

Uniwersytet Gdański Wydział Matematyki, Fizyki i
Informatyki Instytut Informatyki

Oliver Gruba, Maciej Nasiadka

22 grudnia 2025

Imię i Nazwisko (nr indeksu)	Oliver Gruba (292583) Maciej Nasiadka (292574)
Nazwa uczelni	Uniwersytet Gdański
Kierunek	Informatyka (profil praktyczny)
Prowadzący	dr inż. Stanisław Witkowski
Nazwa ćwiczenia	Tworzenie diagramów UML z wykorzystaniem sztucznej inteligencji (AI)
Numer sprawozdania	9
Data zajęć	18.12.2025
Data oddania	07.01.2026
Miejsce na ocenę	

Spis treści

1	Wprowadzenie	4
2	Metodyka przeprowadzonych badań	4
3	Darmowe narzędzia UML wykorzystujące AI	4
3.1	Uzasadnienie ograniczonego kontekstu wejściowego	5
3.2	Dobór typów diagramów UML	5
3.3	Claude + Mermaid	6
3.3.1	Wygenerowany diagram przypadków użycia	6
3.3.2	Wygenerowany opis diagramu	7
3.3.3	Poprawki	7
3.4	ChatGPT + diagram aktywności	7
3.4.1	Wygenerowany diagram	8
3.4.2	Wygenerowany opis diagramu	9
3.4.3	Poprawki	9
3.5	Grok + diagram BPMN	9
3.5.1	Wygenerowany diagram	10
3.5.2	Wygenerowany opis diagramu	10
3.5.3	Poprawki	10
3.6	Gemini + Mermaid	11
3.6.1	Wygenerowany diagram	12
3.6.2	Wygenerowany opis diagramu	12
3.6.3	Poprawki	12
4	Kryteria oceny wygenerowanych diagramów	13
5	Analiza porównawcza zachowania modeli AI	13
6	Zalety zastosowania AI w tworzeniu diagramów UML	13
6.1	Automatyzacja procesu projektowego	13
6.2	Wsparcie dla mniej doświadczonych projektantów	14
6.3	Spójność i standaryzacja diagramów	14
6.4	Integracja z procesem wytwarzania oprogramowania	14
7	Wady i ograniczenia wykorzystania AI	14
7.1	Ograniczone zrozumienie kontekstu biznesowego	14
7.2	Jakość danych wejściowych	14

7.3	Ograniczona elastyczność projektowa	15
8	Przyszłość tworzenia diagramów UML	15
8.1	Dynamiczne i adaptacyjne diagramy	15
8.2	Integracja z narzędziami analitycznymi	15
8.3	Rozszerzenie roli AI jako asystenta architekta	15
9	Przyszłe wytyczne do projektowania oprogramowania z AI	15
9.1	Rola projektanta	16
9.2	Transparentność i kontrola	16
9.3	Edukacja i kompetencje	16
10	Propozycje zadania laboratoryjnego.	16
10.1	Zakres zadania	16
10.2	Cele dydaktyczne	16
10.3	Efekty końcowe	17
11	Wnioski z przeprowadzonego eksperymentu	17
12	Podsumowanie	17

1. Wprowadzenie

Rozwój sztucznej inteligencji (Artificial Intelligence, AI) w ostatnich latach znacząco wpłynął na sposób projektowania oraz dokumentowania systemów informatycznych. Narzędzia wykorzystujące AI coraz częściej wspierają programistów i architektów oprogramowania w procesach analitycznych, projektowych oraz decyzyjnych.

Jednym z obszarów, w których obserwuje się rosnące zastosowanie AI, jest tworzenie diagramów UML (Unified Modeling Language). Automatyczne generowanie diagramów na podstawie opisu tekstowego, kodu źródłowego lub analizy wymagań pozwala przyspieszyć proces projektowy oraz zwiększyć jego dostępność dla osób o różnym poziomie doświadczenia.

Celem niniejszego sprawozdania jest omówienie zalet, ograniczeń oraz perspektyw rozwoju tworzenia diagramów UML z wykorzystaniem sztucznej inteligencji, a także wskazanie potencjalnych kierunków dalszego zastosowania AI w projektowaniu oprogramowania.

2. Metodyka przeprowadzonych badań

Przeprowadzone badania miały charakter eksploracyjny i jakościowy. Ich celem nie było ilościowe porównanie modeli językowych ani ocena ich „jakości” w sensie benchmarkowym, lecz analiza zachowania narzędzi opartych o sztuczną inteligencję w kontekście wspomaganie procesu tworzenia diagramów UML przy ograniczonym wkładzie użytkownika.

Badanie skoncentrowało się na obserwacji:

- sposobu interpretacji krótkiego i nieprecyzyjnego opisu problemu,
- zdolności modelu do samodzielnego doboru odpowiedniej notacji,
- poprawności strukturalnej generowanych diagramów,
- konieczności ingerencji użytkownika w celu uzyskania poprawnego rezultatu.

Zastosowana metodyka odpowiada rzeczywistemu scenariuszowi pracy początkującego użytkownika, który nie posiada pełnej wiedzy z zakresu notacji UML lub nie potrafi jednoznacznie sformułować wymagań projektowych.

3. Darmowe narzędzia UML wykorzystujące AI

W celach dydaktycznych wykorzystano różne narzędzia i modele językowe. Użyliśmy najnowszych dostępnych i darmowych do wykorzystania modeli językowych w trybie agenta:

- Claude Sonnet 4.5

- ChatGPT 5.2
- Grok Code Fast 1
- Gemini 3 Fast

Typy diagramu dla agenta:

- Diagram przypadków użycia - Claude
- Diagram aktywności - ChatGPT
- Diagram BPMN - Grok
- Diagram sekwencji - Gemini

3.1. Uzasadnienie ograniczonego kontekstu wejściowego

Wykorzystane prompty zostały celowo skrócone i pozbawione szczegółowych wytycznych. Zabieg ten miał na celu symulację sytuacji, w której użytkownik:

- nie zna pełnej specyfikacji UML,
- nie jest pewien oczekiwanego poziomu szczegółowości diagramu,
- traktuje AI jako narzędzie wspierające proces myślowy, a nie jedynie generator kodu.

Ograniczenie liczby tokenów wejściowych pozwalało również zaobserwować mechanizmy autokorekty modeli językowych oraz ich skłonność do:

- nadmiernego uogólniania,
- tworzenia zbędnych elementów diagramu,
- pomijania istotnych informacji technicznych.

3.2. Dobór typów diagramów UML

Każdemu modelowi językowemu przypisano inny typ diagramu UML. Decyzja ta była celowa i miała na celu:

- uniknięcie bezpośredniego porównywania modeli na identycznym zadaniu,
- sprawdzenie elastyczności AI w różnych aspektach modelowania systemu,
- ocenę, czy AI lepiej radzi sobie z modelami statycznymi czy dynamicznymi.

Zastosowane diagramy reprezentują odmienne perspektywy projektowe:

- diagram przypadków użycia — perspektywa funkcjonalna,
- diagram aktywności — perspektywa procesowa,
- diagram BPMN — perspektywa biznesowa,
- diagram sekwencji — perspektywa interakcyjna.

Takie podejście pozwala na szerszą ocenę przydatności AI w całym cyklu projektowym, zamiast ograniczać analizę do jednego wycinka UML.

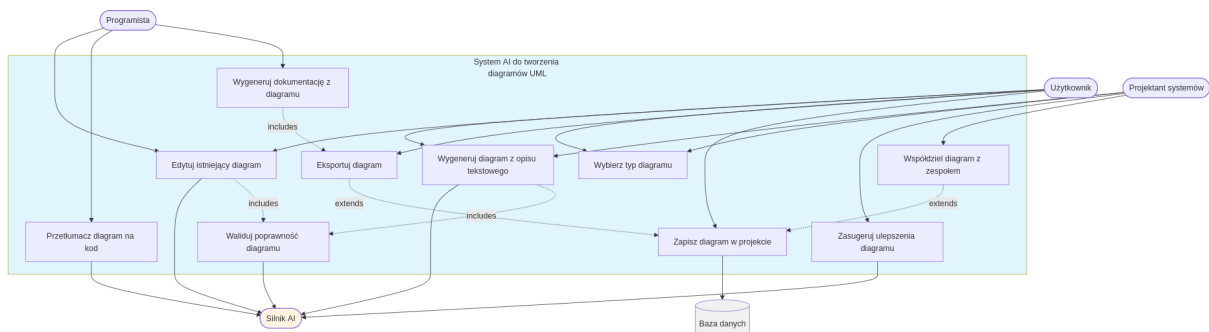
3.3. Claude + Mermaid

Wygenerowany diagram przypadków użycia dla *systemu tworzenia diagramów przez AI*

Wykorzystany prompt:

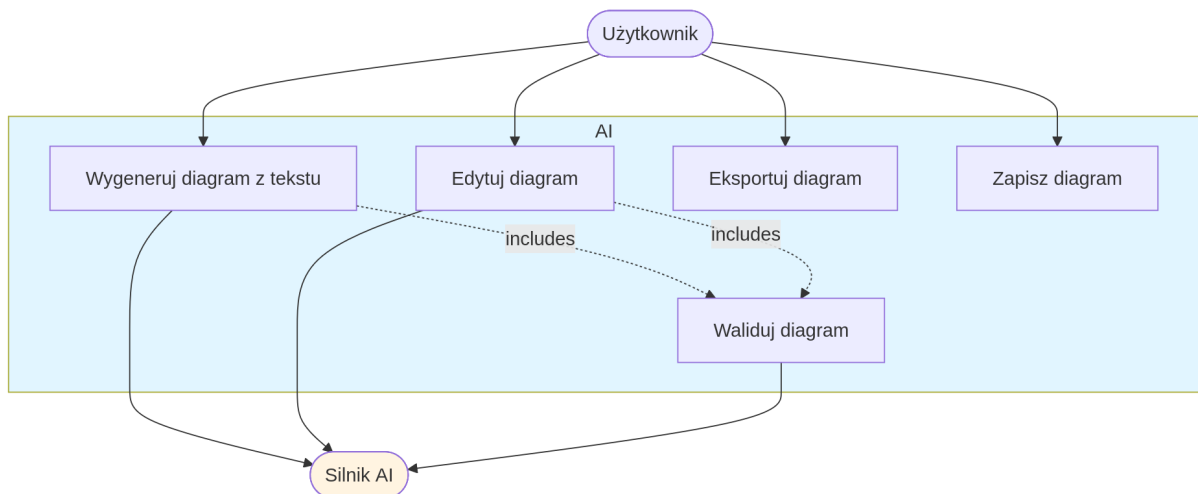
Stwórz diagram przypadków użycia o "tworzeniu diagramów UML z wykorzystaniem sztucznej inteligencji".
Wybierz odpowiednie narzędzie.

3.3.1 Wygenerowany diagram przypadków użycia



Rysunek 1: Oryginalna wersja diagramu przypadków użycia wygenerowana przez Claude Sonnet 4.5

Możemy zauważyć w modelu dużą tendencję do nadmiernego rozpisywania się, co nieraz powoduje niemałe problemy podczas tworzenia projektów lub dokumentacji.



Rysunek 2: Uproszczona wersja diagramu przypadków użycia wygenerowana przez Claude Sonnet 4.5

3.3.2 Wygenerowany opis diaramu

Opis wygenerowany przez Claude:

Ten diagram przedstawia system, w którym Użytkownik może wygenerować diagram UML z opisu tekstowego oraz edytować istniejące diagramy – obie te operacje automatycznie angażują Silnik AI, który dodatkowo waliduje poprawność utworzonego lub zmodyfikowanego diagramu. Użytkownik ma również możliwość wyeksportowania gotowego diagramu do pliku oraz zapisania go w systemie. Relacje "includes" wskazują, że walidacja przez AI jest integralną częścią zarówno generowania, jak i edycji diagramów, wykonującą się automatycznie podczas tych operacji.

3.3.3 Poprawki

Diagram nie wymagał poprawek po wstawieniu do narzędzia.

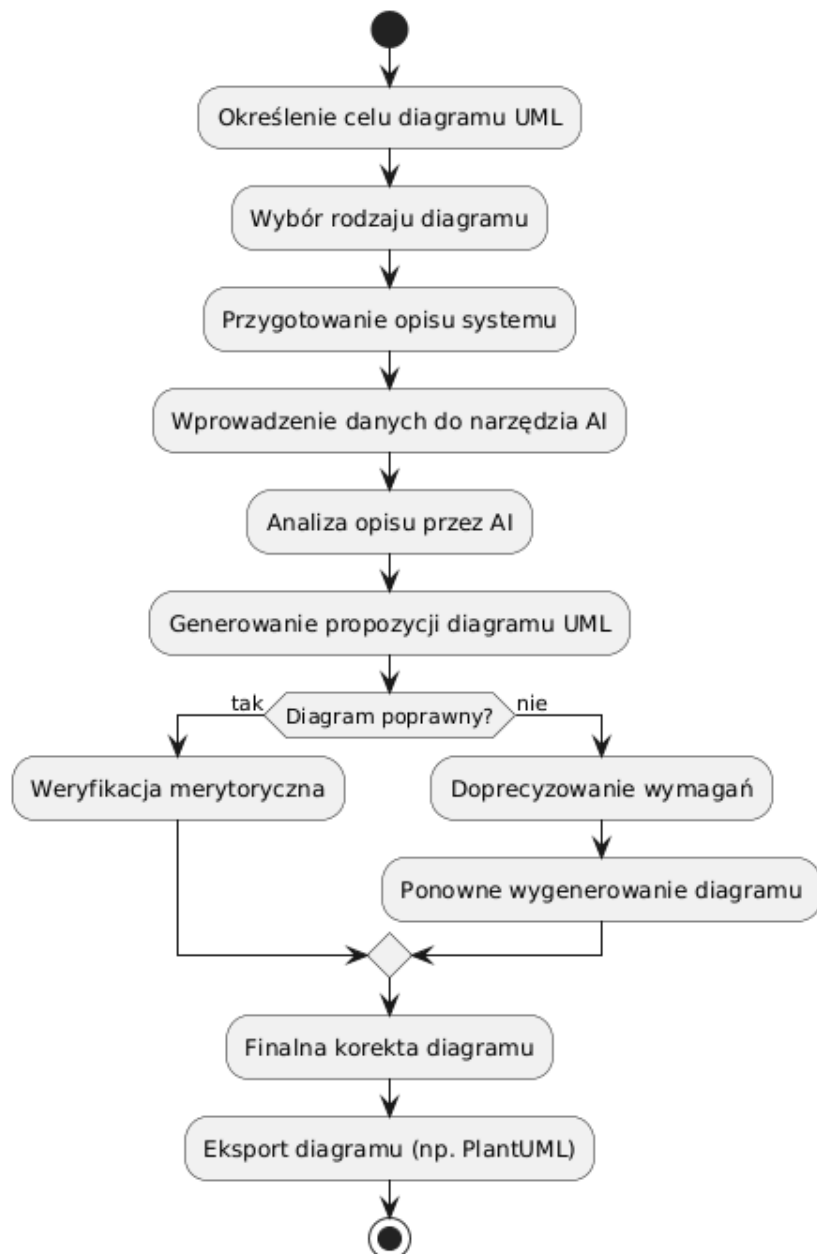
3.4. ChatGPT + diagram aktywności

Wygenerowany diagram aktywności przy użyciu ChatGPT modelu 5.2 z celem wygenerowania *diagramu aktywności dla systemu tworzącego diagramy UML przy użyciu sztucznej inteligencji*

Wykorzystany prompt:

Stwórz diagram aktywności o tworzeniu diagramów UML z wykorzystaniem sztucznej inteligencji.
Wybierz odpowiednie narzędzie.
Zrób to dokładnie, ale bez nadmiernej szczegółowości

3.4.1 Wygenerowany diagram



Rysunek 3: Oryginalna wersja diagramu przypadków użycia wygenerowana przez ChatGPT i wyrenderowana poprzez zewnętrzne narzędzie

Diagram aktywności przedstawia proces tworzenia diagramu UML z wykorzystaniem sztucznej inteligencji, rozpoczynający się od określenia celu oraz wyboru rodzaju diagramu UML. Następnie użytkownik przygotowuje opis systemu i wprowadza go do narzędzia AI, które analizuje dostarczone informacje i generuje wstępną propozycję diagramu. W kolejnym kroku następuje decyzja, czy wygenerowany diagram jest poprawny — w przypadku negatywnej oceny użytkownik doprecyzowuje wymagania, a AI ponownie generuje diagram. Proces kończy się finalną korektą oraz eksportem gotowego diagramu w wybranym formacie, np. PlantUML.

3.4.2 Wygenerowany opis diagramu

Opis wygenerowany przez ChatGPT:

Diagram aktywności przedstawia proces tworzenia diagramu UML z wykorzystaniem sztucznej inteligencji, rozpoczynający się od określenia celu oraz wyboru rodzaju diagramu UML. Następnie użytkownik przygotowuje opis systemu i wprowadza go do narzędzia AI, które analizuje dostarczone informacje i generuje wstępną propozycję diagramu. W kolejnym kroku następuje decyzja, czy wygenerowany diagram jest poprawny - w przypadku negatywnej oceny użytkownik doprecyzowuje wymagania, a AI ponownie generuje diagram. Proces kończy się finalną korektą oraz eksportem gotowego diagramu w wybranym formacie, np. PlantUML.

3.4.3 Poprawki

Kod nie wymagał poprawek.

3.5. Grok + diagram BPMN

Wygenerowany diagram aktywności przy użyciu *Grok Code Fast 1* z celem wygenerowania *diagramu BPMN dla systemu tworzącego diagramy UML przy użyciu sztucznej inteligencji*

Wykorzystany prompt:

Stwórz diagram BPMN o "tworzeniu diagramów UML z wykorzystaniem sztucznej inteligencji".
Wybierz odpowiednie narzędzie.
Zadbaj o poprawność

Drugi prompt:

Po imporcie do narzędzi, nie ma treści

3.5.1 Wygenerowany diagram

Poniżej przedstawiliśmy wygenerowany diagram. *Tak, pierwszy wygenerowany diagram był pusty.*

Rysunek 4: Oryginalna wersja diagramu BPMN wygenerowana przez Grok i wyrenderowana poprzez zewnętrzne narzędzie. Wystąpiły komplikacje, przez co kod BPMN poprawnie się importował, ale nie powodował wyświetlenia żadnej treści.

3.5.2 Wygenerowany opis diagramu

Opis wygenerowany przez Grok:

Diagram BPMN przedstawia proces tworzenia diagramów UML przy użyciu sztucznej inteligencji, zaczynając od zdarzenia startowego "Rozpoczęcie procesu". Następnie użytkownik definiuje wymagania dla diagramu UML, wybiera narzędzie AI (np. GitHub Copilot) i generuje diagram za pomocą AI. Bramka decyzyjna sprawdza, czy diagram jest poprawny:

jeśli tak, proces kończy się zdarzeniem końcowym;

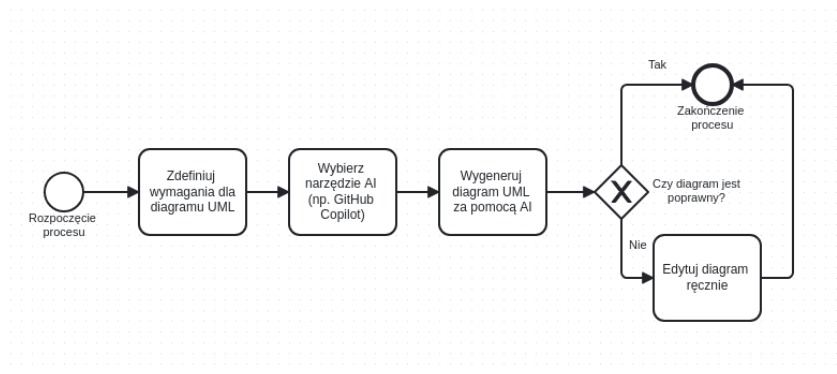
jeśli nie, diagram jest edytowany ręcznie przed zakończeniem.

Przepływy sekwencyjne łączą zadania w liniowym układzie z jedną pętlą warunkową, z elementami rozmieszczonymi poziomo od lewej do prawej strony płótna.

3.5.3 Poprawki

Wygenerowany kod wymagał poprawki, przez brak informacji o położeniu elementów na planszy. Drugi prompt symulujący zwykłego użytkownika korzystającego z technologii: Wykorzystany prompt:

Po imporcie do narzędzi, nie ma treści



Rysunek 5: Poprawiona wersja diagramu BPMN wygenerowana przez Grok i wyrenderowana poprzez zewnętrzne narzędzie. Po zwróceniu modelowi uwagi, wygenerował znacznie obszerniejszą odpowiedź, uzupełniając informacje o położeniu elementów na planszy.

3.6. Gemini + Mermaid

Wygenerowany diagram aktywności przy użyciu *Gemini 3 Fast* z celem wygenerowania *diagramu sekwencji dla systemu tworzącego diagramy UML przy użyciu sztucznej inteligencji*

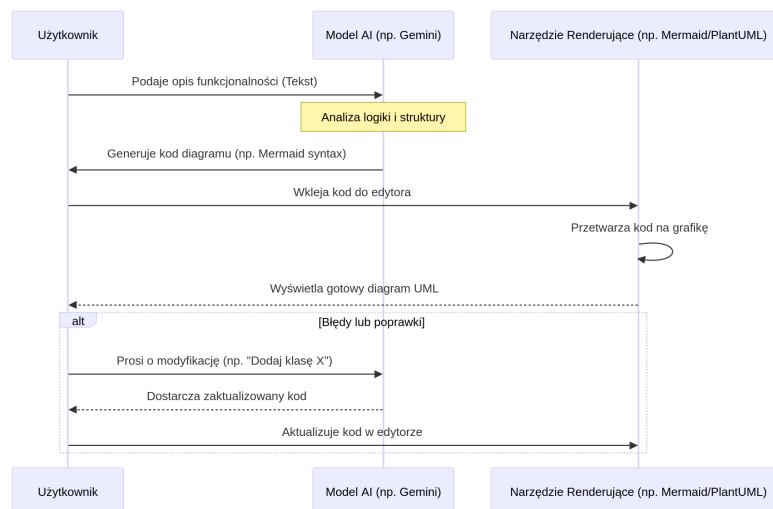
Wykorzystany prompt:

Stwórz diagram sekwencji o "tworzeniu diagramów UML z wykorzystaniem sztucznej inteligencji".

Wykorzystaj odpowiednie narzędzie.

Zrób to dokładnie, ale bez nadmiernej szczegółowości

3.6.1 Wygenerowany diagram



Rysunek 6: Oryginalna wersja diagramu sekwencji wygenerowana przez Gemini i wyrenderowana poprzez zewnętrzne narzędzie.

3.6.2 Wygenerowany opis diagramu

Odpowiedź LLM verbatim:

Ten konkretny diagram sekwencji ilustruje trójstronną interakcję pomiędzy **Użytkownikiem**, **Modelem AI** (takim jak Gemini) oraz **Narzędziem Renderującym** (np. Mermaid).

Proces rozpoczyna się od przekazania przez użytkownika opisu tekstowego, który AI przetwarza na ustrukturyzowany kod (syntax), pełniący rolę pośrednika między myślą a obrazem. Kluczowym elementem diagramu jest blok **alt**, który pokazuje specyfikę pracy z AI - możliwość błyskawicznego wprowadzenia poprawek i wygenerowania zaktualizowanego kodu w odpowiedzi na prośbę o modyfikację. Całość obrazuje, że AI w tym procesie nie jest rysownikiem, lecz "tłumaczem" logiki biznesowej na język zrozumiały dla silników graficznych.

3.6.3 Poprawki

Diagram nie wymagał poprawek.

4. Kryteria oceny wygenerowanych diagramów

Ocena wygenerowanych diagramów została przeprowadzona w oparciu o następujące kryteria jakościowe:

- poprawność notacyjna (zgodność z zasadami UML/BPMN),
- kompletność logiczna procesu lub struktury,
- czytelność i zrozumiałość diagramu,
- stopień wymaganej ingerencji użytkownika,
- zgodność diagramu z opisem tekstowym.

Analiza nie miała charakteru punktowego, lecz opisowy, co jest zgodne z eksploracyjnym charakterem badania.

5. Analiza porównawcza zachowania modeli AI

Analiza wygenerowanych diagramów wskazuje na istotne różnice w podejściu modeli językowych do zadania modelowania systemów.

Modele wykazujące tendencję do nadmiernej elaboracji generowały diagramy bogate w szczegóły, lecz trudniejsze w dalszej edycji. Z kolei modele bardziej zachowawcze tworzyły struktury prostsze, wymagające doprecyzowania, ale łatwiejsze do dalszej rozbudowy.

Istotnym wnioskiem jest fakt, że problemy techniczne (np. brak informacji o położeniu elementów w BPMN) nie wynikały z błędnej logiki procesu, lecz z niedostosowania wygenerowanego kodu do wymagań konkretnego silnika renderującego.

6. Zalety zastosowania AI w tworzeniu diagramów UML

Wykorzystanie sztucznej inteligencji w procesie tworzenia diagramów UML przynosi szereg istotnych korzyści, zarówno na etapie analizy, jak i projektowania systemu.

6.1. Automatyzacja procesu projektowego

- generowanie diagramów na podstawie opisu tekstowego lub kodu źródłowego,
- ograniczenie ręcznego rysowania elementów diagramu,
- skrócenie czasu potrzebnego na przygotowanie dokumentacji projektowej.

6.2. Wsparcie dla mniej doświadczonych projektantów

- pomoc w doborze odpowiednich typów diagramów UML,
- sugerowanie poprawnych relacji pomiędzy elementami,
- redukcja liczby błędów wynikających z nieznamomości notacji UML.

6.3. Spójność i standaryzacja diagramów

- zachowanie jednolitej struktury i stylu diagramów,
- automatyczne stosowanie zasad UML,
- łatwiejsze utrzymanie zgodności dokumentacji w dużych projektach.

6.4. Integracja z procesem wytwarzania oprogramowania

- możliwość analizy kodu i generowania diagramów aktualizowanych na bieżąco,
- wsparcie pracy zespołowej i iteracyjnego rozwoju systemu,
- lepsza synchronizacja dokumentacji z implementacją.

7. Wady i ograniczenia wykorzystania AI

Pomimo licznych zalet, zastosowanie AI w tworzeniu diagramów UML wiąże się również z pewnymi ograniczeniami, które należy uwzględnić w praktyce projektowej.

7.1. Ograniczone zrozumienie kontekstu biznesowego

- AI może nie uwzględniać specyficznych wymagań domenowych,
- brak pełnego rozumienia intencji projektanta,
- ryzyko generowania diagramów nadmiernie uproszczonych lub zbyt złożonych.

7.2. Jakość danych wejściowych

- skuteczność AI zależy od jakości opisu lub kodu źródłowego,
- nieprecyzyjne dane wejściowe prowadzą do niepoprawnych diagramów,
- konieczność ręcznej weryfikacji wygenerowanych modeli.

7.3. Ograniczona elastyczność projektowa

- generowane diagramy mogą nie odpowiadać preferowanemu stylowi architektonicznemu,
- trudności w odwzorowaniu niestandardowych rozwiązań projektowych,
- ryzyko nadmiernego polegania na sugestjach AI.

8. Przyszłość tworzenia diagramów UML

Rozwój technologii AI wskazuje na dalszą automatyzację i inteligentne wsparcie procesu projektowania systemów informatycznych.

8.1. Dynamiczne i adaptacyjne diagramy

- automatyczna aktualizacja diagramów wraz ze zmianami w kodzie,
- dostosowywanie poziomu szczegółowości do potrzeb użytkownika,
- interaktywne modele architektury.

8.2. Integracja z narzędziami analitycznymi

- analiza jakości architektury na podstawie diagramów,
- identyfikacja wąskich gardeł i zależności krytycznych,
- wsparcie decyzji architektonicznych w czasie rzeczywistym.

8.3. Rozszerzenie roli AI jako asystenta architekta

- sugerowanie alternatywnych rozwiązań projektowych,
- wykrywanie antywzorców architektonicznych,
- wspomaganie optymalizacji struktury systemu.

9. Przyszłe wytyczne do projektowania oprogramowania z AI

Wykorzystanie AI w projektowaniu systemów informatycznych wymaga wypracowania nowych zasad i dobrych praktyk.

9.1. Rola projektanta

- projektant powinien pełnić rolę decyzyjną, a nie jedynie wykonawczą,
- AI należy traktować jako narzędzie wspomagające, a nie zastępujące analizę,
- konieczność krytycznej oceny wyników generowanych przez AI.

9.2. Transparentność i kontrola

- zrozumienie sposobu generowania diagramów przez AI,
- możliwość modyfikacji i korekty modeli,
- zapewnienie zgodności z wymaganiami projektu.

9.3. Edukacja i kompetencje

- rozwijanie umiejętności współpracy z narzędziami AI,
- łączenie wiedzy projektowej z kompetencjami analitycznymi,
- świadome wykorzystywanie automatyzacji.

10. Propozycje zadania laboratoryjnego.

W ramach zajęć laboratoryjnych można zaproponować zadanie polegające na praktycznym wykorzystaniu AI do tworzenia diagramów UML.

10.1. Zakres zadania

- przygotowanie opisu systemu w formie tekstowej,
podany prompt (ręczny opis wymagania diagramu).
wykorzystanie sztucznej inteligencji do opisanie istniejących diagramów.
- wygenerowanie diagramu UML przy użyciu narzędzia AI,
- ręczna weryfikacja poprawności diagramu.

10.2. Cele dydaktyczne

- zrozumienie możliwości i ograniczeń AI,
- nauka analizy i korekty wygenerowanych modeli,
- rozwijanie umiejętności krytycznego myślenia projektowego.

10.3. Efekty końcowe

- poprawnie przygotowany diagram UML,
- raport z analizy działania narzędzia AI,
- wnioski dotyczące jakości wygenerowanej architektury.

11. Wnioski z przeprowadzonego eksperymentu

Przeprowadzony eksperyment potwierdza, że narzędzia AI mogą skutecznie wspierać proces tworzenia diagramów UML, jednak ich użycie wymaga:

- podstawowej wiedzy z zakresu notacji,
- umiejętności krytycznej analizy wyników,
- gotowości do iteracyjnej współpracy z modelem.

AI nie eliminuje potrzeby myślenia projektowego, lecz zmienia jego charakter: z manualnego rysowania na nadzorowanie, korektę i interpretację modeli.

12. Podsumowanie

Sztuczna inteligencja stanowi istotne wsparcie w procesie tworzenia diagramów UML, przyczyniając się do automatyzacji, poprawy spójności oraz skrócenia czasu projektowania. Jednocześnie jej wykorzystanie wymaga świadomego podejścia, uwzględniającego ograniczenia technologiczne oraz konieczność kontroli ze strony projektanta.

Połączenie wiedzy inżynierskiej z narzędziami AI może w przyszłości znacząco podnieść jakość dokumentacji projektowej oraz usprawnić proces wytwarzania oprogramowania, pod warunkiem zachowania równowagi pomiędzy automatyzacją a odpowiedzialnością projektową.

Narzędzia na dany moment w czasie dobrze sobie radzą przy odpowiedniej wielkości okna kontekstowego, o co będzie musiał zadbać przyszły projektant diagramów. Szala przesunie się z umiejętności tworzenia diagramów na ich opis i umiejętność stosowania poprawek. Element ludzki jeszcze przez długi czas będzie potrzebny do nadzorowania logiki, zwłaszcza dla bardziej skomplikowanych i mniej popularnych rozwiązań w systemach informatycznych lub biznesowych.