактора форм панелі `Workflow Form Editor Palette` та з використанням мови опису визначень та реалізацій процесів `XPDL`.

Програму автоматизації БП «Оформлення online-замовлення» розроблено з використанням мови формального опису бізнес-процесів та протоколів їх взаємодії між собою `BPEL`. Вона дозволяє користувачеві в режимі реального часу замовити в ІМ продукцію, що його цікавить, а адміністратору надає інформацію про стан БП.

4 МЕТОДИКА ВИКОРИСТАННЯ РОЗРОБЛЕНОЇ ПІДСИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСУ «ОФОРМЛЕННЯ ONLINE-ЗАМОВЛЕННЯ»

4.1 Призначення підсистеми та інструкція з використання

Розроблена підсистема автоматизації бізнес-процесу «Оформлення online-замовлення» надає користувачеві можливість оформити замовлення на необхідний товар, а адміністратору – в режимі реального часу здійснювати моніторинг БП, а саме:

- стартувати процес;
- отримувати інформацію про стан процесу;
- отримувати інформацію про співробітника, відповідального за процес;
- отримувати інформацію у вигляді діаграми процесу;
- призначати пріоритети процесам;
- додавати/видаляти/зупиняти процеси.

Щоб завантажити Intalio | Server необхідно у командному рядку прописати шлях до файлу startup.bat. Після цього запуститься Geronimo. Потім в адресному рядку веббраузера необхідно ввести адресу консолі. Для користувача це <http://localhost:8080/ui-fw/login.htm>. Після чого користувач потрапляє на сторінку, що містить консоль користувача, в якій потрібно ввести логін користувача та пароль (рис. 4.1). Після того як користувач ввів логін та пароль, з'являється сторінка користувача, що містить 3 вкладки:

- Tasks (завдання);
- Notifications (оповіщення);
- Processes (процеси).

Розглянемо докладніше кожен із елементів цього меню.

При виборі закладки «Tasks» з'являється сторінка (рис. 4.2), де видно всі завдання користувача. У цьому прикладі - це заповнення форм.

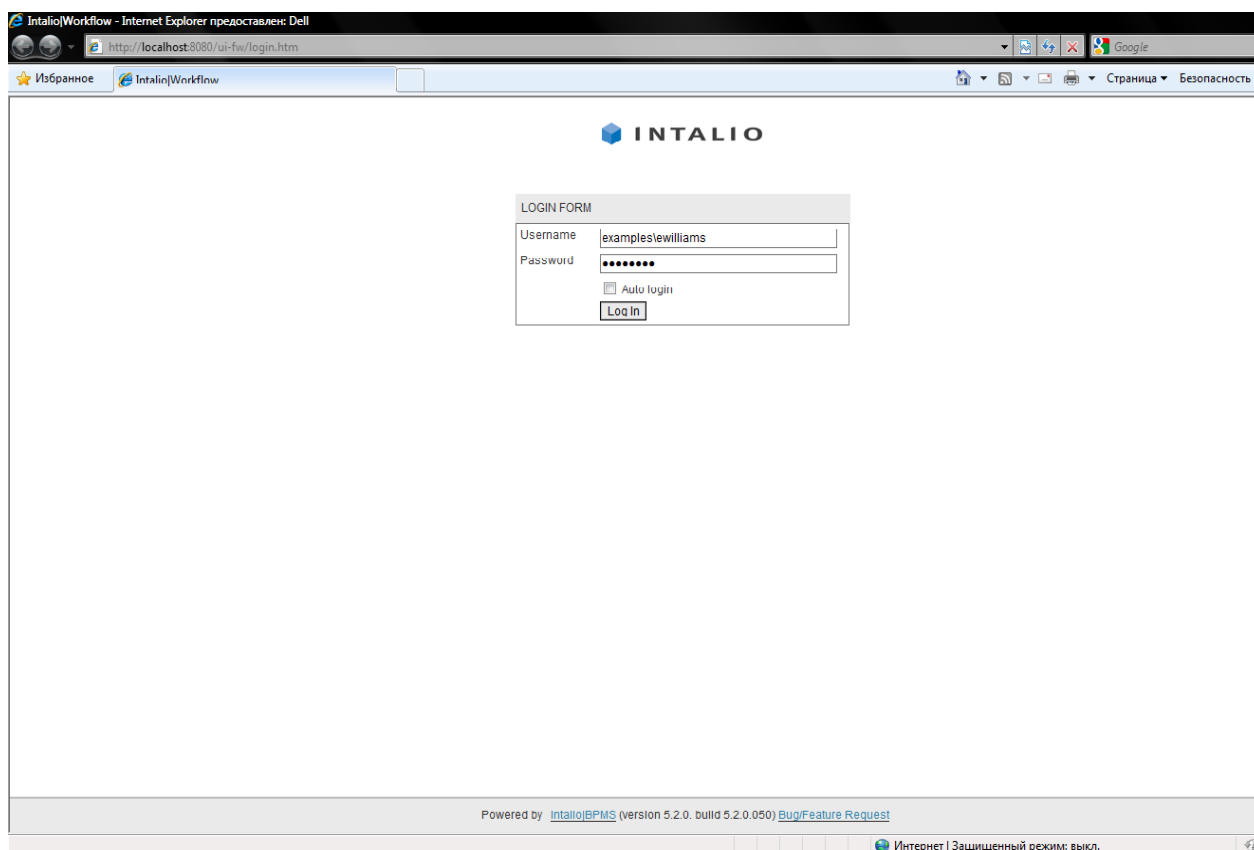


Рисунок 4.1 – Сторінка консолі користувача

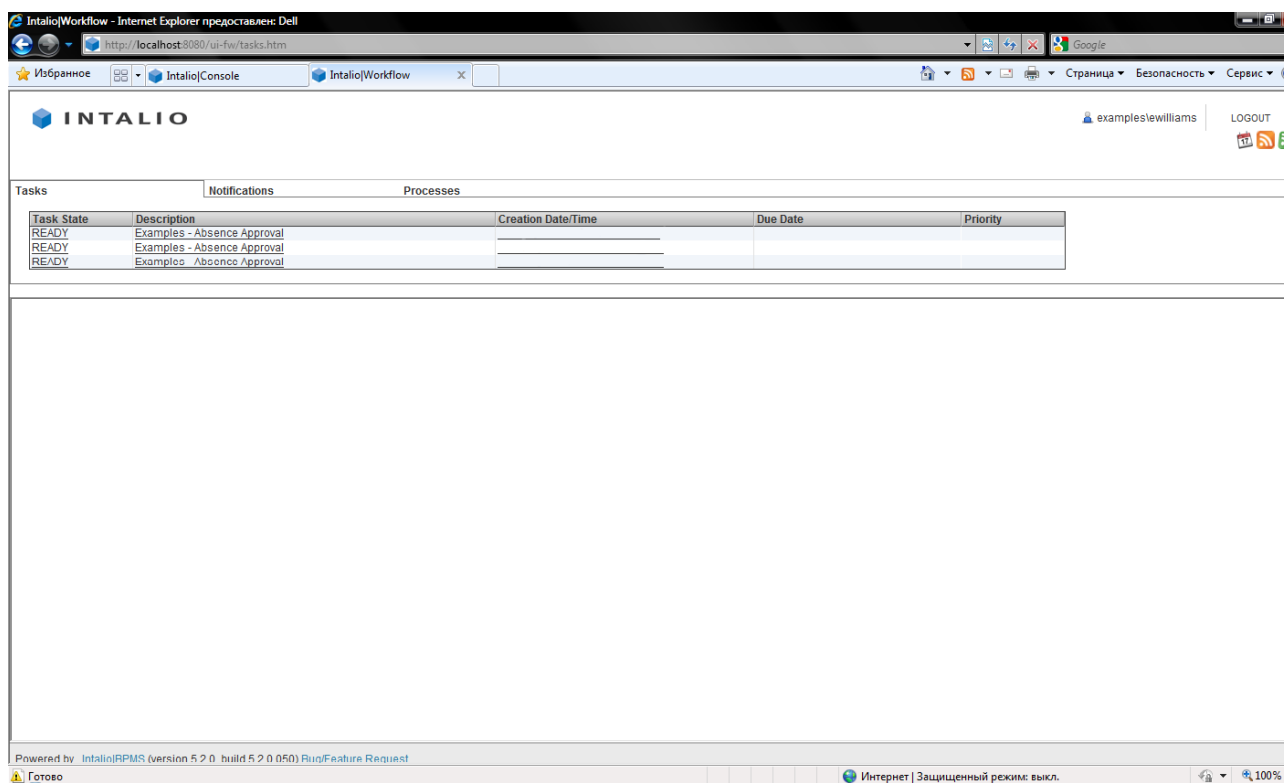


Рисунок 4.2 – Сторінка завдань користувача

На вкладці Notifications містяться повідомлення від адміністратора або менеджера, в яких можуть бути примітки про угоду або відмову, щодо прийняття будь-якого рішення.

На вкладці Processes містяться всі процеси, які потрібно виконати користувачеві (рис. 4.3).

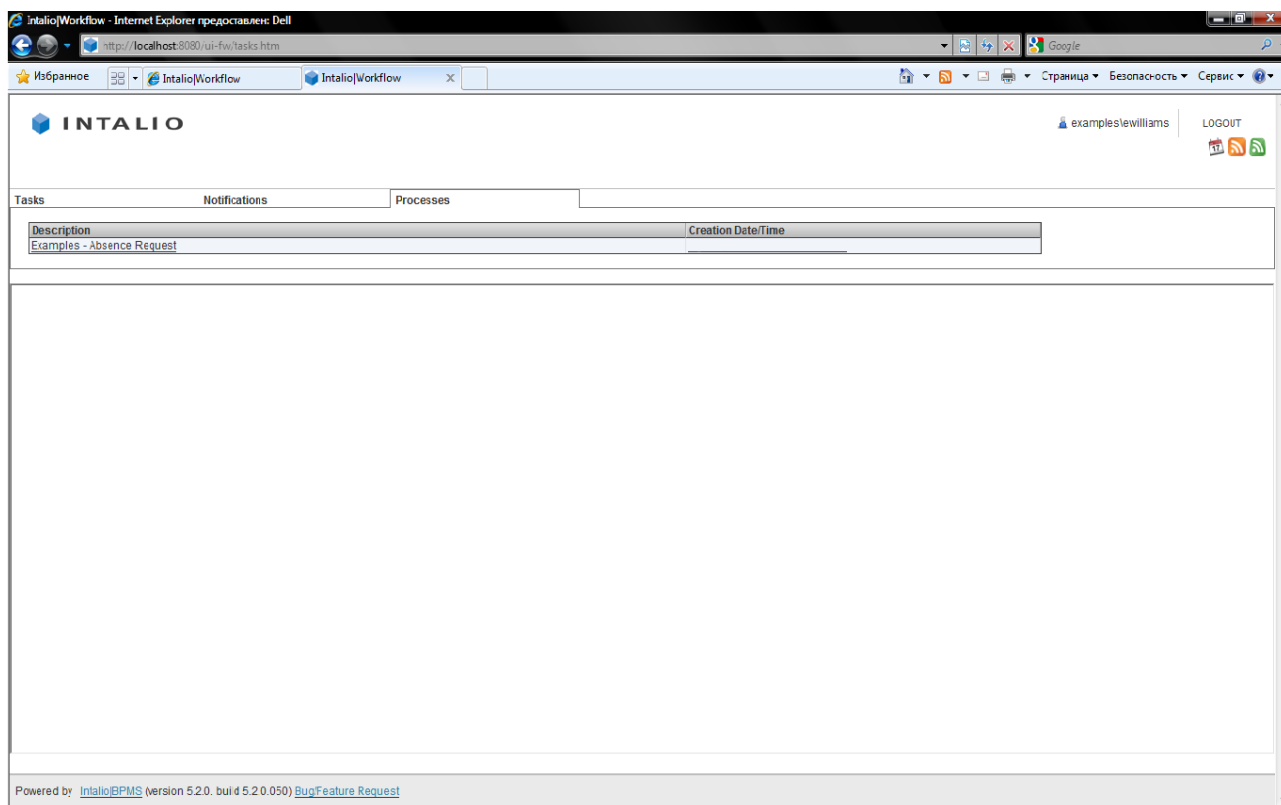


Рисунок 4.3 – Сторінка процесів користувача

Щоб розпочати процес, необхідно з вкладки «Processes» вибрати необхідний процес і натиснути на кнопку «Start». Після чого процес автоматично запуститься і з'явиться в списку завдань. Після цього необхідно зі списку завдань вибрати процес і за кілька секунд з'явиться форма вибору товару (рис. 4.4). У цій формі необхідно вибрати найменування товару для покупки та натиснути кнопку «Complete».

Product: SELECT

SELECT
Product #2
Product #3

Claim Save Complete

Powered by [IntalioBPMS](#) (version 5.2.0. build 5.2.0.050) [Bug/Feature Request](#)

Рисунок 4.4 – Форма вибору товару

Ця форма автоматично замінюється на наступну форму «Адреса доставки» (рис. 5.5). в якій необхідно заповнити поля, що вказують місце доставки товару. Зауважимо, що верхнє поле про найменування товару перенеслося автоматично на цю форму. Після того, як переконайтеся, що всі поля форми заповнені, натисніть кнопку «Complete».

Знову за кілька секунд з'явиться третя форма «Оплата товару» (рис. 5.6) з уже заповненими полями про товар та адресу. У цій формі необхідно вказати всю необхідну інформацію про Вашу кредитну картку, після чого натисніть кнопку «Complete». Це остання форма. Вона веде процес «Оформлення online-замовлення» до завершення. Список завдань користувача оновлюється.

The screenshot shows a web form for entering shipping information. At the top left is a small icon of a person. Below it, the 'Ordered Items' section contains a 'Reference:' label followed by a text box with the value 'P3'. The 'Shipping Address' section includes several text boxes: 'Street1:' with 'S-91', 'Street2:' with 'Parkin Street', 'City:' with 'Redwood', 'Postal Code:' with '160020', 'State/Region:' with 'Green Park', and 'Country:' with 'USA'. At the bottom of the form are three buttons: 'Claim', 'Save', and 'Complete'. A 'Chained_Execution' label is positioned on the right side of the form. The footer of the browser window displays the text 'Powered by IntalioBPMS (version 5.2.0. build 5.2.0.050) Bug/Feature Request'.

Ordered Items
Reference: P3

Shipping Address
Street1: S-91
Street2: Parkin Street
City: Redwood Postal Code: 160020 State/Region: Green Park
Country: USA

Claim Save Complete

Chained_Execution

Powered by IntalioBPMS (version 5.2.0. build 5.2.0.050) Bug/Feature Request

Рисунок 4.5 – Форма введення адреси доставки

This screenshot shows a web form for entering payment details. It includes the same 'Ordered Items' section as the previous form. The 'Shipping Address' section is also present. Below this, the 'Credit Card' section features a dropdown menu for 'Type:' set to 'MasterCard', a 'Card number:' text box with '36147-426574687', an 'Expiration date (MM/YY)' text box, and a 'Name as displayed on the card:' text box with 'Charles James'. A 'Chained_Execution' label is on the right. At the bottom are 'Claim', 'Save', and 'Complete' buttons. The footer text is identical to the previous form.

Ordered Items
Reference: P3

Shipping Address
Street1: S-91
Street2: Parkin Street
City: Redwood Postal Code: 160020 State/Region: Green Park
Country: USA

Credit Card
Type: MasterCard
Card number: 36147-426574687 Expiration date (MM/YY)
Name as displayed on the card: Charles James

Claim Save Complete

Chained_Execution

Powered by IntalioBPMS (version 5.2.0. build 5.2.0.050) Bug/Feature Request

Рисунок 4.6 – Форма оплати товару

Завершуємо роботу користувача. Щоб вийти з консолі користувача, натискаємо кнопку «Logout».

Для того, щоб зайти адміністратору, необхідно в адресному рядку веббраузера вести адресу консолі: `http://localhost:8080/bpms-console/login.htm`. Адміністратор потрапляє на консоль, де необхідно ввести логін і пароль.

Після того як ввели логін та пароль, з'являється сторінка адміністратора. Ця сторінка містить 3 вкладки:

- Processes (процеси);
- Instances (запити);
- Tools (інструменти).

Розглянемо докладніше кожен із елементів цього меню.

На вкладці Processes (рис. 4.7) можна переглянути всі існуючі процеси та їх характеристики.

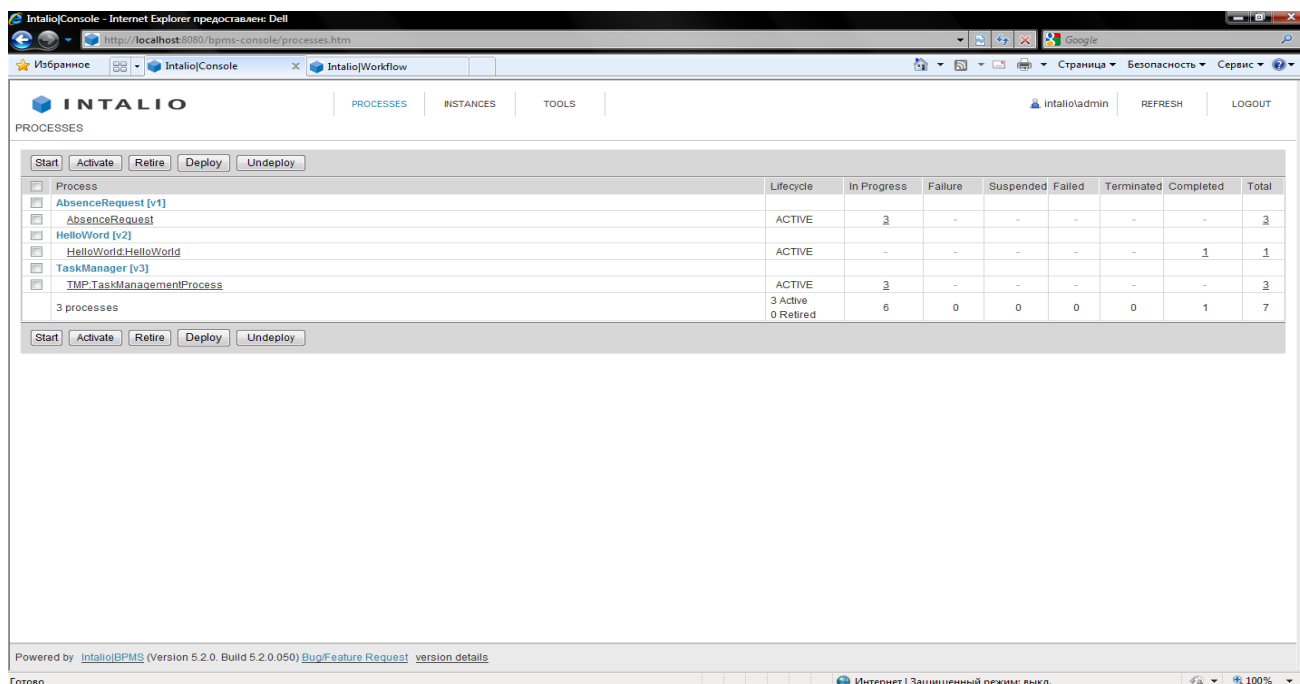


Рисунок 4.7 – Сторінка процесів адміністратора

На вкладці Instances вибираємо процес і натискаємо на нього. З'являється діаграма процесу (рис. 4.8). На діаграмі зеленим кружечком

відзначені виконані стадії процесу, а також коли вони були розпочаті і закінчені. Виконані події можна переглянути на вкладці «Events».

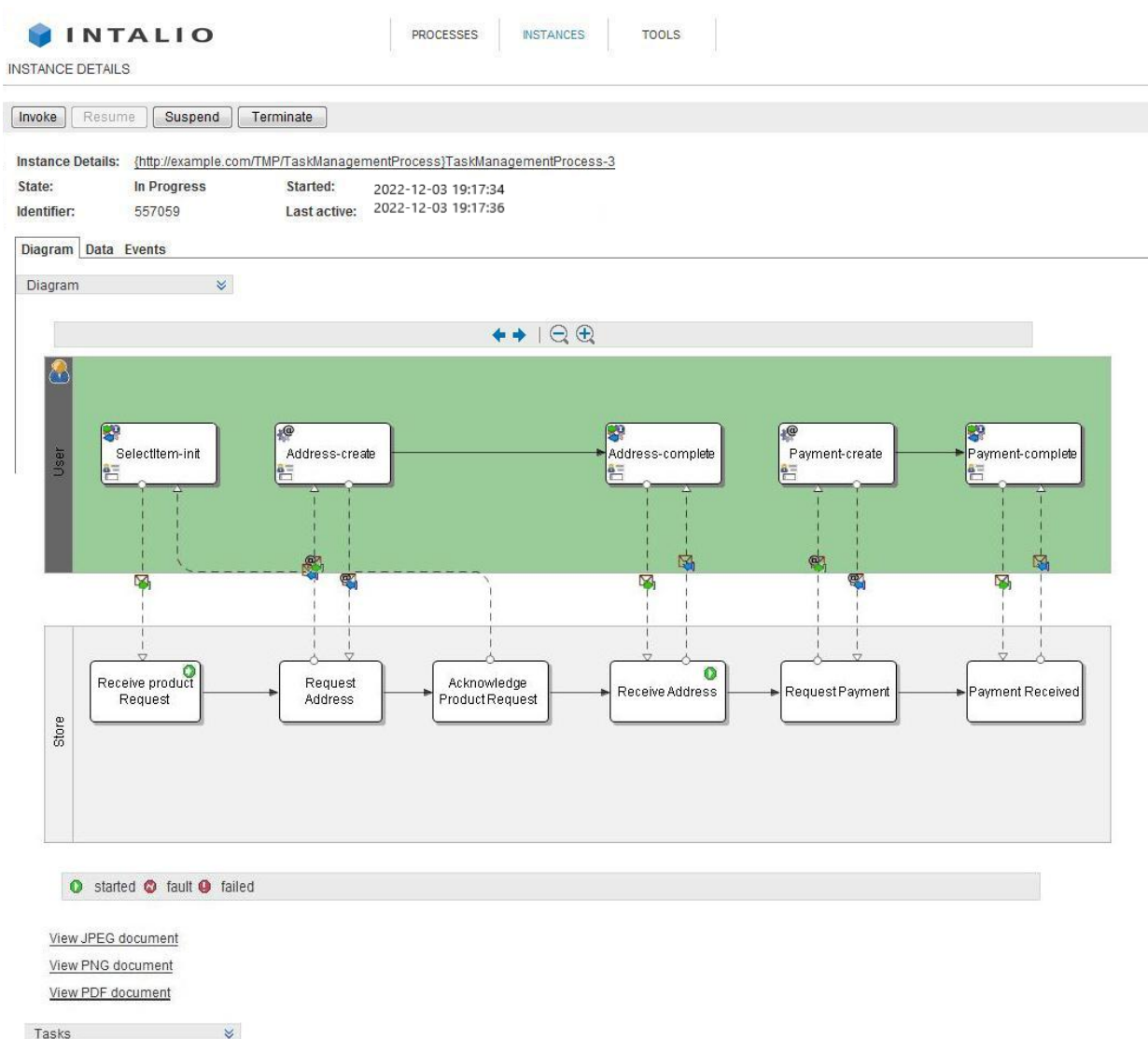


Рисунок 4.8 – Сторінка запитів

Завершуємо роботу адміністратора. Щоб вийти з консолі адміністратора, необхідно натиснути кнопку «Logout».

4.2 Повідомлення про помилки

Розроблений програмний продукт забезпечує стійку роботу та обробку помилкових ситуацій. Передбачено два види повідомлень:

- інформаційні повідомлення (табл. 4.1);
- повідомлення про помилки (табл. 4.2).

Таблиця 4.1 – Інформаційні повідомлення

Текст повідомлення	Зміст повідомлення
Ваш процес успішно стартував	Система підтверджує, що процес успішно запусився, і сповіщає про це користувача або адміністратора (залежить від того, в якій консолі запускався процес).
Замовлення прийнято	Система підтверджує, що обробила вхідні дані та підтверджує замовлення

Таблиця 4.2 – Повідомлення про помилкові ситуації

Повідомлення	Умова виникнення	Реакція користувача
Ви вказали неправильне ім'я користувача або пароль.	У консолі було введено неправильний логін або пароль.	У разі виникнення цієї помилки користувач повинен ввести правильні дані
Невірні дані кредитної картки. Спробуйте ще раз.	Дані, введені користувачем, не збігаються з даними, що зберігаються в базі даних системи	У разі виникнення цієї помилки користувач повинен перевірити правильність даних за своєю кредитною карткою
Обов'язкові поля заповнені не в повному обсязі	Дані, які ввів користувач неповні	У разі виникнення цієї помилки користувач повинен ввести всі обов'язкові поля

Таким чином, якщо виникає помилкова ситуація, то виводиться відповідне повідомлення з підказкою, що полегшує та прискорює роботу користувача надалі.

4.3 Висновки до розділу 4

В цьому розділі відповідно до мети та завдань магістерської роботи виконано розробку методики використанн розробленої підсистеми автоматизації БП «Оформлення online-замовлення». Наведено інструкцію користувача та адміністратора, а також опис інформаційних повідомлень підсистеми та повідомлень про помилкові ситуації.

ВИСНОВКИ

В результаті проведених досліджень були розглянуті особливості методології BPM, яка дозволяє автоматизувати та оптимізувати БП підприємства, що сприяє реструктуризації, контролю та управлінню виробничими процесами, що включають людей і системи, для більш ефективного ведення бізнесу.

Для успішної реалізації управлінської методології BPM потрібне впровадження сучасних BPM-систем, функціональні можливості яких можуть бути розширені за допомогою відповідного програмного інструментарію.

В середовищі системи Intalio | BPMS було розроблено підсистему автоматизації БП «Оформлення online-замовлення», що надає користувачеві можливість оформити замовлення на потрібний товар, а адміністратору – в режимі реального часу здійснювати моніторинг БП.

Моделювання БП виконується в середовищі Intalio | Designer з використанням опису визначень та реалізацій процесів XPDЛ, а моніторинг реалізації БП – у середовищі Intalio|Server з використанням мови формального опису БП та протоколів їх взаємодії між собою BPEL.

Наукова новизна роботи полягає в тому, що запропоновано методику автоматизації БП на основі удосконалених функцій моделювання та моніторингу в середовищі BPM-системи.

Практична цінність роботи полягає в тому, що підсистема автоматизації БП надає зручні інструменти моделювання та моніторингу, що дозволять підвищити ефективність автоматизованої системи управління підприємством.

В подальшій роботі можливо розширити функціональні можливості підсистеми шляхом реалізації процедури оптимізації БП.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Швиданенко, Г. О. Формування бізнес-моделі підприємства: навчальний посібник / Г. О. Швиданенко, Н. В. Ревуцька. — К.: КНЕУ, 2013. — 423 с.
2. Повышение эффективности внедрения ВРМ-систем: командный подход [Электрон. ресурс]. — Режим доступа: <http://iso.ru/ru/press-center/publications/1438.phtml>.
3. Пинаев, Д. Моделирование бизнес-процессов: доступно о сложном [Электрон. ресурс] / Д. Пинаев, Д. Веретенников. — Режим доступа: <https://finance-abm.com/ru/library/articles/modelirovanie-biznes-processov-dostupno-o-slojnom>.
4. Опис бізнес-процесів компанії. Кому, навіщо і як? [Електрон. ресурс]. — Режим доступу: <https://gc.ua/uk/opis-biznes-procesiv-kompani%D1%97-komu-navishho-i-yak/>.
5. Гриненко, В. В. Конспект лекцій з курсу «Управління та удосконалення бізнес-процесів» / В. В. Гриненко. — Харків: ХНУМГ ім. О. М. Бекетова, 2018 — 147 с.
6. Стратегічна карта як інструмент ефективного управління фінансовою стійкістю підприємства [Електрон. ресурс]. — Режим доступу: <http://www.ukr.vipreshebnik.ru/strateg/4461-strategichna-karta-yak-instrument-efektivnogo-upravlinnya-finansovoyu-stijkisty-pidpriemstva.html>.
7. Чернобай, Л. І. Бізнес-процеси підприємства: класифікація та структурно-ієрархічна модель / Л.І. Чернобай // Економічний аналіз. - 2015. - Том 22, № 2. - С.171-182.
8. Гурова, В. О. Оптимізація бізнес-процесів на підприємстві в умовах кризи / В. О. Гурова // Ukrainian Journal of Applied Economics. - 2016. — Vol. 1, № 1. - С.75-82.

9. Криворучко, О.М. Інтегрований підхід до удосконалення бізнес-процесів підприємства / О.М. Криворучко // Економіка транспортного комплексу. – 2018. - Вип. 32, С. 17-29.
10. Гончарова, О.М. Реінжиніринг бізнес-процесів як спосіб підвищення ефективності управління [Електрон. ресурс] / О.М. Гончарова. – Режим доступу: <http://www.economy.nauka.com.ua/?op=1&z=943#>.
11. Технології Workflow [Електрон. ресурс]. – Режим доступу: <http://rua.pp.ua/tehnologiya-upravleniya-potokami-rabot-24056.html>.
12. What is BPMS? [Electronic resource]. – Access mode: <https://www.creatio.com/bpms>.
13. What is business process management software (BPMS)? [Electronic resource]. – Access mode: <https://www.techtarget.com/searchcio/definition/Business-process-management-suite-BPMS>.
14. Основні системи управління бізнес-процесами. Системи керування бізнес-процесами. Розробка та впровадження інформаційної системи [Електрон. ресурс]. – Режим доступу: <https://sa-gri.ru/uk/franchising/osnovnye-sistemy-upravleniya-biznes-processami-sistemy-upravleniya.html>.
15. ARIS Express [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/ARIS_Express.
16. What is jBPM? [Electronic resource]. – Access mode: <https://www.jbpm.org/>.
17. RunaWFE Free Workflow System [Electronic resource]. – Access mode: https://runawfe.org/RunaWFE_Free_Workflow_System.
18. Bizagi. Описание. Пример [Электрон. ресурс]. – Режим доступа: <https://habr.com/ru/company/trinion/blog/273017/>.
19. Система управления бизнес процессами Intalio BPMS [Электрон. ресурс]. – Режим доступа: <http://www.asit.kz/index.php%3Fid=37.html>.
20. Camunda [Electronic resource]. – Access mode: <https://en.wikipedia.org/wiki/Camunda> <https://www.crmoz.com/blogs/post/business-process-modeling>.

21. Business consulting services [Electronic resource]. – Access mode: https://www.ibm.com/uk-en/consulting/business-strategy?utm_content=SRCWW&p1=Search&p4=43700068004150948&p5=p&gclid=CjwKCAiAhKycBhAQEiwAgf19emS42vRcijLk-zdmRtFX3egvyl8Hhn9pun3qCGO_-rAt633CWb2BxRoCoZQQA_vD_BwE&gclidsrc=aw.ds.
22. Побудова бізнес-моделі підприємства [Електрон. ресурс]. – Режим доступу: <https://www.fox-manager.com.ua/fox-manager-2-0-bpa.html>.
23. Архитектура INTALIO BPMS [Электрон. ресурс]. – Режим доступа: <https://www.citeck.ru/services/intalio-bpms/intalio-architecture/>.
24. Виноградова, О.В. Електронний бізнес: навчальний посібник / О.В. Виноградова, Н.І. Дрокіна. - Київ: Державний університет телекомунікацій, 2018. - 292 с.
25. Як створити успішний інтернет-магазин? [Електрон. ресурс]. – Режим доступу: <https://torgsoft.ua/articles/stati/uspeshnyij-internet-magazin/>.
26. What is XML Process Definition Language (XPDL) [Electronic resource]. – Access mode: <http://xpdl.org/what-is-xml-process-definition-language-xpdl/>.
27. BPEL (Business Process Execution Language) Gavin Wright [Electronic resource]. – Access mode: <https://www.techtarget.com/searchapparchitecture/definition/BPEL-Business-Process-Execution-Language>.

ДОДАТОК А
Текст програми

A.1 Файл Address.xform.all.xpdl

```

<?xml version="1.0" encoding="utf-8" ?>
<xpdl:definitions
targetNamespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address"
  xmlns:Address="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address"
  xmlns:xpdl="http://schemas.xmlsoap.org/xpdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/xpdl/soap/">

  <xpdl:types>
    <!-- declare the namespace prefix again just to make sure
    this particular fragment is valid stand-alone schema. -->
    <xsd:schema targetNamespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address"
      xmlns:Address="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified">
      <xsd:include schemaLocation="Address.xform.xsd"/>
      <!-- the xpdl message elements -->
      <xsd:element name="initProcessRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="taskId" type="xsd:string"/>
            <xsd:element name="participantToken" type="xsd:string"/>
            <xsd:element name="taskOutput">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element ref="Address:output"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="initProcessResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="status" type="xsd:string"/>
            <xsd:element name="errorCode" type="xsd:string" minOccurs="0"/>
            <xsd:element name="errorReason" type="xsd:string" minOccurs="0"/>
            <xsd:element name="taskMetaData" minOccurs="0">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="nextTaskId" type="xsd:string"/>
                  <xsd:element name="nextTaskURL" type="xsd:string"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="createTaskRequest">
        <xsd:complexType>

```

```

<xsd:sequence>
  <xsd:element name="taskMetaData" type="Address:taskMetaDataType"/>
  <xsd:element name="participantToken" type="xsd:string"/>
  <xsd:element name="taskInput">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Address:input"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="response">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="isChainedAfter" type="xsd:boolean" minOccurs="0"/>
      <xsd:element name="taskMetaData" type="Address:taskMetaDataType"/>
      <xsd:element name="status" type="xsd:string"/>
      <xsd:element name="errorCode" type="xsd:string" minOccurs="0"/>
      <xsd:element name="errorReason" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="accessControlType">
  <xsd:sequence>
    <xsd:element name="user" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="role" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="scheduledActionsDetails">
  <xsd:choice minOccurs="0" maxOccurs="1">
    <xsd:element name="until" type="xsd:dateTime" maxOccurs="1" minOccurs="0" />
    <xsd:element name="for" type="xsd:duration" maxOccurs="1"
minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="taskMetaDataType">
  <xsd:sequence>
    <xsd:element name="taskId" type="xsd:string"/>
    <xsd:element name="taskState" type="xsd:token"/>
    <xsd:element name="taskType" type="xsd:token"/>
    <xsd:element name="description" type="xsd:string"/>
    <xsd:element name="processId" type="xsd:string"/>
    <xsd:element name="creationDate" type="xsd:dateTime"/>
    <xsd:element name="userOwner" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
    <xsd:element name="roleOwner" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>

    <xsd:element name="claimAction" type="Address:accessControlType" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="revokeAction" type="Address:accessControlType" maxOccurs="1"
minOccurs="0"/>
    <xsd:element name="saveAction" type="Address:accessControlType" maxOccurs="1" minOccurs="0"/>
  </xsd:sequence>

```

```

        <xsd:element name="completeAction" type="Address:accessControlType" maxOccurs="1"
minOccurs="0"/>

        <xsd:element name="formUrl" type="xsd:anyURI"/>
        <xsd:element name="failureCode" type="xsd:string"/>
        <xsd:element name="failureReason" type="xsd:string"/>
        <xsd:element name="priority" type="xsd:int"/>
        <xsd:element name="scheduledActions">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="expiration" type="Address:scheduledActionsDetails"/>
                    <xsd:element name="deferActivation"
type="Address:scheduledActionsDetails"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="userProcessCompleteSOAPAction" type="xsd:string"/>
        <xsd:element name="isChainedBefore" type="xsd:boolean"/>
        <xsd:element name="previousTaskId" type="xsd:string"/>
        <xsd:element name="userProcessEndpoint" type="xsd:string"/>
        <xsd:element name="userProcessNamespaceURI" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="notifyTaskCompletionRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="taskMetaData" type="Address:taskMetaDataType"/>
            <xsd:element name="taskOutput">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="Address:output"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="status" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="notifyRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="metadata">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
                        <xsd:element name="creationDate" type="xsd:dateTime" minOccurs="0" maxOccurs="1"/>
                        <xsd:element name="userOwner" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
                        <xsd:element name="roleOwner" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
                        <xsd:element name="formUrl" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>

```

```

        <xsd:element name="input">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element ref="Address:input"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="notifyResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="status" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="errorCode" type="xsd:string" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="errorReason" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="escalateRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="taskId" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="userOwner" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="roleOwner" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="escalateResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="status" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

    <!-- end of the xpdL message elements -->
</xsd:schema>
</xpdL:types>

<!-- Messages Create -->
<xpdL:message name="createTaskRequest">
    <xpdL:part name="root" element="Address:createTaskRequest"/>
</xpdL:message>
<xpdL:message name="createTaskResponse">
    <xpdL:part name="root" element="Address:response"/>
</xpdL:message>
<!-- Messages Complete -->
<xpdL:message name="notifyTaskCompletionRequest">
    <xpdL:part name="root" element="Address:notifyTaskCompletionRequest"/>
</xpdL:message>

```

```

<xpdl:message name="notifyTaskCompletionResponse">
  <xpdl:part name="root" element="Address:response"/>
</xpdl:message>
<!-- Messages Escalate-->
<xpdl:message name="escalateRequest">
  <xpdl:part name="root" element="Address:escalateRequest"/>
</xpdl:message>
<xpdl:message name="escalateResponse">
  <xpdl:part name="root" element="Address:escalateResponse"/>
</xpdl:message>
<!-- Messages Init -->
<xpdl:message name="initProcessRequest">
  <xpdl:part name="root" element="Address:initProcessRequest"/>
</xpdl:message>
<xpdl:message name="initProcessResponse">
  <xpdl:part name="root" element="Address:initProcessResponse"/>
</xpdl:message>
<!-- Messages -->
<xpdl:message name="notifyRequest">
  <xpdl:part name="root" element="Address:notifyRequest"/>
</xpdl:message>
<xpdl:message name="notifyResponse">
  <xpdl:part name="root" element="Address:notifyResponse"/>
</xpdl:message>

<!-- Operations provided by the processes interacting with workflow.
no port and binding for those as they are provided by the process. -->
<xpdl:portType name="Process">
  <xpdl:operation name="initProcess">
    <xpdl:input message="Address:initProcessRequest" name="initProcessRequest"/>
    <xpdl:output message="Address:initProcessResponse" name="initProcessResponse"/>
  </xpdl:operation>
  <xpdl:operation name="notifyTaskCompletion">
    <xpdl:input message="Address:notifyTaskCompletionRequest" name="notifyTaskCompletionRequest"/>
    <xpdl:output message="Address:notifyTaskCompletionResponse"
name="notifyTaskCompletionResponse"/>
  </xpdl:operation>
</xpdl:portType>

<!-- operation provided by the workflow -->
<xpdl:portType name="Workflow">
  <xpdl:operation name="createTask">
    <xpdl:input message="Address:createTaskRequest" name="createTaskRequest"/>
    <xpdl:output message="Address:createTaskResponse" name="createTaskResponse"/>
  </xpdl:operation>
  <xpdl:operation name="escalate">
    <xpdl:input message="Address:escalateRequest" name="escalateRequest"/>
    <xpdl:output message="Address:escalateResponse" name="escalateResponse"/>
  </xpdl:operation>
  <xpdl:operation name="notify">
    <xpdl:input message="Address:notifyRequest" name="notifyRequest"/>
    <xpdl:output message="Address:notifyResponse" name="notifyResponse"/>
  </xpdl:operation>

```

```

</xpd:portType>

<xpd:binding name="WorkflowSoapBinding" type="Address:Workflow">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <xpd:operation name="createTask">
    <soap:operation soapAction="createTask" style="document"/>
    <xpd:input><soap:body use="literal"/></xpd:input>
    <xpd:output><soap:body use="literal"/></xpd:output>
  </xpd:operation>
  <xpd:operation name="escalate">
    <soap:operation soapAction="escalate" style="document"/>
    <xpd:input><soap:body use="literal"/></xpd:input>
    <xpd:output><soap:body use="literal"/></xpd:output>
  </xpd:operation>
  <xpd:operation name="notify">
    <soap:operation soapAction="notify" style="document"/>
    <xpd:input><soap:body use="literal"/></xpd:input>
    <xpd:output><soap:body use="literal"/></xpd:output>
  </xpd:operation>
</xpd:binding>

<xpd:service name="WorkflowSoapService">
  <xpd:port name="AddressPort" binding="Address:WorkflowSoapBinding">
    <soap:address location="http://localhost:8080/fds/workflow/Address"/>
  </xpd:port>
</xpd:service>
</xpd:definitions>

```

A.2 Файл Payment.xform.all.xpd

```

<?xml version="1.0" encoding="utf-8" ?>
<xpd:definitions
targetNamespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment"
  xmlns:Payment="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment"
  xmlns:xpd="http://schemas.xmlsoap.org/xpd/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/xpd/soap/">

  <xpd:types>
    <!-- declare the namespace prefix again just to make sure
    this particular fragment is valid stand-alone schema. -->
    <xsd:schema targetNamespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment"
      xmlns:Payment="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified">

      <xsd:include schemaLocation="Payment.xform.xsd"/>
      <!-- the xpd message elements -->
      <xsd:element name="initProcessRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="taskId" type="xsd:string"/>
            <xsd:element name="participantToken" type="xsd:string"/>
            <xsd:element name="taskOutput">

```

```

        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Payment:output"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="initProcessResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="status" type="xsd:string"/>
            <xsd:element name="errorCode" type="xsd:string" minOccurs="0"/>
            <xsd:element name="errorReason" type="xsd:string" minOccurs="0"/>
            <xsd:element name="taskMetaData" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="nextTaskId" type="xsd:string"/>
                        <xsd:element name="nextTaskURL" type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="createTaskRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="taskMetaData" type="Payment:taskMetaDataType"/>
            <xsd:element name="participantToken" type="xsd:string"/>
            <xsd:element name="taskInput">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="Payment:input"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="response">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="isChainedAfter" type="xsd:boolean" minOccurs="0"/>
            <xsd:element name="taskMetaData" type="Payment:taskMetaDataType"/>
            <xsd:element name="status" type="xsd:string"/>
            <xsd:element name="errorCode" type="xsd:string" minOccurs="0"/>
            <xsd:element name="errorReason" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

```

<xsd:complexType name="accessControlType">
  <xsd:sequence>
    <xsd:element name="user" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="role" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="scheduledActionsDetails">
  <xsd:choice minOccurs="0" maxOccurs="1">
    <xsd:element name="until" type="xsd:dateTime" maxOccurs="1" minOccurs="0" />
    <xsd:element name="for" type="xsd:duration" maxOccurs="1"
minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="taskMetaDataType">
  <xsd:sequence>
    <xsd:element name="taskId" type="xsd:string"/>
    <xsd:element name="taskState" type="xsd:token"/>
    <xsd:element name="taskType" type="xsd:token"/>
    <xsd:element name="description" type="xsd:string"/>
    <xsd:element name="processId" type="xsd:string"/>
    <xsd:element name="creationDate" type="xsd:dateTime"/>
    <xsd:element name="userOwner" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
    <xsd:element name="roleOwner" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>

    <xsd:element name="claimAction" type="Payment:accessControlType" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="revokeAction" type="Payment:accessControlType" maxOccurs="1"
minOccurs="0"/>
    <xsd:element name="saveAction" type="Payment:accessControlType" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="completeAction" type="Payment:accessControlType" maxOccurs="1"
minOccurs="0"/>

    <xsd:element name="formUrl" type="xsd:anyURI"/>
    <xsd:element name="failureCode" type="xsd:string"/>
    <xsd:element name="failureReason" type="xsd:string"/>
    <xsd:element name="priority" type="xsd:int"/>
    <xsd:element name="scheduledActions">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="expiration" type="Payment:scheduledActionsDetails"/>
          <xsd:element name="deferActivation"
type="Payment:scheduledActionsDetails"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="userProcessCompleteSOAPAction" type="xsd:string"/>
    <xsd:element name="isChainedBefore" type="xsd:boolean"/>
    <xsd:element name="previousTaskId" type="xsd:string"/>
    <xsd:element name="userProcessEndpoint" type="xsd:string"/>
    <xsd:element name="userProcessNamespaceURI" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="notifyTaskCompletionRequest">

```

```

<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="taskMetaData" type="Payment:taskMetaDataType"/>
    <xsd:element name="taskOutput">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="Payment:output"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="status" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="notifyRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="metadata">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="creationDate" type="xsd:dateTime" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="userOwner" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="roleOwner" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="formUrl" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="input">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="Payment:input"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="notifyResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="status" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="errorCode" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="errorReason" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="escalateRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="taskId" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="userOwner" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>

```

```

        <xsd:element name="roleOwner" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="escalateResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="status" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

    <!-- end of the xpdL message elements -->
</xsd:schema>
</xpdL:types>

<!-- Messages Create -->
<xpdL:message name="createTaskRequest">
    <xpdL:part name="root" element="Payment:createTaskRequest"/>
</xpdL:message>
<xpdL:message name="createTaskResponse">
    <xpdL:part name="root" element="Payment:response"/>
</xpdL:message>
<!-- Messages Complete -->
<xpdL:message name="notifyTaskCompletionRequest">
    <xpdL:part name="root" element="Payment:notifyTaskCompletionRequest"/>
</xpdL:message>
<xpdL:message name="notifyTaskCompletionResponse">
    <xpdL:part name="root" element="Payment:response"/>
</xpdL:message>
<!-- Messages Escalate-->
<xpdL:message name="escalateRequest">
    <xpdL:part name="root" element="Payment:escalateRequest"/>
</xpdL:message>
<xpdL:message name="escalateResponse">
    <xpdL:part name="root" element="Payment:escalateResponse"/>
</xpdL:message>
<!-- Messages Init -->
<xpdL:message name="initProcessRequest">
    <xpdL:part name="root" element="Payment:initProcessRequest"/>
</xpdL:message>
<xpdL:message name="initProcessResponse">
    <xpdL:part name="root" element="Payment:initProcessResponse"/>
</xpdL:message>
<!-- Messages -->
<xpdL:message name="notifyRequest">
    <xpdL:part name="root" element="Payment:notifyRequest"/>
</xpdL:message>
<xpdL:message name="notifyResponse">
    <xpdL:part name="root" element="Payment:notifyResponse"/>
</xpdL:message>

```

```

<!-- Operations provided by the processes interacting with workflow.
no port and binding for those as they are provided by the process. -->
<xpdl:portType name="Process">
  <xpdl:operation name="initProcess">
    <xpdl:input message="Payment:initProcessRequest" name="initProcessRequest"/>
    <xpdl:output message="Payment:initProcessResponse" name="initProcessResponse"/>
  </xpdl:operation>
  <xpdl:operation name="notifyTaskCompletion">
    <xpdl:input message="Payment:notifyTaskCompletionRequest" name="notifyTaskCompletionRequest"/>
    <xpdl:output message="Payment:notifyTaskCompletionResponse"
name="notifyTaskCompletionResponse"/>
  </xpdl:operation>
</xpdl:portType>

<!-- operation provided by the workflow -->
<xpdl:portType name="Workflow">
  <xpdl:operation name="createTask">
    <xpdl:input message="Payment:createTaskRequest" name="createTaskRequest"/>
    <xpdl:output message="Payment:createTaskResponse" name="createTaskResponse"/>
  </xpdl:operation>
  <xpdl:operation name="escalate">
    <xpdl:input message="Payment:escalateRequest" name="escalateRequest"/>
    <xpdl:output message="Payment:escalateResponse" name="escalateResponse"/>
  </xpdl:operation>
  <xpdl:operation name="notify">
    <xpdl:input message="Payment:notifyRequest" name="notifyRequest"/>
    <xpdl:output message="Payment:notifyResponse" name="notifyResponse"/>
  </xpdl:operation>
</xpdl:portType>

<xpdl:binding name="WorkflowSoapBinding" type="Payment:Workflow">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <xpdl:operation name="createTask">
    <soap:operation soapAction="createTask" style="document"/>
    <xpdl:input><soap:body use="literal"/></xpdl:input>
    <xpdl:output><soap:body use="literal"/></xpdl:output>
  </xpdl:operation>
  <xpdl:operation name="escalate">
    <soap:operation soapAction="escalate" style="document"/>
    <xpdl:input><soap:body use="literal"/></xpdl:input>
    <xpdl:output><soap:body use="literal"/></xpdl:output>
  </xpdl:operation>
  <xpdl:operation name="notify">
    <soap:operation soapAction="notify" style="document"/>
    <xpdl:input><soap:body use="literal"/></xpdl:input>
    <xpdl:output><soap:body use="literal"/></xpdl:output>
  </xpdl:operation>
</xpdl:binding>

<xpdl:service name="WorkflowSoapService">
  <xpdl:port name="PaymentPort" binding="Payment:WorkflowSoapBinding">
    <soap:address location="http://localhost:8080/fds/workflow/Payment"/>
  </xpdl:port>

```

```
</xpdل:service>
```

```
</xpdل:definitions>
```

A.3 Файл SelectItem.xform.all.xpdл

```
<?xml version="1.0" encoding="utf-8" ?>
<xpdл:definitions
targetNamespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem"
  xmlns:SelectItem="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem"
  xmlns:xpdл="http://schemas.xmlsoap.org/xpdл/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/xpdл/soap/">

  <xpdл:types>
    <!-- declare the namespace prefix again just to make sure
    this particular fragment is valid stand-alone schema. -->
    <xsd:schema
targetNamespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem"

  xmlns:SelectItem="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
    <xsd:include schemaLocation="SelectItem.xform.xsd"/>
    <!-- the xpdл message elements -->
    <xsd:element name="initProcessRequest">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="taskId" type="xsd:string"/>
          <xsd:element name="participantToken" type="xsd:string"/>
          <xsd:element name="taskOutput">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element ref="SelectItem:output"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="initProcessResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="status" type="xsd:string"/>
          <xsd:element name="errorCode" type="xsd:string" minOccurs="0"/>
          <xsd:element name="errorReason" type="xsd:string" minOccurs="0"/>
          <xsd:element name="taskMetadata" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="nextTaskId" type="xsd:string"/>
                <xsd:element name="nextTaskURL" type="xsd:string"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>

```

```

        </xsd:complexType>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="createTaskRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="taskMetaData" type="SelectItem:taskMetaDataType"/>
            <xsd:element name="participantToken" type="xsd:string"/>
            <xsd:element name="taskInput">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="SelectItem:input"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="response">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="isChainedAfter" type="xsd:boolean" minOccurs="0"/>
            <xsd:element name="taskMetaData" type="SelectItem:taskMetaDataType"/>
            <xsd:element name="status" type="xsd:string"/>
            <xsd:element name="errorCode" type="xsd:string" minOccurs="0"/>
            <xsd:element name="errorReason" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:complexType name="accessControlType">
    <xsd:sequence>
        <xsd:element name="user" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="role" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="scheduledActionsDetails">
    <xsd:choice minOccurs="0" maxOccurs="1">
        <xsd:element name="until" type="xsd:dateTime" maxOccurs="1" minOccurs="0" />
        <xsd:element name="for" type="xsd:duration" maxOccurs="1"
minOccurs="0" />
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="taskMetaDataType">
    <xsd:sequence>
        <xsd:element name="taskId" type="xsd:string"/>
        <xsd:element name="taskState" type="xsd:token"/>
        <xsd:element name="taskType" type="xsd:token"/>
        <xsd:element name="description" type="xsd:string"/>
        <xsd:element name="processId" type="xsd:string"/>
        <xsd:element name="creationDate" type="xsd:dateTime"/>
    </xsd:sequence>
</xsd:complexType>

```

```

<xsd:element name="userOwner" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
<xsd:element name="roleOwner" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>

<xsd:element name="claimAction" type="SelectedItem:accessControlType" maxOccurs="1"
minOccurs="0"/>
<xsd:element name="revokeAction" type="SelectedItem:accessControlType" maxOccurs="1"
minOccurs="0"/>
<xsd:element name="saveAction" type="SelectedItem:accessControlType" maxOccurs="1"
minOccurs="0"/>
<xsd:element name="completeAction" type="SelectedItem:accessControlType" maxOccurs="1"
minOccurs="0"/>

<xsd:element name="formUrl" type="xsd:anyURI"/>
<xsd:element name="failureCode" type="xsd:string"/>
<xsd:element name="failureReason" type="xsd:string"/>
<xsd:element name="priority" type="xsd:int"/>
<xsd:element name="scheduledActions">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="expiration"
type="SelectedItem:scheduledActionsDetails"/>
      <xsd:element name="deferActivation"
type="SelectedItem:scheduledActionsDetails"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="userProcessCompleteSOAPAction" type="xsd:string"/>
<xsd:element name="isChainedBefore" type="xsd:boolean"/>
<xsd:element name="previousTaskId" type="xsd:string"/>
<xsd:element name="userProcessEndpoint" type="xsd:string"/>
<xsd:element name="userProcessNamespaceURI" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>

<xsd:element name="notifyTaskCompletionRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="taskMetaData" type="SelectedItem:taskMetaDataType"/>
      <xsd:element name="taskOutput">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="SelectedItem:output"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="status" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="notifyRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="metadata">

```

```

    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="creationDate" type="xsd:dateTime" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="userOwner" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="roleOwner" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="formUrl" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="input">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="SelectItem:input"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="notifyResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="status" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="errorCode" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="errorReason" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="escalateRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="taskId" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="userOwner" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element name="roleOwner" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="escalateResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="status" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

  <!-- end of the xpd1 message elements -->
</xsd:schema>
</xpd1:types>

<!-- Messages Create -->

```

```

<xpdl:message name="createTaskRequest">
  <xpdl:part name="root" element="SelectedItem:createTaskRequest"/>
</xpdl:message>
<xpdl:message name="createTaskResponse">
  <xpdl:part name="root" element="SelectedItem:response"/>
</xpdl:message>
<!-- Messages Complete -->
<xpdl:message name="notifyTaskCompletionRequest">
  <xpdl:part name="root" element="SelectedItem:notifyTaskCompletionRequest"/>
</xpdl:message>
<xpdl:message name="notifyTaskCompletionResponse">
  <xpdl:part name="root" element="SelectedItem:response"/>
</xpdl:message>
<!-- Messages Escalate-->
<xpdl:message name="escalateRequest">
  <xpdl:part name="root" element="SelectedItem:escalateRequest"/>
</xpdl:message>
<xpdl:message name="escalateResponse">
  <xpdl:part name="root" element="SelectedItem:escalateResponse"/>
</xpdl:message>
<!-- Messages Init -->
<xpdl:message name="initProcessRequest">
  <xpdl:part name="root" element="SelectedItem:initProcessRequest"/>
</xpdl:message>
<xpdl:message name="initProcessResponse">
  <xpdl:part name="root" element="SelectedItem:initProcessResponse"/>
</xpdl:message>
<!-- Messages -->
<xpdl:message name="notifyRequest">
  <xpdl:part name="root" element="SelectedItem:notifyRequest"/>
</xpdl:message>
<xpdl:message name="notifyResponse">
  <xpdl:part name="root" element="SelectedItem:notifyResponse"/>
</xpdl:message>

<!-- Operations provided by the processes interacting with workflow.
no port and binding for those as they are provided by the process. -->
<xpdl:portType name="Process">
  <xpdl:operation name="initProcess">
    <xpdl:input message="SelectedItem:initProcessRequest" name="initProcessRequest"/>
    <xpdl:output message="SelectedItem:initProcessResponse" name="initProcessResponse"/>
  </xpdl:operation>
  <xpdl:operation name="notifyTaskCompletion">
    <xpdl:input message="SelectedItem:notifyTaskCompletionRequest"
name="notifyTaskCompletionRequest"/>
    <xpdl:output message="SelectedItem:notifyTaskCompletionResponse"
name="notifyTaskCompletionResponse"/>
  </xpdl:operation>
</xpdl:portType>

<!-- operation provided by the workflow -->
<xpdl:portType name="Workflow">
  <xpdl:operation name="createTask">

```

```

    <xpdl:input message="SelectedItem:createTaskRequest" name="createTaskRequest"/>
    <xpdl:output message="SelectedItem:createTaskResponse" name="createTaskResponse"/>
  </xpdl:operation>
  <xpdl:operation name="escalate">
    <xpdl:input message="SelectedItem:escalateRequest" name="escalateRequest"/>
    <xpdl:output message="SelectedItem:escalateResponse" name="escalateResponse"/>
  </xpdl:operation>
  <xpdl:operation name="notify">
    <xpdl:input message="SelectedItem:notifyRequest" name="notifyRequest"/>
    <xpdl:output message="SelectedItem:notifyResponse" name="notifyResponse"/>
  </xpdl:operation>
</xpdl:portType>

<xpdl:binding name="WorkflowSoapBinding" type="SelectedItem:Workflow">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <xpdl:operation name="createTask">
    <soap:operation soapAction="createTask" style="document"/>
    <xpdl:input><soap:body use="literal"/></xpdl:input>
    <xpdl:output><soap:body use="literal"/></xpdl:output>
  </xpdl:operation>
  <xpdl:operation name="escalate">
    <soap:operation soapAction="escalate" style="document"/>
    <xpdl:input><soap:body use="literal"/></xpdl:input>
    <xpdl:output><soap:body use="literal"/></xpdl:output>
  </xpdl:operation>
  <xpdl:operation name="notify">
    <soap:operation soapAction="notify" style="document"/>
    <xpdl:input><soap:body use="literal"/></xpdl:input>
    <xpdl:output><soap:body use="literal"/></xpdl:output>
  </xpdl:operation>
</xpdl:binding>

<xpdl:service name="WorkflowSoapService">
  <xpdl:port name="SelectedItemPort" binding="SelectedItem:WorkflowSoapBinding">
    <soap:address location="http://localhost:8080/fds/workflow/SelectedItem"/>
  </xpdl:port>
</xpdl:service>
</xpdl:definitions>

```

A.4 Файл Store-Store.xform.all.xpdl

```

<?xml version='1.0' encoding='utf-8'?>
<xpdl:definitions xmlns:User="http://example.com/Store/User"
xmlns:xpdl="http://schemas.xmlsoap.org/xpdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:Payment="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment/attachments"
xmlns:Payment1="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment"
xmlns:SelectedItem="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectedItem/attachments"
xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
xmlns:diag="http://example.com/Store"

```

```

xmlns:Address1="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address"
xmlns:Address="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address/attachments"
xmlns:vprop="http://docs.oasis-open.org/wsbpel/2.0/varprop"
xmlns:soap="http://schemas.xmlsoap.org/xpdl/soap/"
xmlns:SelectItem1="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem"
xmlns:this="http://example.com/Store/Store" xmlns:plnk="http://docs.oasis-
open.org/wsbpel/2.0/plnktype" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
targetNamespace="http://example.com/Store/Store">
    <xpdl:import namespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address"
location="Address.xform.all.xpdl"/>
    <xpdl:import namespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment"
location="Payment.xform.all.xpdl"/>
    <xpdl:import namespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem"
location="SelectItem.xform.all.xpdl"/>
    <xpdl:binding name="WfMagic_QwWOGNDEd2hi4Tx6BbOrAServiceBinding" type="Address1:Process">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <xpdl:operation name="initProcess">
            <soap:operation style="document"
soapAction="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address/Process/initProcess"/
>
            <xpdl:input name="initProcessRequest">
                <soap:body use="literal"/>
            </xpdl:input>
            <xpdl:output name="initProcessResponse">
                <soap:body use="literal"/>
            </xpdl:output>
        </xpdl:operation>
        <xpdl:operation name="notifyTaskCompletion">
            <soap:operation style="document"
soapAction="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address/Process/notifyTaskCom
pletion"/>
            <xpdl:input name="notifyTaskCompletionRequest">
                <soap:body use="literal"/>
            </xpdl:input>
            <xpdl:output name="notifyTaskCompletionResponse">
                <soap:body use="literal"/>
            </xpdl:output>
        </xpdl:operation>
    </xpdl:binding>
    <xpdl:binding name="WfMagic_RlVgONNDEd2hi4Tx6BbOrAServiceBinding" type="Payment1:Process">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <xpdl:operation name="initProcess">
            <soap:operation style="document"
soapAction="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment/Process/initProcess"/
>
            <xpdl:input name="initProcessRequest">
                <soap:body use="literal"/>
            </xpdl:input>
            <xpdl:output name="initProcessResponse">
                <soap:body use="literal"/>
            </xpdl:output>
        </xpdl:operation>
        <xpdl:operation name="notifyTaskCompletion">

```

```

        <soap:operation style="document"
soapAction="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment/Process/notifyTaskCom
pletion"/>
        <xpdl:input name="notifyTaskCompletionRequest">
            <soap:body use="literal"/>
        </xpdl:input>
        <xpdl:output name="notifyTaskCompletionResponse">
            <soap:body use="literal"/>
        </xpdl:output>
    </xpdl:operation>
</xpdl:binding>
<xpdl:binding name="StoreAndUserForPortTypeProcessPlkServiceBinding" type="SelectItem1:Process">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <xpdl:operation name="initProcess">
        <soap:operation style="document"
soapAction="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem/Process/initProces
s"/>
        <xpdl:input name="initProcessRequest">
            <soap:body use="literal"/>
        </xpdl:input>
        <xpdl:output name="initProcessResponse">
            <soap:body use="literal"/>
        </xpdl:output>
    </xpdl:operation>
    <xpdl:operation name="notifyTaskCompletion">
        <soap:operation style="document"
soapAction="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem/Process/notifyTask
Completion"/>
        <xpdl:input name="notifyTaskCompletionRequest">
            <soap:body use="literal"/>
        </xpdl:input>
        <xpdl:output name="notifyTaskCompletionResponse">
            <soap:body use="literal"/>
        </xpdl:output>
    </xpdl:operation>
</xpdl:binding>
<xpdl:service name="WFmagic_QwWOGNNEd2hi4Tx6BbOrAService">
    <xpdl:port name="Store_for_UserForThePortTypeProcessPort"
binding="this:WFmagic_QwWOGNNEd2hi4Tx6BbOrAServiceBinding">
        <soap:address
location="http://localhost:8080/ode/processes/ChainedExecution/Store/Store/User/WFmagic_QwWOGNNEd2h
i4Tx6BbOrA"/>
    </xpdl:port>
</xpdl:service>
<xpdl:service name="WFmagic_R1VgONNDEd2hi4Tx6BbOrAService">
    <xpdl:port name="Store_for_UserForThePortTypeProcessPort"
binding="this:WFmagic_R1VgONNDEd2hi4Tx6BbOrAServiceBinding">
        <soap:address
location="http://localhost:8080/ode/processes/ChainedExecution/Store/Store/User/WFmagic_R1VgONNDEd2h
i4Tx6BbOrA"/>
    </xpdl:port>
</xpdl:service>
<xpdl:service name="StoreAndUserForPortTypeProcessPlkService">

```

```

        <xpdl:port name="Store_for_UserPort"
binding="this:StoreAndUserForPortTypeProcessPlkServiceBinding">
        <soap:address
location="http://localhost:8080/ode/processes/ChainedExecution/Store/Store/User/StoreAndUserForPortT
ypeProcessPlk"/>
        </xpdl:port>
    </xpdl:service>
</xpdl:definitions>

```

A.5 Файл Store.bpel

```

<?xml version='1.0' encoding='UTF-8'?>
<bpel:process xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:vprop="http://docs.oasis-open.org/wsbpel/2.0/varprop" xmlns:plnk="http://docs.oasis-
open.org/wsbpel/2.0/plnktype" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:Payment="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment/attachments"
xmlns:SelectItem="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem/attachments"
xmlns:Address="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address/attachments"
xmlns:User="http://example.com/Store/User"
xmlns:Payment1="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment"
xmlns:this="http://example.com/Store/Store"
xmlns:Address1="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address"
xmlns:SelectItem1="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem"
xmlns:diag="http://example.com/Store" xmlns:bpmn="http://www.intalio.com/bpms"
xmlns:atomic="http://ode.apache.org/atomicScope"
queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0"
expressionLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0" bpmn:label="Store"
bpmn:id="_KwMpgNNDed2hi4Tx6BbOrA" name="Store" targetNamespace="http://example.com/Store/Store">
    <bpel:import namespace="http://example.com/Store" location="Store.wsdl"
importType="http://schemas.xmlsoap.org/wsdl/"></bpel:import>
    <bpel:import namespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address"
location="Address.xform.all.wsdl" importType="http://schemas.xmlsoap.org/wsdl/"></bpel:import>
    <bpel:import namespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment"
location="Payment.xform.all.wsdl" importType="http://schemas.xmlsoap.org/wsdl/"></bpel:import>
    <bpel:import namespace="http://www.intalio.com/bpms/workflow/xforms/Examples/Store/SelectItem"
location="SelectItem.xform.all.wsdl" importType="http://schemas.xmlsoap.org/wsdl/"></bpel:import>
    <bpel:partnerLinks>
        <bpel:partnerLink name="wFmagic_QwWOGNNDed2hi4Tx6BbOrAPlkVar"
partnerLinkType="diag:WFMagic_QwWOGNNDed2hi4Tx6BbOrA" initializePartnerRole="false"
myRole="Store_for_UserForThePortTypeProcess"
partnerRole="User_for_StoreForAddressPort"></bpel:partnerLink>
        <bpel:partnerLink name="wFmagic_R1VgONNDed2hi4Tx6BbOrAPlkVar"
partnerLinkType="diag:WFMagic_R1VgONNDed2hi4Tx6BbOrA" initializePartnerRole="false"
myRole="Store_for_UserForThePortTypeProcess"
partnerRole="User_for_StoreForPaymentPort"></bpel:partnerLink>
        <bpel:partnerLink name="storeAndUserForPortTypeProcessPlkVar"
partnerLinkType="diag:StoreAndUserForPortTypeProcessPlk" myRole="Store_for_User"></bpel:partnerLink>
    </bpel:partnerLinks>
    <bpel:variables>

```

```

        <bpel:variable name="selectItem1InitProcessRequestMsg"
messageType="SelectItem1:initProcessRequest"></bpel:variable>
        <bpel:variable name="address1CreateTaskRequestMsg"
messageType="Address1:createTaskRequest"></bpel:variable>
        <bpel:variable name="address1CreateTaskResponseMsg"
messageType="Address1:createTaskResponse"></bpel:variable>
        <bpel:variable name="selectItem1InitProcessResponseMsg"
messageType="SelectItem1:initProcessResponse"></bpel:variable>
        <bpel:variable name="address1NotifyTaskCompletionRequestMsg"
messageType="Address1:notifyTaskCompletionRequest"></bpel:variable>
        <bpel:variable name="address1NotifyTaskCompletionResponseMsg"
messageType="Address1:notifyTaskCompletionResponse"></bpel:variable>
        <bpel:variable name="payment1CreateTaskRequestMsg"
messageType="Payment1:createTaskRequest"></bpel:variable>
        <bpel:variable name="payment1CreateTaskResponseMsg"
messageType="Payment1:createTaskResponse"></bpel:variable>
        <bpel:variable name="payment1NotifyTaskCompletionRequestMsg"
messageType="Payment1:notifyTaskCompletionRequest"></bpel:variable>
        <bpel:variable name="payment1NotifyTaskCompletionResponseMsg"
messageType="Payment1:notifyTaskCompletionResponse"></bpel:variable>
    </bpel:variables>
    <bpel:sequence>
        <bpel:receive partnerLink="storeAndUserForPortTypeProcessPlkVar" portType="SelectItem1:Process"
operation="initProcess" variable="selectItem1InitProcessRequestMsg" createInstance="yes"
bpmn:label="Receive product Request" bpmn:id="_UY9TgNNDEd2hi4Tx6BbOrA"></bpel:receive>
        <bpel:assign name="init-variables-Store">
            <bpel:copy bpmn:label="$address1CreateTaskRequestMsg">
                <bpel:from>
                    <bpel:literal>
<Address1:createTaskRequest>
    <Address1:taskMetaData>
        <Address1:taskId>_QwWOGNDEd2hi4Tx6BbOrA</Address1:taskId>
        <Address1:taskState></Address1:taskState>
        <Address1:taskType></Address1:taskType>
        <Address1:description />
        <Address1:processId></Address1:processId>
        <Address1:creationDate></Address1:creationDate>
        <Address1:userOwner>intalio/admin</Address1:userOwner>
        <Address1:roleOwner />
        <Address1:claimAction>
            <Address1:user></Address1:user>
            <Address1:role></Address1:role>
        </Address1:claimAction>
        <Address1:revokeAction>
            <Address1:user></Address1:user>
            <Address1:role></Address1:role>
        </Address1:revokeAction>
        <Address1:saveAction>
            <Address1:user></Address1:user>
            <Address1:role></Address1:role>
        </Address1:saveAction>
        <Address1:completeAction>
            <Address1:user></Address1:user>

```

```

    <Address1:role></Address1:role>
  </Address1:completeAction>
  <Address1:formUrl>oxf://ChainedExecution/Address.xform</Address1:formUrl>
  <Address1:failureCode></Address1:failureCode>
  <Address1:failureReason></Address1:failureReason>
  <Address1:priority></Address1:priority>
  <Address1:scheduledActions>
    <Address1:expiration>
      <Address1:until></Address1:until>
      <Address1:for></Address1:for>
    </Address1:expiration>
    <Address1:deferActivation>
      <Address1:until></Address1:until>
      <Address1:for></Address1:for>
    </Address1:deferActivation>
  </Address1:scheduledActions>

  <Address1:userProcessCompleteSOAPAction>http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address/Process/notifyTaskCompletion</Address1:userProcessCompleteSOAPAction>
  <Address1:isChainedBefore></Address1:isChainedBefore>
  <Address1:previousTaskId></Address1:previousTaskId>

  <Address1:userProcessEndpoint>http://localhost:8080/ode/processes/ChainedExecution/Store/Store/User/WFmagic_QwWOGNNDEd2hi4Tx6BbOrA</Address1:userProcessEndpoint>

  <Address1:userProcessNamespaceURI>http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address</Address1:userProcessNamespaceURI>
  </Address1:taskMetaData>
  <Address1:participantToken></Address1:participantToken>
  <Address1:taskInput>
    <Address1:input>
      <Address1:Order>
        <Address1:reference></Address1:reference>
      </Address1:Order>
    </Address1:input>
  </Address1:taskInput>
  </Address1:createTaskRequest></bpel:literal>
    </bpel:from>
    <bpel:to>$address1CreateTaskRequestMsg.root</bpel:to>
  </bpel:copy>
  <bpel:copy bpmn:label="$selectItem1InitProcessResponseMsg">
    <bpel:from>
      <bpel:literal>
        <SelectItem1:initProcessResponse>
          <SelectItem1:status>OK</SelectItem1:status>
          <SelectItem1:errorCode></SelectItem1:errorCode>
          <SelectItem1:errorReason></SelectItem1:errorReason>
          <SelectItem1:taskMetaData>
            <SelectItem1:nextTaskId></SelectItem1:nextTaskId>
            <SelectItem1:nextTaskURL></SelectItem1:nextTaskURL>
          </SelectItem1:taskMetaData>
        </SelectItem1:initProcessResponse></bpel:literal>
      </bpel:from>

```

```

        <bpel:to>$selectItem1InitProcessResponseMsg.root</bpel:to>
    </bpel:copy>
    <bpel:copy bpmn:label="$address1NotifyTaskCompletionResponseMsg">
        <bpel:from>
            <bpel:literal>
<Address1:response>
    <Address1:isChainedAfter></Address1:isChainedAfter>
    <Address1:taskMetaData>
        <Address1:taskId></Address1:taskId>
        <Address1:taskState></Address1:taskState>
        <Address1:taskType></Address1:taskType>
        <Address1:description></Address1:description>
        <Address1:processId></Address1:processId>
        <Address1:creationDate></Address1:creationDate>
        <Address1:userOwner></Address1:userOwner>
        <Address1:roleOwner></Address1:roleOwner>
        <Address1:claimAction>
            <Address1:user></Address1:user>
            <Address1:role></Address1:role>
        </Address1:claimAction>
        <Address1:revokeAction>
            <Address1:user></Address1:user>
            <Address1:role></Address1:role>
        </Address1:revokeAction>
        <Address1:saveAction>
            <Address1:user></Address1:user>
            <Address1:role></Address1:role>
        </Address1:saveAction>
        <Address1:completeAction>
            <Address1:user></Address1:user>
            <Address1:role></Address1:role>
        </Address1:completeAction>
        <Address1:formUrl></Address1:formUrl>
        <Address1:failureCode></Address1:failureCode>
        <Address1:failureReason></Address1:failureReason>
        <Address1:priority></Address1:priority>
        <Address1:scheduledActions>
            <Address1:expiration>
                <Address1:until></Address1:until>
                <Address1:for></Address1:for>
            </Address1:expiration>
            <Address1:deferActivation>
                <Address1:until></Address1:until>
                <Address1:for></Address1:for>
            </Address1:deferActivation>
        </Address1:scheduledActions>
        <Address1:userProcessCompleteSOAPAction></Address1:userProcessCompleteSOAPAction>
        <Address1:isChainedBefore></Address1:isChainedBefore>
        <Address1:previousTaskId></Address1:previousTaskId>
        <Address1:userProcessEndpoint></Address1:userProcessEndpoint>
        <Address1:userProcessNamespaceURI></Address1:userProcessNamespaceURI>
    </Address1:taskMetaData>
    <Address1:status>OK</Address1:status>

```

```

    <Address1:errorCode></Address1:errorCode>
    <Address1:errorReason></Address1:errorReason>
  </Address1:response></bpel:literal>
  </bpel:from>
  <bpel:to>$address1NotifyTaskCompletionResponseMsg.root</bpel:to>
</bpel:copy>
  <bpel:copy bpmn:label="$payment1CreateTaskRequestMsg">
    <bpel:from>
      <bpel:literal>
<Payment1:createTaskRequest>
  <Payment1:taskMetaData>
    <Payment1:taskId>_RlVgONNDEd2hi4Tx6BbOrA</Payment1:taskId>
    <Payment1:taskState></Payment1:taskState>
    <Payment1:taskType></Payment1:taskType>
    <Payment1:description />
    <Payment1:processId></Payment1:processId>
    <Payment1:creationDate></Payment1:creationDate>
    <Payment1:userOwner>intalio/admin</Payment1:userOwner>
    <Payment1:roleOwner />
    <Payment1:claimAction>
      <Payment1:user></Payment1:user>
      <Payment1:role></Payment1:role>
    </Payment1:claimAction>
    <Payment1:revokeAction>
      <Payment1:user></Payment1:user>
      <Payment1:role></Payment1:role>
    </Payment1:revokeAction>
    <Payment1:saveAction>
      <Payment1:user></Payment1:user>
      <Payment1:role></Payment1:role>
    </Payment1:saveAction>
    <Payment1:completeAction>
      <Payment1:user></Payment1:user>
      <Payment1:role></Payment1:role>
    </Payment1:completeAction>
    <Payment1:formUrl>oxf://ChainedExecution/Payment.xform</Payment1:formUrl>
    <Payment1:failureCode></Payment1:failureCode>
    <Payment1:failureReason></Payment1:failureReason>
    <Payment1:priority></Payment1:priority>
    <Payment1:scheduledActions>
      <Payment1:expiration>
        <Payment1:until></Payment1:until>
        <Payment1:for></Payment1:for>
      </Payment1:expiration>
      <Payment1:deferActivation>
        <Payment1:until></Payment1:until>
        <Payment1:for></Payment1:for>
      </Payment1:deferActivation>
    </Payment1:scheduledActions>

    <Payment1:userProcessCompleteSOAPAction>http://www.intalio.com/bpms/workflow/xforms/Examples/Store/P
ayment/Process/notifyTaskCompletion</Payment1:userProcessCompleteSOAPAction>
    <Payment1:isChainedBefore></Payment1:isChainedBefore>

```

```

    <Payment1:previousTaskId></Payment1:previousTaskId>

    <Payment1:userProcessEndpoint>http://localhost:8080/ode/processes/ChainedExecution/Store/Store/User/
    Wfmagic_R1VgONNDEd2hi4Tx6BbOrA</Payment1:userProcessEndpoint>

    <Payment1:userProcessNamespaceURI>http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment
    </Payment1:userProcessNamespaceURI>
    </Payment1:taskMetaData>
    <Payment1:participantToken></Payment1:participantToken>
    <Payment1:taskInput>
    <Payment1:input>
    <Payment1:address>
    <Payment1:country></Payment1:country>
    <Payment1:state></Payment1:state>
    <Payment1:zip></Payment1:zip>
    <Payment1:city></Payment1:city>
    <Payment1:street2></Payment1:street2>
    <Payment1:street1></Payment1:street1>
    </Payment1:address>
    <Payment1:Order>
    <Payment1:reference></Payment1:reference>
    </Payment1:Order>
    </Payment1:input>
    </Payment1:taskInput>
    </Payment1:createTaskRequest></bpel:literal>
    </bpel:from>
    <bpel:to>$payment1CreateTaskRequestMsg.root</bpel:to>
    </bpel:copy>
    <bpel:copy bpmn:label="$payment1NotifyTaskCompletionResponseMsg">
    <bpel:from>
    <bpel:literal>
    <Payment1:response>
    <Payment1:isChainedAfter></Payment1:isChainedAfter>
    <Payment1:taskMetaData>
    <Payment1:taskId></Payment1:taskId>
    <Payment1:taskState></Payment1:taskState>
    <Payment1:taskType></Payment1:taskType>
    <Payment1:description></Payment1:description>
    <Payment1:processId></Payment1:processId>
    <Payment1:creationDate></Payment1:creationDate>
    <Payment1:userOwner></Payment1:userOwner>
    <Payment1:roleOwner></Payment1:roleOwner>
    <Payment1:claimAction>
    <Payment1:user></Payment1:user>
    <Payment1:role></Payment1:role>
    </Payment1:claimAction>
    <Payment1:revokeAction>
    <Payment1:user></Payment1:user>
    <Payment1:role></Payment1:role>
    </Payment1:revokeAction>
    <Payment1:saveAction>
    <Payment1:user></Payment1:user>
    <Payment1:role></Payment1:role>

```

```

</Payment1:saveAction>
<Payment1:completeAction>
  <Payment1:user></Payment1:user>
  <Payment1:role></Payment1:role>
</Payment1:completeAction>
<Payment1:formUrl></Payment1:formUrl>
<Payment1:failureCode></Payment1:failureCode>
<Payment1:failureReason></Payment1:failureReason>
<Payment1:priority></Payment1:priority>
<Payment1:scheduledActions>
  <Payment1:expiration>
    <Payment1:until></Payment1:until>
    <Payment1:for></Payment1:for>
  </Payment1:expiration>
  <Payment1:deferActivation>
    <Payment1:until></Payment1:until>
    <Payment1:for></Payment1:for>
  </Payment1:deferActivation>
</Payment1:scheduledActions>
<Payment1:userProcessCompleteSOAPAction></Payment1:userProcessCompleteSOAPAction>
<Payment1:isChainedBefore></Payment1:isChainedBefore>
<Payment1:previousTaskId></Payment1:previousTaskId>
<Payment1:userProcessEndpoint></Payment1:userProcessEndpoint>
<Payment1:userProcessNamespaceURI></Payment1:userProcessNamespaceURI>
</Payment1:taskMetaData>
<Payment1:status>OK</Payment1:status>
<Payment1:errorCode></Payment1:errorCode>
<Payment1:errorReason></Payment1:errorReason>
</Payment1:response></bpel:literal>
  </bpel:from>
  <bpel:to>$payment1NotifyTaskCompletionResponseMsg.root</bpel:to>
</bpel:copy>
</bpel:assign>
<bpel:assign bpmn:label="Request Address" bpmn:id="_VIfTANNEd2hi4Tx6BbOrA">
  <bpel:copy>
    <bpel:from>$selectItem1InitProcessRequestMsg.root/SelectItem1:taskId</bpel:from>
    <bpel:to>$address1CreateTaskRequestMsg.root/Address1:taskMetaData/Address1:previousTaskId</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>"true"</bpel:from>
  </bpel:copy>
</bpel:copy>
<bpel:to>$address1CreateTaskRequestMsg.root/Address1:taskMetaData/Address1:isChainedBefore</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>$selectItem1InitProcessRequestMsg.root/SelectItem1:taskOutput/SelectItem1:output/SelectItem1:product</bpel:from>
  </bpel:copy>
</bpel:assign>
<bpel:assign>
  <bpel:from>$address1CreateTaskRequestMsg.root/Address1:taskInput/Address1:input/Address1:Order/Address1:reference</bpel:from>
  </bpel:copy>
</bpel:assign>

```

```

    <bpel:copy>

<bpel:from>"http://localhost:8080/ode/processes/ChainedExecution/Store/Store/User/WFmagic_QwWOGNNDEd
2hi4Tx6BbOrA"</bpel:from>

<bpel:to>$address1CreateTaskRequestMsg.root/Address1:taskMetaData/Address1:userProcessEndpoint</bpel
:to>
    </bpel:copy>
    <bpel:copy>

<bpel:from>"http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Address/Process/notifyTaskCom
pletion"</bpel:from>

<bpel:to>$address1CreateTaskRequestMsg.root/Address1:taskMetaData/Address1:userProcessCompleteSOAPAc
tion</bpel:to>
    </bpel:copy>
    <bpel:copy>
        <bpel:from>"_Qwf_ENNDEd2hi4Tx6BbOrA"</bpel:from>
        <bpel:to>$address1CreateTaskRequestMsg.root/Address1:taskMetaData/Address1:taskId</bpel:to>
    </bpel:copy>
</bpel:assign>
    <bpel:invoke partnerLink="wFmagic_QwWOGNNDEd2hi4Tx6BbOrAPlkVar" portType="Address1:Workflow"
operation="createTask" inputVariable="address1CreateTaskRequestMsg"
outputVariable="address1CreateTaskResponseMsg" bpmn:label="Request Address"
bpmn:id="_VIfTANNDEd2hi4Tx6BbOrA"></bpel:invoke>
    <bpel:assign bpmn:label="Acknowledge &#xd;&#xa;Product Request "
bpmn:id="_YCA0NNDEd2hi4Tx6BbOrA">
        <bpel:copy>

<bpel:from>$address1CreateTaskRequestMsg.root/Address1:taskMetaData/Address1:formUrl</bpel:from>

<bpel:to>$selectItem1InitProcessResponseMsg.root/SelectItem1:taskMetaData/SelectItem1:nextTaskURL</bp
el:to>
    </bpel:copy>
    <bpel:copy>

<bpel:from>$address1CreateTaskResponseMsg.root/Address1:taskMetaData/Address1:taskId</bpel:from>

<bpel:to>$selectItem1InitProcessResponseMsg.root/SelectItem1:taskMetaData/SelectItem1:nextTaskId</bp
el:to>
    </bpel:copy>
</bpel:assign>
    <bpel:reply partnerLink="storeAndUserForPortTypeProcessPlkVar" portType="SelectItem1:Process"
operation="initProcess" variable="selectItem1InitProcessResponseMsg" bpmn:label="Acknowledge
&#xd;&#xa;Product Request " bpmn:id="_YCA0NNDEd2hi4Tx6BbOrA"></bpel:reply>
    <bpel:receive partnerLink="wFmagic_QwWOGNNDEd2hi4Tx6BbOrAPlkVar" portType="Address1:Process"
operation="notifyTaskCompletion" variable="address1NotifyTaskCompletionRequestMsg"
bpmn:label="Receive Address" bpmn:id="_aVPEsNNDEd2hi4Tx6BbOrA"></bpel:receive>
    <bpel:assign bpmn:label="Receive Address" bpmn:id="_aVPEsNNDEd2hi4Tx6BbOrA">
        <bpel:copy>
            <bpel:from>"true"</bpel:from>
            <bpel:to>$address1NotifyTaskCompletionResponseMsg.root/Address1:isChainedAfter</bpel:to>
        </bpel:copy>

```

```

    </bpel:assign>
    <bpel:reply partnerLink="wFmagic_QwWOGNNDEd2hi4Tx6BbOrAPlkVar" portType="Address1:Process"
operation="notifyTaskCompletion" variable="address1NotifyTaskCompletionResponseMsg"
bpmn:label="Receive Address" bpmn:id="_aVPesNNDEd2hi4Tx6BbOrA"></bpel:reply>

    <bpel:assign bpmn:label="Request Payment" bpmn:id="_ifqvgNNDEd2hi4Tx6BbOrA">
        <bpel:copy>

<bpel:from>$address1NotifyTaskCompletionRequestMsg.root/Address1:taskOutput/Address1:output/Address1
:Order/Address1:reference</bpel:from>

<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskInput/Payment1:input/Payment1:Order/Payment
1:reference</bpel:to>
        </bpel:copy>
        <bpel:copy>

<bpel:from>$address1NotifyTaskCompletionRequestMsg.root/Address1:taskOutput/Address1:output/Address1
:Address/Address1:country</bpel:from>

<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskInput/Payment1:input/Payment1:address/Payme
nt1:country</bpel:to>
        </bpel:copy>
        <bpel:copy>

<bpel:from>$address1NotifyTaskCompletionRequestMsg.root/Address1:taskOutput/Address1:output/Address1
:Address/Address1:state</bpel:from>
<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskInput/Payment1:input/Payment1:address/Payme
nt1:state</bpel:to>
        </bpel:copy>
        <bpel:copy>

<bpel:from>$address1NotifyTaskCompletionRequestMsg.root/Address1:taskOutput/Address1:output/Address1
:Address/Address1:zip</bpel:from>

<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskInput/Payment1:input/Payment1:address/Payme
nt1:zip</bpel:to>
        </bpel:copy>
        <bpel:copy>

<bpel:from>$address1NotifyTaskCompletionRequestMsg.root/Address1:taskOutput/Address1:output/Address1
:Address/Address1:city</bpel:from>

<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskInput/Payment1:input/Payment1:address/Payme
nt1:city</bpel:to>
        </bpel:copy>
        <bpel:copy>

<bpel:from>$address1NotifyTaskCompletionRequestMsg.root/Address1:taskOutput/Address1:output/Address1
:Address/Address1:street2</bpel:from>

<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskInput/Payment1:input/Payment1:address/Payme
nt1:street2</bpel:to>
        </bpel:copy>
        <bpel:copy>

```

```

<bpel:from>$address1NotifyTaskCompletionRequestMsg.root/Address1:taskOutput/Address1:output/Address1:Address/Address1:street1</bpel:from>

<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskInput/Payment1:input/Payment1:address/Payment1:street1</bpel:to>
    </bpel:copy>
    <bpel:copy>

<bpel:from>$address1CreateTaskResponseMsg.root/Address1:taskMetaData/Address1:taskId</bpel:from>
<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskMetaData/Payment1:previousTaskId</bpel:to>
    </bpel:copy>
    <bpel:copy>
        <bpel:from>"true"</bpel:from>

<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskMetaData/Payment1:isChainedBefore</bpel:to>
    </bpel:copy>
</bpel:assign>
<bpel:assign>
    <bpel:copy>

<bpel:from>"http://localhost:8080/ode/processes/ChainedExecution/Store/Store/User/WFmagic_R1VgONNDEd2hi4Tx6BbOrA"</bpel:from>

<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskMetaData/Payment1:userProcessEndpoint</bpel:to>
    </bpel:copy>
    <bpel:copy>

<bpel:from>"http://www.intalio.com/bpms/workflow/xforms/Examples/Store/Payment/Process/notifyTaskCompletion"</bpel:from>

<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskMetaData/Payment1:userProcessCompleteSOAPAction</bpel:to>
    </bpel:copy>
    <bpel:copy>
        <bpel:from>"_R1fRNtNDEd2hi4Tx6BbOrA"</bpel:from>
<bpel:to>$payment1CreateTaskRequestMsg.root/Payment1:taskMetaData/Payment1:taskId</bpel:to>
    </bpel:copy>
</bpel:assign>
    <bpel:invoke partnerLink="wFmagic_R1VgONNDEd2hi4Tx6BbOrAPlkVar" portType="Payment1:Workflow"
operation="createTask" inputVariable="payment1CreateTaskRequestMsg"
outputVariable="payment1CreateTaskResponseMsg" bpmn:label="Request Payment"
bpmn:id="_ifqvgNDEd2hi4Tx6BbOrA"></bpel:invoke>
        <bpel:receive partnerLink="wFmagic_R1VgONNDEd2hi4Tx6BbOrAPlkVar" portType="Payment1:Process"
operation="notifyTaskCompletion" variable="payment1NotifyTaskCompletionRequestMsg"
bpmn:label="Payment Received" bpmn:id="_lkz7gNDEd2hi4Tx6BbOrA"></bpel:receive>
        <bpel:reply partnerLink="wFmagic_R1VgONNDEd2hi4Tx6BbOrAPlkVar" portType="Payment1:Process"
operation="notifyTaskCompletion" variable="payment1NotifyTaskCompletionResponseMsg"
bpmn:label="Payment Received" bpmn:id="_lkz7gNDEd2hi4Tx6BbOrA"></bpel:reply>
    </bpel:sequence>
</bpel:process>

```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

ДИПЛОМНА КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему

«ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ТА ЗАСОБІВ АВТОМАТИЗАЦІЇ БІЗНЕС- ПРОЦЕСІВ»

Виконав студент
групи КНТ – 121м

Куцак С.О.

Керівник
к.т.н., доц.

Гладкова О. М.

Рисунок Б.1 – Слайд 1



Рисунок Б.2 – Слайд 2

Огляд існуючих рішень



BPM-системи:

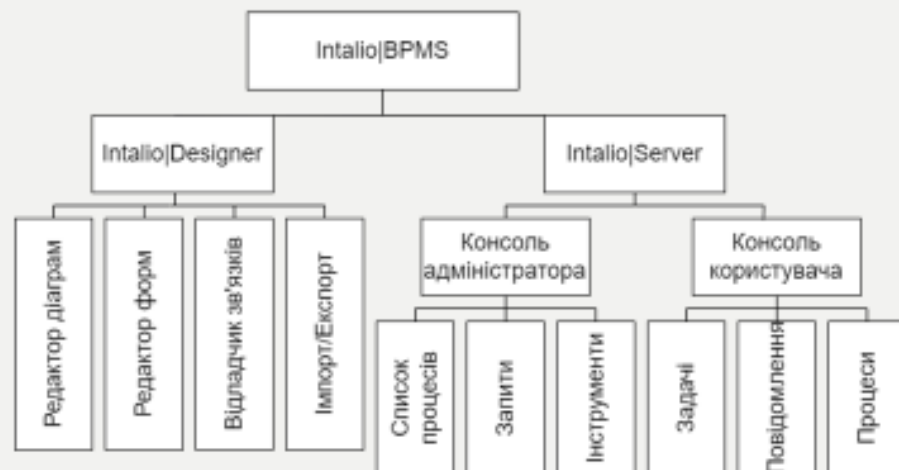
- ARIS Express,
- JBoss jBPM,
- RUNA WFE,
- BizAgi,
- Intalio|BPMS,
- Camunda,
- IBM WebSphere Business Modeler,
- Fox Manager BPA.

Узагальнена структурна схема BPM-системи

5

Рисунок Б.5 – Слайд 5

Особенности системы Intalio|BPMS

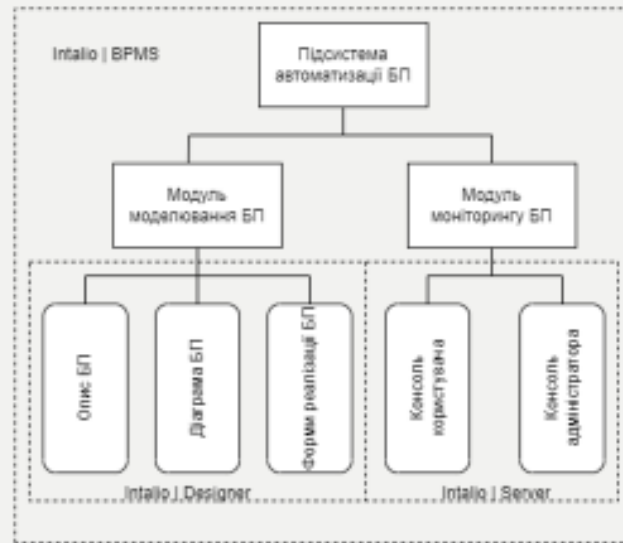


Структурна схема Intalio | BPMS

6

Рисунок Б.6 – Слайд 6

Розробка структури підсистеми автоматизації бізнес-процесів



7

Рисунок Б.7 – Слайд 7

Схема алгоритму моделювання БП в Intalio | Designer



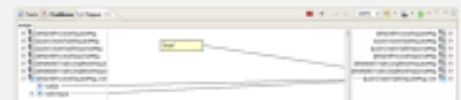
Палітра BPMN-нотації



Діаграма БП



Зв'язки дії



8

Рисунок Б.8 – Слайд 8

Розробка форм з використанням мови XPDЛ

Елемент	Що визначає
<portType>	Методи, що надаються вебсервісом
<message>	Повідомлення, що використовуються вебсервісом
<types>	Типи даних, які використовуються вебсервісом
<binding>	Протоколи зв'язку, які використовуються вебсервісом

Основні елементи
мови XPDЛ

Форма «Вибір товару»

Форма «Адреса доставки»

Форма «Оплата»

9

Рисунок Б.9 – Слайд 9

Розробка модулю моніторингу БП в Intalio | Server

Основні елементи BPEL

Елемент	Що визначає
<partners>	Опис партнерів (елемент, в який вкладено один або кілька елементів <partner>)
<partnerLinks>	Опис зв'язків між партнерами (елемент, в який вкладено один або кілька елементів <partnerLink>)
<variables>	Визначення змінних (елемент, в який вкладено один або кілька елементів <variable> або контейнерів <containers>)
<correlationSets>	Кореляційні набори (елемент, в який вкладено один або кілька елементів <correlationSet>)
<faultHandlers>	Обробники помилок
<eventHandlers>	Обробники подій
<compensationHandler>	Обробник компенсації



10

Рисунок Б.10 – Слайд 10

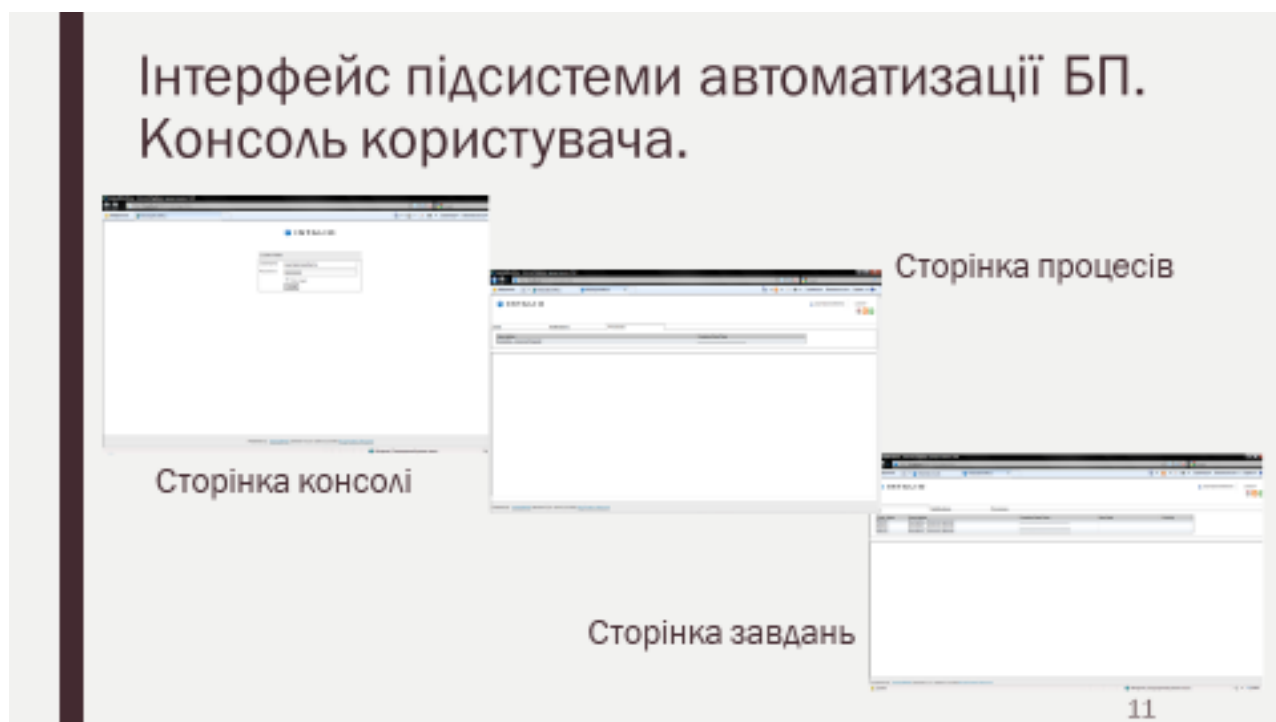


Рисунок Б.11 – Слайд 11

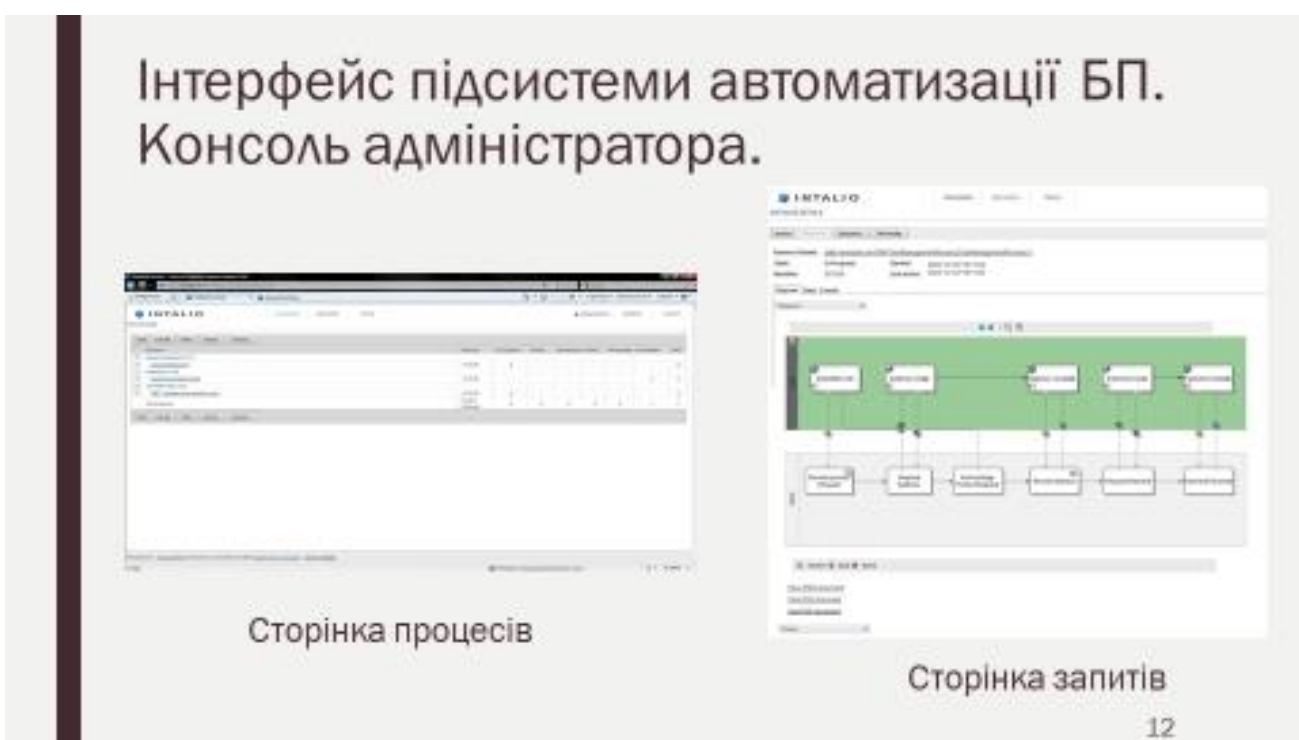


Рисунок Б.12 – Слайд 12

Висновки

- В середовищі системи Intalio | BPMS було розроблено підсистему автоматизації БП «Оформлення online-замовлення», що надає користувачеві можливість оформити замовлення на потрібний товар, а адміністратору – в режимі реального часу здійснювати моніторинг БП.
- Моделювання БП виконується в середовищі Intalio | Designer з використанням опису визначень та реалізації процесів XPDL, а моніторинг реалізації БП – у середовищі Intalio | Server з використанням мови формального опису БП та протоколів їх взаємодії між собою BPEL.
- Наукова новизна роботи полягає в тому, що запропоновано методику автоматизації БП на основі удосконалених функцій моделювання та моніторингу в середовищі BPM-системи.
- Практична цінність роботи полягає в тому, що підсистема автоматизації БП надає зручні інструменти моделювання та моніторингу, що дозволять підвищити ефективність автоматизованої системи управління підприємством.
- В подальшій роботі можливо розширити функціональні можливості підсистеми шляхом реалізації процедури оптимізації БП.

13

Рисунок Б.13 – Слайд 13

Дякую за увагу !!

14

Рисунок Б.14 – Слайд 14