

Uniwersytet Gdański Wydział Matematyki, Fizyki i
Informatyki Instytut Informatyki

Oliver Gruba, Maciej Nasiadka

13 listopada 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
Specjalność	-
Nazwa ćwiczenia	Dynamiczny aspekt systemu informatycznego poprzez tworzenie behawioralnego diagramu czynności
Numer sprawozdania	4
Data zajęć	13.11.2025
Data oddania	19.11.2025
Miejsce na ocenę	

Spis treści

1	Cel diagramów czynności	3
2	Notacja i semantyka	3
3	Przykład 1 - Proces kupowania pojazdu samochodowego	5
4	Przykład 2 - System czytelní	6
5	Przykład 3 - Zadanie zespołowe	6
6	Zastosowania diagramów czynności	6
7	Wnioski	7

1. Cel diagramów czynności

Diagram czynności (ang. *Activity Diagram*) jest jednym z podstawowych diagramów behawioralnych UML, służącym do modelowania dynamicznych aspektów systemu. Jego głównym celem jest przedstawienie przepływu sterowania i danych w procesach biznesowych lub operacjach systemowych.

Diagram ten umożliwia:

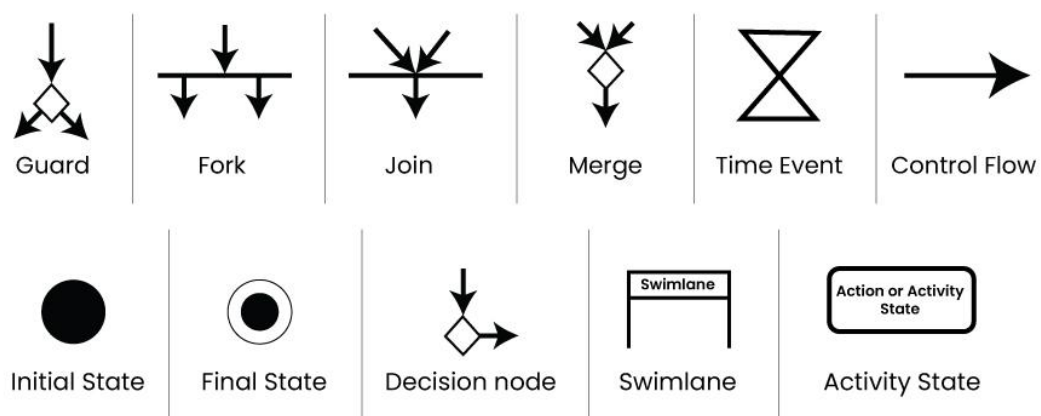
- analizę procesów biznesowych i logiki algorytmów,
- wizualizację sekwencji czynności wykonywanych w ramach przypadków użycia,
- identyfikację punktów decyzyjnych, rozgałęzień i współbieżności,
- ułatwienie komunikacji pomiędzy analitykami, projektantami i użytkownikami.

Diagramy czynności stanowią doskonałe narzędzie do opisu scenariuszy przypadków użycia oraz projektowania przepływów pracy (ang. *workflow*) w systemach informatycznych.

2. Notacja i semantyka

Na diagramach czynności stosuje się zestaw standardowych elementów UML, które pozwalają odwzorować logikę przepływu sterowania i danych:

- **Węzeł początkowy (Initial Node)** - punkt rozpoczęcia aktywności.
- **Czynność (Action)** - pojedynczy krok lub operacja wykonywana w procesie.
- **Węzeł decyzyjny (Decision Node)** - element rozgałęziający przepływ w zależności od warunków logicznych.
- **Węzeł rozwidlenia (Fork Node)** - rozpoczęcie wykonywania czynności równoległych.
- **Węzeł scalenia (Join Node)** - synchronizacja równoległych przepływów.
- **Węzeł końcowy przepływu (Flow Final Node)** - zakończenie części przepływu.
- **Węzeł końcowy (Activity Final Node)** - zakończenie całej aktywności.
- **Partycje aktywności (Swimlanes)** - wydzielenie czynności realizowanych przez różne role lub systemy.



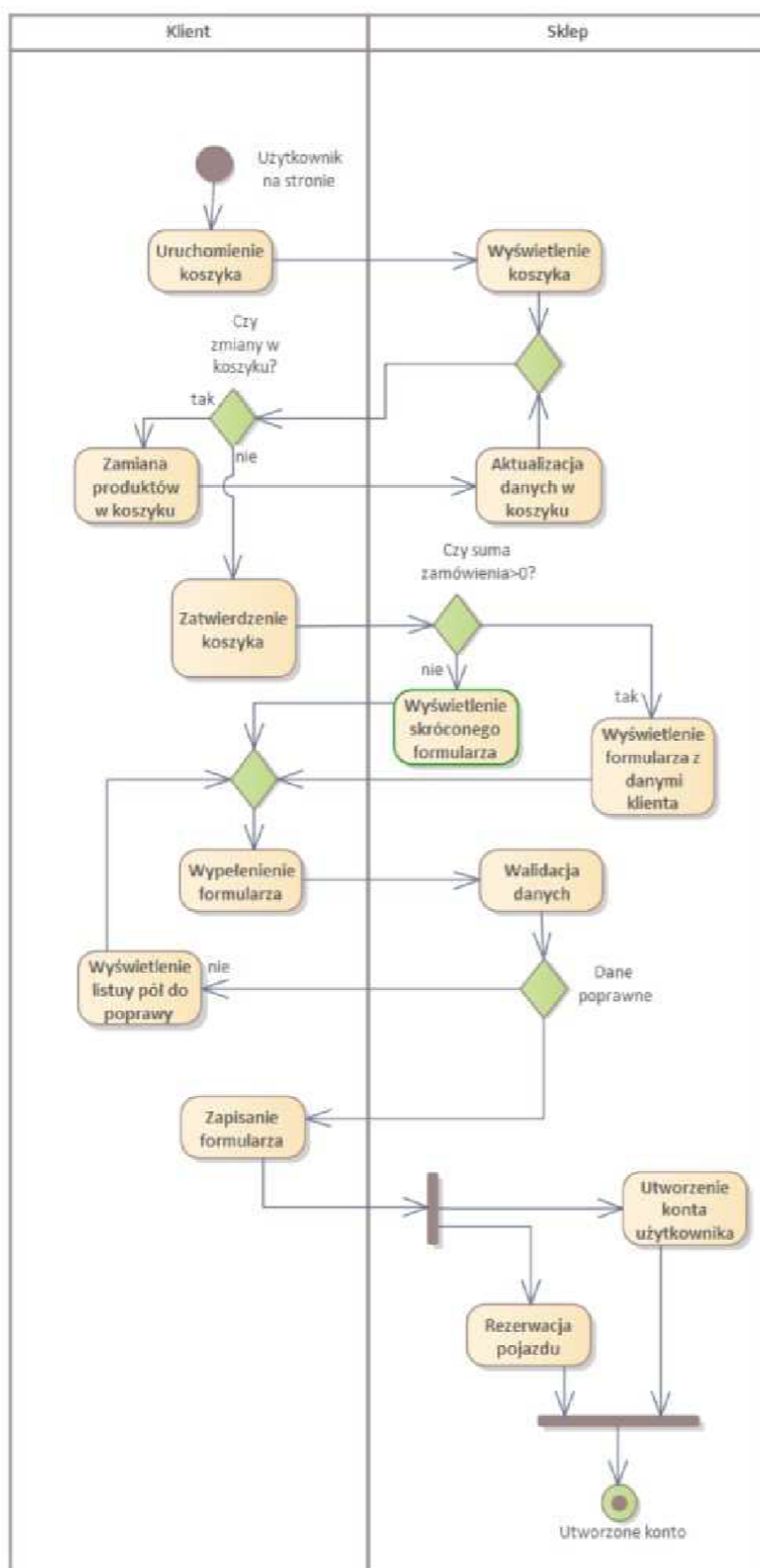
Activity Diagram Notations



Rysunek 1: Schemat przedstawienia graficznych elementów diagramu czynności

Elementy te łączy się za pomocą przepływów sterowania (*Control Flow*), które określają kolejność wykonywania czynności.

3. Przykład 1 - Proces kupowania pojazdu samochodowego



Rysunek 2: Diagram czynności - przedstawia proces zamawiania pojazdu przez użytkownika

Proces przedstawia —.

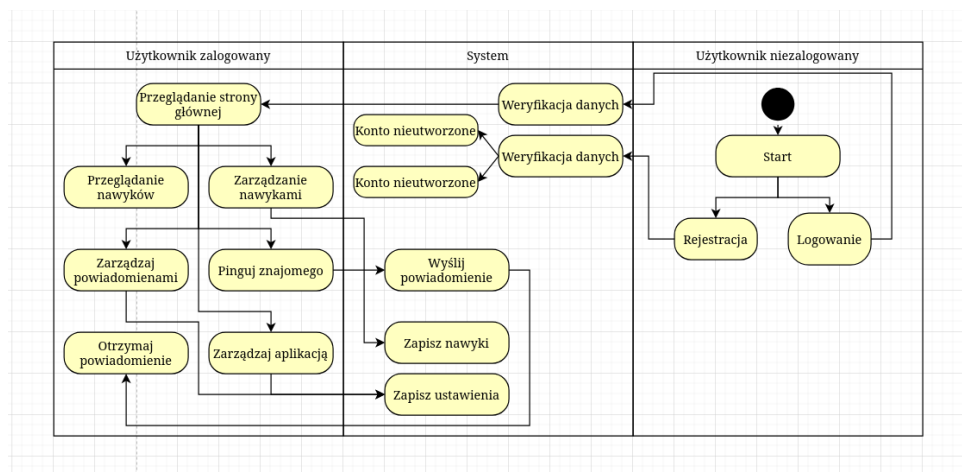
- Węzeł początkowy - —.
- Czynność: —.
- Czynność: —.
- Decyzja: —
 - Jeśli **tak**: przejście do —.
 - Jeśli **nie**: —.
- —.
- Węzeł końcowy - —.

4. Przykład 2 - System czytelní

Opis: Diagram przedstawia czynności w systemie czytelní.

5. Przykład 3 - Zadanie zespołowe

Opis: Diagram ilustruje czynności w projekcie Kairo Habit App.



Rysunek 3: Diagram czynności Kairo Habit App

6. Zastosowania diagramów czynności

Diagramy czynności znajdują szerokie zastosowanie w analizie i projektowaniu systemów informatycznych, w szczególności do:

- modelowania procesów biznesowych,
- dokumentowania przypadków użycia,
- analizy logiki programów i algorytmów,
- projektowania przepływów danych i sterowania w systemach,
- symulacji współbieżnych działań użytkowników i systemów.

7. Wnioski

Diagramy czynności stanowią ważny element procesu analizy systemów informatycznych. Pozwalają na:

- lepsze zrozumienie przepływu informacji i decyzji w systemie,
- identyfikację miejsc potencjalnych błędów lub nieefektywności,
- ułatwienie komunikacji pomiędzy zespołami projektowymi i biznesowymi.

W kontekście systemów informatycznych diagramy te wspierają planowanie wdrożeń, definiowanie procesów roboczych oraz zapewniają czytelne odwzorowanie dynamiki systemu przed implementacją. Stanowią nieodzowne narzędzie analityka i projektanta oprogramowania.