



**Politechnika
Śląska**

PROJEKT INŻYNIERSKI

Aplikacja do tworzenia zautomatyzowanych przepływów pracy i procesów
biznesowych

Grzegorz PIOTROWSKI

Nr albumu: 290596

Kierunek: Informatyka

Specjalność: Bazy Danych i Inżynieria Systemów

PROWADZĄCY PRACĘ

Dr. inż Marcin Połomski

KATEDRA Wydział Algorytmiki i Oprogramowania

Wydział Automatyki, Elektroniki i Informatyki

Gliwice 2023

Tytuł pracy

Aplikacja do tworzenia zautomatyzowanych przepływów pracy i procesów biznesowych

Streszczenie

Celem pracy jest projekt i implementacja aplikacji webowej umożliwiającej modelowanie i automatyzację (w wybranym zakresie, tj. np. automatyzacja procesu, wysyłanie przypomnień, przesyłanie/transformacja danych) zadań i przepływów pracy w środowisku biznesowym. Zakres pracy:

1. Analiza dziedziny i przegląd rozwiązań.
2. Analiza wymagań i wybór odpowiednich narzędzi.
3. Projekt oraz implementacja systemu (w formie aplikacji webowej).
4. Wdrożenie i testowanie systemu.

Słowa kluczowe

- Aplikacja webowa
- Modelowanie i automatyzacja
- Procesy i przepływy pracy

Thesis title

Application to create automated workflows and business processes

Abstract

The purpose of the work is to design and implement a web application that enables modeling and automation (to a selected extent, i.e., process automation, sending reminders, data transfer/transformation) of tasks and workflows in a business environment.

Key words

- Web application
- Modeling and automation
- Processes and workflows.

Spis treści

1	Wstęp	1
1.1	Wprowadzenie w problem	1
1.2	Osadzenie problemu w dziedzinie	1
1.3	Cel pracy	2
1.4	Zakres pracy	3
1.5	Charakterystyka rozdziałów	3
1.6	Określenie wkładu autora	3
2	[Analiza tematu]	5
3	Wymagania i narzędzia	7
4	[Właściwy dla kierunku – np. Specyfikacja zewnętrzna]	9
5	[Właściwy dla kierunku – np. Specyfikacja wewnętrzna]	11
6	Weryfikacja i walidacja	13
7	Podsumowanie i wnioski	15
	Bibliografia	17
	Spis skrótów i symboli	21
	Źródła	23
	Lista dodatkowych plików, uzupełniających tekst pracy	27
	Spis rysunków	29
	Spis tabel	31

Rozdział 1

Wstęp

Obserwowany od lat 90. XX wieku intensywny rozwój technologii, nazywany dziś czwartą rewolucją przemysłową, spowodował wzrost zainteresowania tematyką automatyzacji ze strony środowisk przemysłowych, biznesowych, jak i rządowych. Rozwój technologii trwa nadal i ma ciągły wpływ na przemysł oraz sposób prowadzenia biznesu.

1.1 Wprowadzenie w problem

Zwiększenie świadomości przedsiębiorców na temat automatyzacji otworzyło rynek na nowe podejście do tworzenia oprogramowania w postaci platform No-code i Low-code. Platformy tego rodzaju umożliwiają tworzenie prostych aplikacji, a także modelowanie procesów biznesowych i ich automatyzację za pomocą diagramów przepływów danych, bez głębokiego zrozumienia problematyki tworzenia oprogramowania. Działają one przy użyciu przygotowanych prefabrykatów, w postaci gotowych akcji, wymagających jedynie sparametryzowania.

Przykładami takich akcji może być wysyłka sparametryzowanego emaila, lub wpis o nowym produkcie do Excela.

1.2 Osadzenie problemu w dziedzinie

Ważnym elementem systemów umożliwiającym klientowi automatyzację jego procesów jest możliwość dołączania elementów specyficznych dla ich biznesu w postaci interaktywnej części systemu, bez konieczności tworzenia dodatkowego oprogramowania dla każdego nowego użytkownika platformy.

Problem ten jest uniwersalnym zagadnieniem dla wielu systemów, które muszą mierzyć się z zapewnieniem rozszerzalności zakresu dostępnych operacji, przy jednoczesnym jak największym ich wyabstrakcjonowaniu.

Jedną z odpowiedzi na ten problem jest zaproponowana przez Dr. Alistaira Cockburna

w 2005 roku architektura portów i adapterów[1], zwana też architekturą heksagonalną, której jednym z podstawowych celów było zapewnienie jak najmniejszej zależności pomiędzy logiką biznesową a pozostałą częścią aplikacji.

Zaletą tego podejścia jest łatwość, z jaką mogą być zastępowane lub dodawane wszystkie elementy niezwiązane z logiką biznesową. Jedynym warunkiem, który musi być spełniony, jest spełnianie wymagań narzuconych przez interfejs wystawiany przez rdzeń aplikacji.

Dzięki zastosowaniu w odpowiedni sposób architektury heksagonalnej można rozwiązać problem związany z dodawaniem elementu specyficznego dla klienta, poprzez zostanie portu który może zostać wykorzystany jako wtyczka poprzez odpowiedni protokół komunikacyjny.

1.3 Cel pracy

Celem niniejszej pracy jest zaprojektowanie i stworzenie platformy umożliwiającej modelowanie i automatyzację przepływów pracy w środowisku biznesowym z jednoczesnym udostępnieniem możliwości dopięcia elementów specyficznych dla klienta w postaci portu poprzez protokół HTTP. Założenia funkcjonalne aplikacji:

- aplikacja powinna dawać możliwość modelowania procesów w formie grafów
- aplikacja powinna wykonywać zamodelowane procesy gdy pojawi się odpowiednio zdefiniowany sygnał wejściowy
- korzeń grafu definiuje jaki rodzaj sygnału wyzwalającego rozpocznie wykonywanie procesu
- sygnałami wyzwalającymi procesy mogą być zapytania HTTP lub zdefiniowane momenty czasu
- każdy wierzchołek grafu ma być wykonywalną akcją
- akcje dzielą się na elementy wykonywalne i elementy porównujące
- zamodelowane procesy powinny być przechowywane w bazie danych
- aplikacja udostępnia publiczny interfejs API, umożliwiający rejestrację zewnętrznych serwisów, które po rejestracji stają się elementem wykonywalnym grafu

Założenia нефункционалне aplikacji:

- aplikacja powinna być w stanie jednocześnie wykonywać conajmniej 1000 procesów biznesowych

- system powinien dostarczać wgląd w podstawowy zbiór sygnałów związanych ze złotymi sygnałami ruchu SRE[1]
- system powinien móc skalować się horyzontalnie
- system powinien być zaprojektowany w sposób umożliwiający przetestowanie logiki biznesowej w pamięci

1.4 Zakres pracy

Działająca aplikacja powinna spełniać wszystkie założenia funkcjonalne

1.5 Charakterystyka rozdziałów

1.6 Określenie wkładu autora

Rozdział 2

[Analiza tematu]

- sformułowanie problemu
- osadzenie tematu w kontekście aktualnego stanu wiedzy (*state of the art*) o poruszonym problemie
- studia literaturowe [**bib:internet**, 3, 4, 2] - opis znanych rozwiązań (także opisanych naukowo, jeżeli problem jest poruszany w publikacjach naukowych), algorytmów,

Wzory

$$y = \frac{\partial x}{\partial t} \tag{2.1}$$

jak i pojedyncze symbole x i y składa się w trybie matematycznym.

Rozdział 3

Wymagania i narzędzia

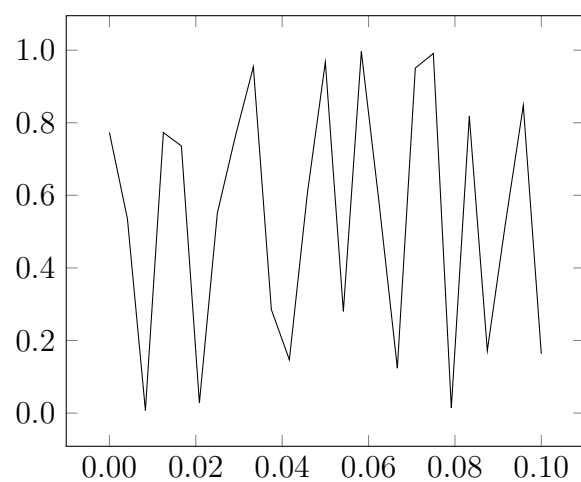
- wymagania funkcjonalne i нефункционалне
- przypadki użycia (diagramy UML) – dla prac, w których mają zastosowanie
- opis narzędzi, metod eksperymentalnych, metod modelowania itp.
- metodyka pracy nad projektowaniem i implementacją – dla prac, w których ma to zastosowanie

Rozdział 4

[Właściwy dla kierunku – np. Specyfikacja zewnętrzna]

Jeśli „Specyfikacja zewnętrzna”:

- wymagania sprzętowe i programowe
- sposób instalacji
- sposób aktywacji
- kategorie użytkowników
- sposób obsługi
- administracja systemem
- kwestie bezpieczeństwa
- przykład działania
- scenariusze korzystania z systemu (ilustrowane zrzutami z ekranu lub generowanymi dokumentami)



Rysunek 4.1: Podpis rysunku po rysunkiem.

Rozdział 5

[Właściwy dla kierunku – np. Specyfikacja wewnętrzna]

Jeśli „Specyfikacja wewnętrzna”:

- przedstawienie idei
- architektura systemu
- opis struktur danych (i organizacji baz danych)
- komponenty, moduły, biblioteki, przegląd ważniejszych klas (jeśli występują)
- przegląd ważniejszych algorytmów (jeśli występują)
- szczegóły implementacji wybranych fragmentów, zastosowane wzorce projektowe
- diagramy UML

Krótką wstawka kodu w linii tekstu jest możliwa, np. **int a;** (biblioteka `listings`). Dłuższe fragmenty lepiej jest umieszczać jako rysunek, np. kod na rys 5.1, a naprawdę długie fragmenty – w załączniku.

```
1 class test : public basic
2 {
3     public:
4         test (int a);
5         friend std::ostream operator<<(std::ostream & s,
6                                         const test & t);
7     protected:
8         int _a;
9
10 };
```

Rysunek 5.1: Pseudokod w `listings`.

Rozdział 6

Weryfikacja i walidacja

- sposób testowania w ramach pracy (np. odniesienie do modelu V)
- organizacja eksperymentów
- przypadki testowe zakres testowania (pełny/niepełny)
- wykryte i usunięte błędy
- opcjonalnie wyniki badań eksperymentalnych

Tabela 6.1: Nagłówek tabeli jest nad tabelą.

ζ	metoda						
	alg. 1	alg. 2	alg. 3			alg. 4, $\gamma = 2$	
			$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

Rozdział 7

Podsumowanie i wnioski

- uzyskane wyniki w świetle postawionych celów i zdefiniowanych wyżej wymagań
- kierunki ewentualnych danych prac (rozbudowa funkcjonalna ...)
- problemy napotkane w trakcie pracy

Bibliografia

- [1] Dr. Alistair Cockburn. *Hexagonal architecture*. 2005. URL: <https://alistair.cockburn.us/hexagonal-architecture/> (term. wiz. 01.04.2005).
- [2] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. „Tytuł artykułu konferencyjnego”. W: *Nazwa konferencji*. 2006, s. 5346–5349.
- [3] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. „Tytuł artykułu w czasopiśmie”. W: *Tytuł czasopisma* 157.8 (2016), s. 1092–1113.
- [4] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. *Tytuł książki*. Warszawa: Wydawnictwo, 2017. ISBN: 83-204-3229-9-434.

Dodatki

Spis skrótów i symboli

DNA kwas deoksyrybonukleinowy (ang. *deoxyribonucleic acid*)

MVC model – widok – kontroler (ang. *model-view-controller*)

N liczebność zbioru danych

μ stopnień przyleżności do zbioru

\mathbb{E} zbiór krawędzi grafu

\mathcal{L} transformata Laplace’a

Źródła

Bibliografia

```
1 if (_nClusters < 1)
2     throw std::string ("unknown number of clusters");
3 if (_nIterations < 1 and _epsilon < 0)
4     throw std::string ("You should set a maximal number of
        iteration or minimal difference — epsilon.");
5 if (_nIterations > 0 and _epsilon > 0)
6     throw std::string ("Both number of iterations and minimal
        epsilon set — you should set either number of iterations
        or minimal epsilon.");
```

Lista dodatkowych plików, uzupełniających tekst pracy

W systemie do pracy dołączono dodatkowe pliki zawierające:

- źródła programu,
- dane testowe,
- film pokazujący działanie opracowanego oprogramowania lub zaprojektowanego i wykonanego urządzenia,
- itp.

Spis rysunków

4.1	Podpis rysunku po rysunkiem.	10
5.1	Pseudokod w <code>listings</code>	12

Spis tabel

6.1	Nagłówek tabeli jest nad tabelą.	14
-----	--	----