| **Obraz zawierający tekst, Czcionka, logo, symbol  Opis wygenerowany automatycznie**  **WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**    Automatyka i Robotyka  Informatyka w sterowaniu i zarządzaniu  Modelowanie i analiza procesów biznesowych    *Przygotowanie i uruchomienie prostego symulatora procesów biznesowych oraz opracowanie przykładu wraz tutorialem użycia na bazie istniejącej implementacji* | | | |
| --- | --- | --- | --- |
| ***L.p.*** | **Członek** | **Numer albumu** | **Adres e-mail** |
| *1* | Michał Motyl | 401943 | michamotyl@student.agh.edu.pl |
| *2* | Kamil Pieprzycki | 402037 | pieprzycki@student.agh.edu.pl |

**Spis treści**

[**1. Wstęp 3**](#_gjdgxs)

[**2. Camunda Platform 7 3**](#_30j0zll)

[2.1. Instrukcja instalacji 3](#_1fob9te)

[2.2. Utworzenie diagramów BPMN 3](#_3znysh7)

[2.3. Ekstrakcja logów z procesów. 5](#_1t3h5sf)

[2.4. Wyniki 6](#_3rdcrjn)

[**3. Wykorzystanie biblioteki PM4PY 7**](#_26in1rg)

[3.1. Implementacja 7](#_lnxbz9)

[3.2. Wyniki 10](#_1y810tw)

[**4. Podsumowanie 10**](#_4i7ojhp)

# Wstęp

Założeniem projektu było utworzenie symulatora zwracającego dane sumaryczne na podstawie utworzonych wcześniej diagramów BPMN w celu ich analizy i tworzenia statystyk, oraz opracowanie przykładu wraz tutorialem użycia. W tym celu wykorzystano dwa sposoby ekstrakcji logów z wykonywanych procesów by na ich podstawie umożliwić dalszą obróbkę danych oraz utworzenie aplikacji.

Wykorzystane narzędzia i bilbioteki:

* Camunda Platform 7
* Biblioteka pm4py

# Camunda Platform 7

W celu realizacji projektu wykorzystano oprogramowanie Camunda Platform 7 będące kompleksowym narzędziem do zarządzania procesami biznesowymi, wspierającym w modelowaniu, automatyzacji i optymalizacji procesów. Platforma zawiera moduły BPMN Workflow Engine, the DMN Decision Engine, Tasklist i Cockpit Basic

Dodatkowo Camunda udostępnia interfejs API REST, umożliwiający programowalną interakcje z platformą, a także przeprowadzenie operacji uruchamiania i zamykania instancji procesów, oraz pobieranie ich danych procesów i zarządzanie nimi.

## Instrukcja instalacji

Aby umożliwić instalację aplikacji na urządzeniu wymagane jest pobranie oraz zainstalowanie:

* Pakietu Java SE Development Kit,
* Camunda Open Source Desktop Modeler
* Camunda Platform 7 Open Source Community Edition

Następnie należy uruchomić plik start.bat znajdujący się w folderze Camunda Platform. Uruchomienie pliku inicjuje platformę Camunda i umożliwia jej użycie, co pozwala na interakcję z aplikacjami internetowymi, a także dostęp do uruchomionych procesów.

Po pomyślnym zainicjowaniu, platforma Camunda jest ustawiona na localhost:8080. Aby rozpocząć pracę, należy zalogować się do Tasklist Camunda przy użyciu danych uwierzytelniających demo/demo. Po wykonaniu tych kroków można przejść do tworzenia diagramów BPMN.

## Utworzenie diagramów BPMN

Tworzenie diagramów BPMN w oprogramowaniu Camunda jest operacją bardzo szybką i intuicyjną, jednak aby umożliwić ich poprawne uruchomienie i dalszą analizę wymagane jest utworzenie oprogramowania odpowiedzialnego za realizacje warunków logicznych poszczególnych zadań. Na ilustracjach Rys. 2.1, Rys. 2.2, Rys. 2.3 przedstawiono proste przykładowe diagramy w celu realizacji dalszych założeń projektu.

Obraz zawierający diagram, linia, Plan, Rysunek techniczny

Opis wygenerowany automatycznie

*Rys. 2.1. Diagram kolokwium.bpmn*

Obraz zawierający diagram, linia, Plan, Rysunek techniczny

Opis wygenerowany automatycznie

*Rys. 2.2. Diagram rome.bpmn*

Obraz zawierający diagram, linia, Prostokąt, Rysunek techniczny

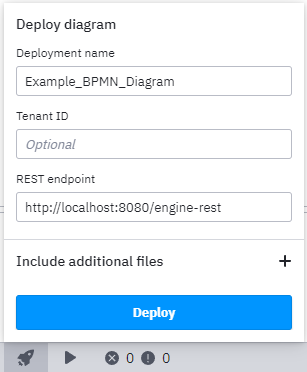
Opis wygenerowany automatycznie

*Rys. 2.3. diagram obiad.bpmn*

## Ekstrakcja logów z procesów.

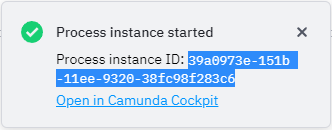
Po utworzeniu diagramów BPMN należy je wdrożyć do uruchomionego środowisko poprzez przycisk deploy

Następnie uruchom Camunda Modeler, stwórz pierwszy diagram BPMN i wdroż go co zostało przedstawione na rysunku Rys. 2.4



*Rys. 2.4. Deployment chart*

Następnie należy uruchomić proces i skopiować ID instancji procesu które się pojawi następujący komunikat (Rys. 2.5).



*Rys. 2.5. ID instancji procesu*

Skopiowany kod należy wkleić do uruchomionego programu, wysyłającego żądanie do silnika Camundy i zapisującego historię procesu do pliku CSV. Kod został przedstawiony na ilustracji Rys. 2.6.

**Zapisane logi są gotowe do preparacji symulatora, lecz ze względów opisanych w podrozdziale 2.4 rozwiązanie okazało się nieoptymalne i wymaganym było zmienienie podejścia do problemu projektowego.**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

*Rys. 2.6. Program wyciągający logi z Camundy*

## Wyniki

Podejście z wykorzystaniem platformy Camunda nie zadziałało, ze względu stopnia skomplikowania uruchamiania procesów z uwagi na konieczność pisania skryptów do każdego diagramu BPMN. Dodatkowo ekstrakcja danych przy takiej konfiguracji nie posiada symulowania tokenów, które wdrażają element losowości i zmiennych potrzebnych do dalszych działań.

Docelowo byłoby to rozwiązanie, które przy większej dostępności zasobów przyniosłoby zadowalające efekty umożliwiające dogłębną i intuicyjną analizę procesów biznesowych.

# Wykorzystanie biblioteki PM4PY

Ze względu na znaczące trudności prowadzenia projektu przy wykorzystaniu Camunda Platform zdecydowano na zmianę podejścia do generacji symulacji i implementacji przy wykorzystaniu skryptu napisanego w języku Python z wykorzystaniem dedykowanej do niego biblioteki PM4PY. Została ona stworzona do obsługi i analizy procesów biznesowych. Oferuje wiele funkcji do wczytywania, przetwarzania, analizy i wizualizacji modeli procesów oraz dzienników zdarzeń

## Implementacja

Przygotowany skrypt wykorzystuje wcześniej przygotowane diagramy BPMN. W celu ekstrakcji logów biblioteka wymaga importu diagramu oraz jego reprezentacji w postacji modelu Petri net. Poniżej prezentowane jest rozwiązanie krok po kroku.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

*Rys. 3.1. Kod odpowiedzialny za import danych*

Wykorzystana funkcja **pm4py.convert\_to\_petri\_net(bpmn)** jest obarczona pewnym błędem, skutującym utratą częsci danych w przypadku bardziej skomplikowanych diagramów. W celu weryfikacji interpretacji skrypt został wyposażony w funkcję wizualizującą przekonwertowany diagram co zostało przedstawione na Rys. 3.2

Obraz zawierający diagram, linia, biały

Opis wygenerowany automatycznie

*Rys. 3.2. Reprezentacja procesu w reprezentacji Petri Net*

Kolejna część kodu odpowiada za znalezienie wszystkich możliwych permutacji wykonania procesów, a następnie ekstrakcje logów do struktury Data Rys. 3.3

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

*Rys. 3.3. Kod odpowiedziany za ekstrakcje logów*

Następnie następuje ich pogrupowanie zgodnie z wartością ID case:concept:name i utworzenie tabeli definiującej wartości czasu wykonania, kosztów i zysków poszczególnych procesów, które mogą być modyfikowane przez użytkownika Rys. 3.4

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

*Rys. 3.4. Grupowanie logów*

Kolejną częścią jest ta odpowiedzialna za utworzenie słowników mających służyć do sumowania wartości poszczególnych zadań podczas wykonywania symulacji. Algorytm przechodzi po każdym Data Frame i oblicza wartości czasu, kosztów i zysku na podstawie wczytanych danych dotyczących rozkładów czasu, kosztów i zysku dla poszczególnych zadań. Obliczone wartości są sumowane dla odpowiednich jednostek i przechowywane w słownikach *time\_dict, costs\_dict* i *profit\_dict. Przedstawiony kod został na rysunkach* Rys. 3.5 *-* Rys. 3.8

*Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie*

*Rys. 3.5. Tworzenie słowników*

*Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie*

*Rys. 3.6. Kalkulacja czasu*

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

*Rys. 3.7. Kalkulacja kosztów*

Obraz zawierający tekst, oprogramowanie, Oprogramowanie multimedialne, multimedia

Opis wygenerowany automatycznie

*Rys. 3.8.Kalkulacja zysków*

Ostatnim krokiem symulacji jest kalkulacja bilansów i zapis ich do pliku csv Rys. 3.9

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

*Rys. 3.9. Kalkulacja bilansów i zapis do pliku*

Ostatecznie prezentowany wyżej notebook został w prosty sposób przerobiony na postać skryptową, dzięki czemu finalny użytkownik może wywołać i uruchomić symulacje bez potrzeby zagłębiania się w plik notebookowy, lecz tylko wywołanie skryptu i wykonywanie instrukcji w nim zawartych. Dopiero jeżeli w wynikach zobaczymy dane których użytkownik mógł się nie spodziewać, można wejść do notatnika i przeglądnąć etapami algorytm: sprawdzić czy diagramy są poprawnie wczytywane, czy poprawnie są przerabiane na sieci petriego, oraz czy znalezione ścieżki są kompletne.

## Wyniki

Wykorzystana implementacja symulatora w postaci skryptu zwraca interesujące nas dane na temat wykonujących się procesów, nie jest jednak rozwiązaniem w pełni automatycznym, wymaga ona interakcji z użytkownikiem. Program faktycznie zwraca interesujące nas dane, lecz bazuje on przede wszystkim na naiwnym założeniu równego prawdopodobieństwa wystąpienia każdej z wykrytych ścieżek w diagramie. Dodatkowo, wykorzystywane przez nas rozwiązanie (a dokładniej metoda z biblioteki pm4py) wymusza reprezentacje procesu w postaci sieci petriego, co potencjalnie może powodować problemy. Po pierwsze, wygenerowana sieć petriego na podstawie diagramu bpmn może być zwyczajnie błędna. Metody automatycznej generacji sieci petriego na podstawie diagramu bpmn dostępne w bibliotece pm4py nie nadają się do automatycznej generacji sieci dla dużych i skomplikowanych diagramów. Stąd ograniczenie na złożoność plików bpmn. Dodatkowo dla niektórych plików bpmn po poprawnym wygenerowaniu sieci petriego, algorytm poszukujący wszystkich możliwych ścieżek przez sieć czasami pomija ostatnie zadanie w sieci. Nie udało nam się znaleźć żadnej prawidłowości w tym zachowaniu. Zależnie od diagramu takie zadanie może być szybko dodane do ścieżek w sposób ręczny, lecz jako że celem było stworzenie automatycznego symulatora takie rozwiązanie nie jest pożądane.

# Podsumowanie

Projekt miał na celu stworzenie symulatora procesów biznesowych i opracowanie przykładu z tutorialem. Mimo trudności na początku, zmiana podejścia i wykorzystanie biblioteki PM4PY pozwoliły na uzyskanie interesujących wyników analizy procesów biznesowych.

Docelowo wykorzystanie oprogramowania Camunda Platform jest rozwiązaniem, do którego należałoby dążyć gdyż jest w stanie umożliwić pełną automatyzację działania symulacji. Aby to osiągnąć, projekt wymagałby jednak znacznego nakładu pracy przekraczającego wstępne założenia.

Jako jedną ze ścieżek, którą można by przyjąć za krok ku rozwojowi tego problemu byłoby utworzenie osobnego symulatora generującego samodzielnie działające diagramy BPMN, jednak ilość pracy potrzebna do wykonania tego założenia wymagałaby wielokrotnie większego nakładu pracy niż zakładaliśmy podczas realizacji obecnego projektu.