

Uniwersytet Gdańskie Wydział Matematyki, Fizyki i
Informatyki Instytut Informatyki

Oliver Gruba, Maciej Nasiadka

18 listopada 2025

Imię i Nazwisko (nr indeksu)	Oliver Gruba (292583) Maciej Nasiadka (292574)
Nazwa uczelni	Uniwersytet Gdańskie
Kierunek	Informatyka (profil praktyczny)
Prowadzący	dr inż. Stanisław Witkowski
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
2.1 Elementy podstawowe	4
2.2 Elementy kontrolne	4
2.3 Równoległość	5
2.4 Obiekty i dane	6
2.5 Semantyka	6
3 Przykład 1 - Proces kupowania pojazdu samochodowego	7
4 Przykład 2 - System czytelni	9
5 Przykład 3 - Zadanie zespołowe	11
6 Zastosowania diagramów czynności	12
6.1 Analiza i modelowanie wymagań	12
6.2 Projektowanie logiki biznesowej	12
6.3 Komunikacja w zespole	12
6.4 Testowanie	13
6.5 Optymalizacja procesów	13
7 Wnioski	13

1. Cel diagramów czynności

Diagram czynności (ang. *Activity Diagram*) jest jednym z podstawowych diagramów behawioralnych UML, służącym do modelowania dynamicznych aspektów systemu. Jego celem jest przedstawienie:

- przepływu działań wykonywanych w systemie,
- kolejności operacji,
- warunków decyzyjnych,
- równoległego wykonywania czynności,
- przepływu informacji oraz sterowania,
- interakcji użytkownika z systemem na poziomie procesów,
- reakcji systemu na zdarzenia wewnętrzne i zewnętrzne (np. zadziałanie przypomnień, wysłanie powiadomień).

Diagramy czynności stanowią doskonałe narzędzie do opisu scenariuszy przypadków użycia oraz projektowania przepływów pracy (ang. *workflow*) w systemach informatycznych. Pozwalają nie tylko opisać zachowanie systemu, ale też je przeanalizować pod względami jak:

- logiki biznesowej,
- przebiegu scenariuszy użytkownika (use cases),
- punktów optymalizacji procesów,
- potencjalnych błędów sekwencji lub niejednoznaczności funkcjonalnych,
- miejsc wymagających synchronizacji lub dodatkowej validacji.

W projektach zespołowych diagramy czynności pełnią również rolę komunikacyjną, między innymi pozwalając członkom zespołu szybko zrozumieć procesy, nawet jeśli nie są sztywnie odpowiedzialni za ich implementację.

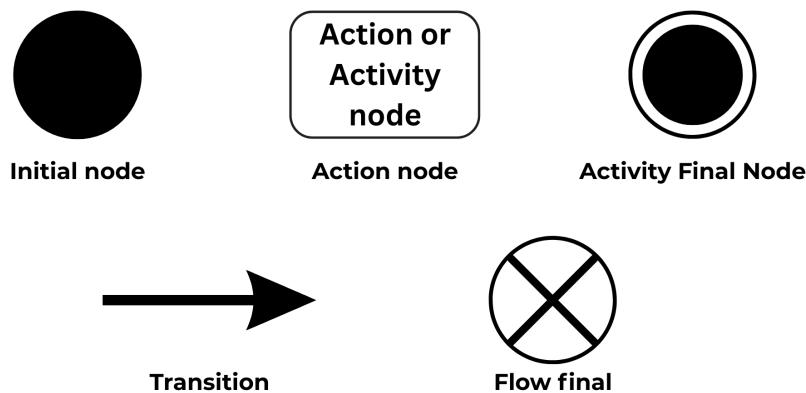
2. Notacja i semantyka

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

2.1. Elementy podstawowe

- Początek (Initial Node) — symbol wznowienia działania, zwykle jako czarne koło.
- Czynność (Action Node) — pojedynczy krok, operacja wykonywana przez system lub użytkownika.
- Przejście (Transition) — strzałka określająca przepływ sterowania między czynnościami.
- Zakończenie (Activity Final Node) — punkt kończący cały proces.
- Flow Final — punkt kończący tylko jedną z gałęzi przepływu.

Podstawowe elementy



Autor: Oliver Gruba

Rysunek 1: Graficzne przedstawienie podstawowych elementów diagramu: Initial node, Action node, Transition, Activity final node i Flow final

2.2. Elementy kontrolne

- Węzeł decyzyjny (Decision Node) — wybór jednej z możliwych ścieżek na podstawie warunku.
- Węzeł łączenia (Merge Node) — ponowne scalenie alternatywnych ścieżek.

Elementy kontrolne



Autor: Oliver Gruba

Rysunek 2: Graficzne przedstawienie elementów kontrolnych diagramu: Decision node i Merge node

2.3. Równoległość

- Fork Node — rozdzielenie jednego przepływu na wiele równoległych.
- Join Node — synchronizacja kilku równoległych ścieżek w jedną.

Równoległość



Autor: Oliver Gruba

Rysunek 3: Graficzne przedstawienie podstawowych elementów diagramu: Fork node i Join node

2.4. Obiekty i dane

- Object Node — reprezentuje obiekt, dokument lub dane przechodzące między czynnościami.
- Swimlane (tor aktywności) — podział diagramu na obszary reprezentujące odpowiedzialność uczestników (np. Użytkownik, System, Moduł Powiadomień).

Obiekty i dane



Autor: Oliver Gruba

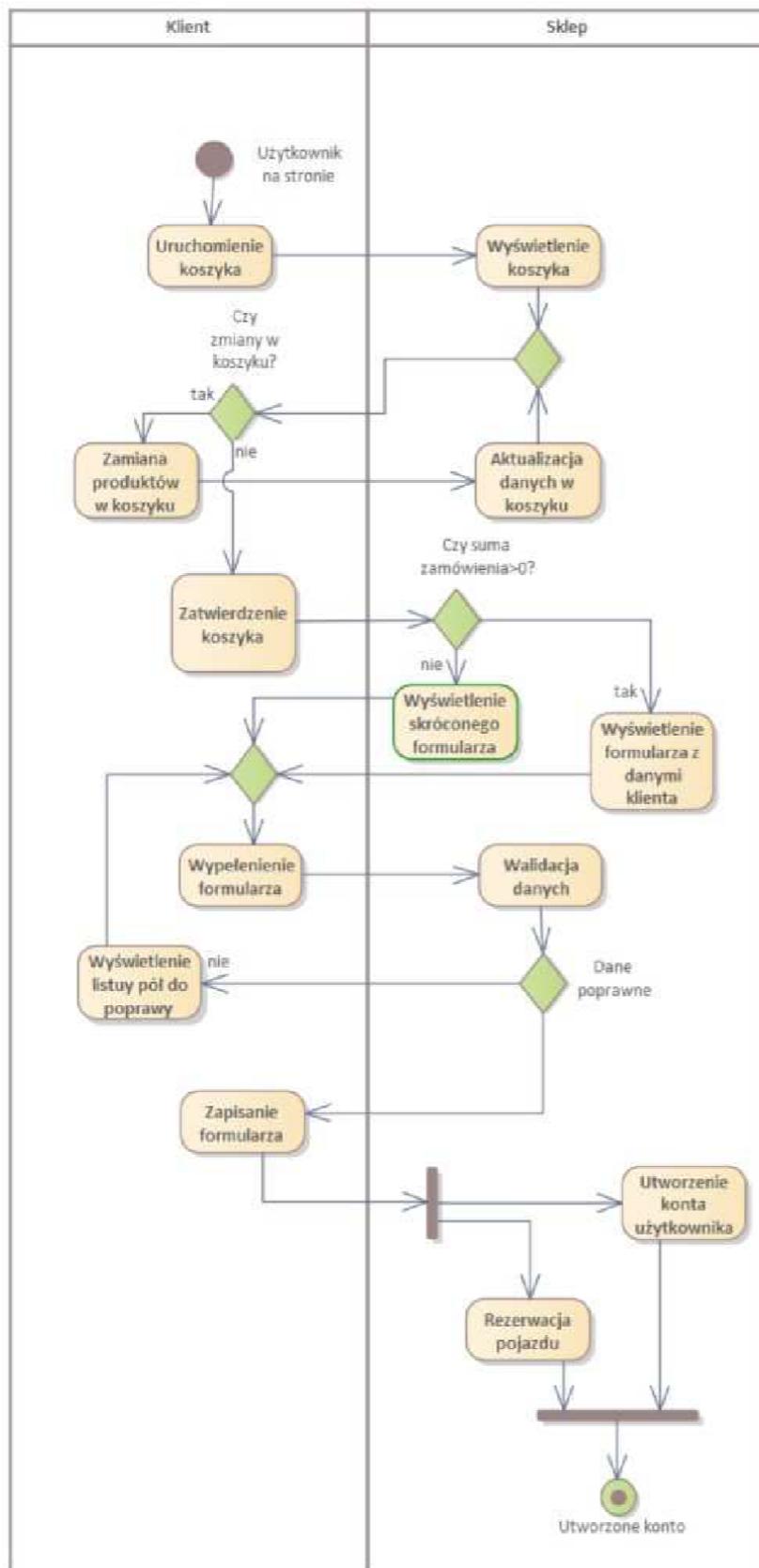
Rysunek 4: Graficzne przedstawienie podstawowych elementów diagramu: Object node i Swimlane

2.5. Semantyka

- Przepływ odbywa się sekwencyjnie lub równolegle w zależności od struktury.
- Czynność wykonuje się w atomicznie (w całości).
- Warunki decyzyjne muszą być wzajemnie rozłączne, chyba że diagram zakłada inaczej.
- Równoległość wymaga jawniej synchronizacji Join Node, aby uniknąć niespójności.

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

3. Przykład 1 - Proces kupowania pojazdu samochodowego



Rysunek 5: Diagram czynności - przedstawiający proces zamawiania pojazdu przez użytkownika

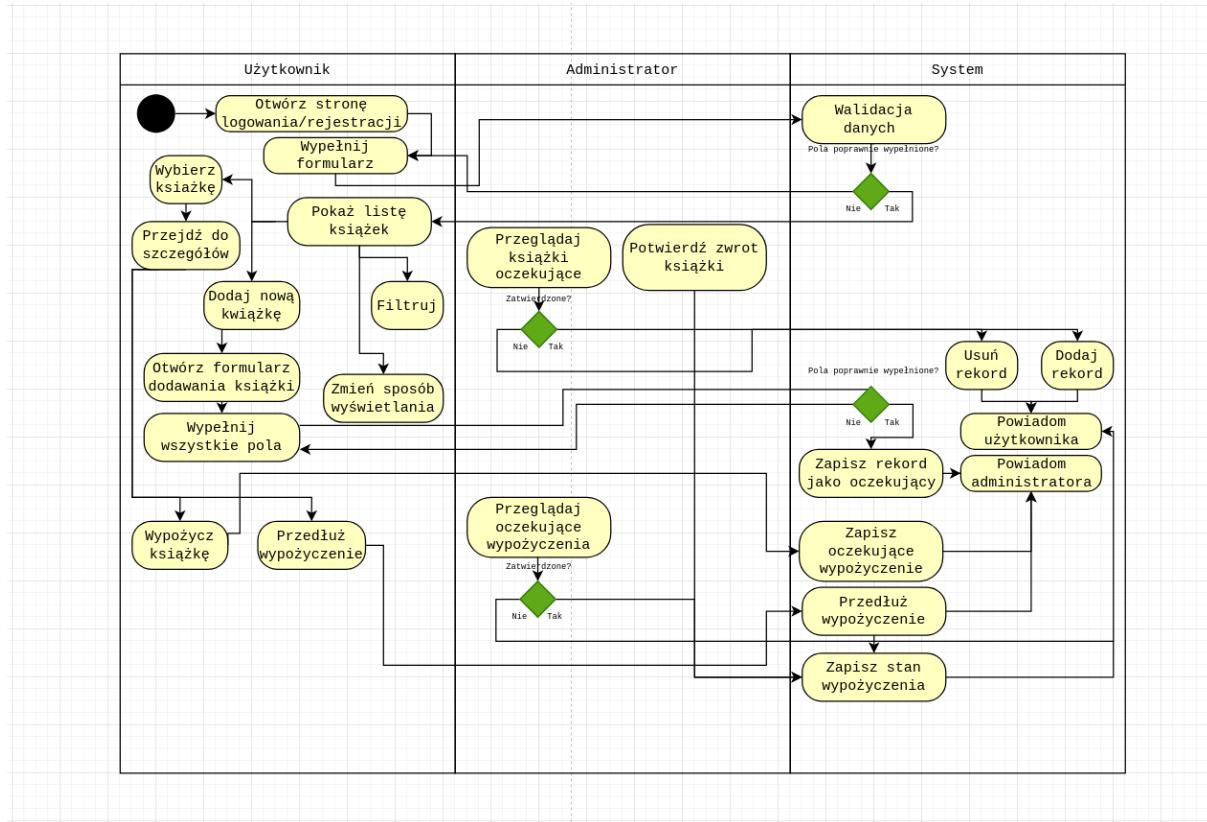
Proces przedstawia system zamawiania pojazdów samochodowych przez użytkownika.

- Initial node - *"Użytkownik na stronie"*.
- Action node: *"Uruchomienie koszyka"*.
- Action node: *"Wyświetlenie koszyka"*.
- Merge node:
 - *"Wyświetlenie koszyka"*: przejście do *Decision Node "Czy zmiany w koszyku?"*.
 - *"Aktualizacja danych w koszyku"*: przejście do *Decision Node "Czy zmiany w koszyku?"*.
- Decision node: *Czy zmiany w koszyku?>:*
 - Jeśli **tak**: przejście do *Action node: "Zamiana produktów w koszyku"*.
 - * Action node: *"Aktualizacja danych w koszyku"* przejście do *Merge node*
 - * Jeśli **nie**: przejście do *Action node: "Zatwierdzenie koszyka"*.
- Decision node: *"Czy suma zamówienia > 0?"*
 - Jeśli **tak**: przejście do *Action node: "Wyświetlenie formularza z danymi klienta"*.
 - Jeśli **nie**: przejście do *Action node: "Wyświetlenie skróconego formularza"*.
- Merge node:
 - *"Wyświetlenie formularza z danymi klienta"*: przejście do *Decision Node "Wypełnianie formularza"*.
 - *"Wyświetlanie skróconego formularza"*: przejście do *Decision Node "Wypełnianie formularza"*.
 - *"Wyświetlanie listy pól do poprawy"*: przejście do *Decision Node "Wypełnianie formularza"*.
- Action node: *"Wypełnianie formularza"*
- Action node: *"Walidacja danych"*
- Decision node: *"Dane poprawne?"*
 - Jeśli **tak**: przejście do *Action node: "Zapisanie formularza"*.
 - Jeśli **nie**: przejście do *Action node: "Wyświetlanie listy pól do poprawy"*.

- Action node: "Zapisanie formularza"
- Fork node: Action node: "Zapisanie formularza" przechodzi do:
 - Action node: "Utworzenie konta użytkownika".
 - Action node: "Rezerwacja pojazdu".
- Join node: Action nodes "Utworzenie konta użytkownika" i "Rezerwacja pojazdu" łączą się do:
 - Activity final node: "Utworzzone konto".
- Węzeł końcowy - "Utworzzone konto".

4. Przykład 2 - System czytelni

Opis: Diagram przedstawia czynności w systemie czytelni.



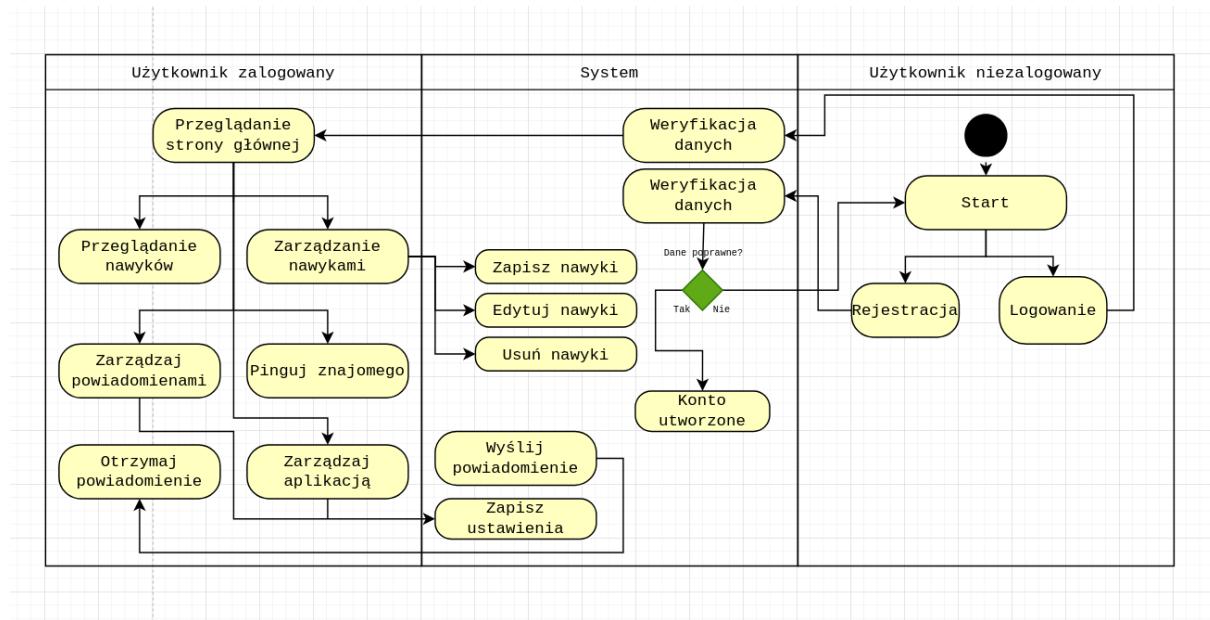
Rysunek 6: Diagram czynności systemu bibliotecznego

- Initial node - "Otwórz stronę logowania/rejestracji".

- Action node: "Wypełnij formularz".
- Action node: "Walidacja danych".
- Decision node: "Pola poprawnie wypełnione?"
 - Jeśli **tak**: przejście do Action node: "Pokaż listę książek".
 - Jeśli **nie**: przejście do Action node: "Wypełnij formularz".
- Action node: "Pokaż listę książek".
- Action node: "Filtruj".
- Action node: "Zmień sposób wyświetlania".
- Action node: "Wybierz książkę".
 - Action node: "Przejdź do szczegółów".
 - * Action node: "Wypożycz książkę".
 - Action node: "Zapisz oczekujące wypożyczenie".
 - Action node: "Powiadom administratora".
 - * Action node: "Przedłuż wypożyczenie".
 - * Action node: "Zapisz stan użytkownika".
 - * Action node: "Powiadom użytkownika".
- Action node: "Dodaj nową książkę".
 - Action node: "Otwórz formularz oddawania książek".
 - Action node: "Wypełnij wszystkie pola".
 - Decision node: "Pola poprawnie wypełnione?"
 - * Jeśli **tak**: przejście do Action node: "Zapisz rekord jako czekający".
 - * Action node: "Powiadom administratora".
 - * Jeśli **nie**: przejście do Action node: "Wypełnij wszystkie pola".
 - Action node: "Przeglądaj książki oczekujące".
 - Decision node: "Zatwierdzone?"
 - * Jeśli **tak**: przejście do Action node: "Dodaj rekord".
 - * Action node: "Powiadom użytkownika".
 - * Jeśli **nie**: przejście do Action node: "Usuń rekord".
 - * Action node: "Powiadom użytkownika".

5. Przykład 3 - Zadanie zespołowe

Opis: Diagram ilustruje czynności w projekcie Kairo Habit App.



Rysunek 7: Diagram czynności Kairo Habit App

- Initial node - "Start".
- Action node: "Uruchomienie koszyka".
- Action node: "Wyświetlenie koszyka".
- Action node: "Rejestracja".
- Action node: "Logowanie".
- Action node: "Weryfikacja danych".
- Decision node: "Dane poprawne?"
 - Jeśli **tak**: przejście do *Konto utworzone*.
 - Jeśli **nie**: przejście do *Initial node: "Start"*.
- Action node: "Przeglądarka strony głównej".
- Jeśli **nawyki**: przejście do *Przeglądanie nawyków*.
- Jeśli **zarządzanie**: przejście do Action node: "Zarządzanie nawykami".

- Decision node: "*Która opcja?*"
 - Jeśli **zapisz**: przejście do *Zapisz nawyki*".
 - Jeśli **edytuj**: przejście do *Edytuj nawyki*".
 - Jeśli **usuń**: przejście do *Usuń nawyki*".
- Jeśli **powiadomienia**: przejście do Action node: "*Zarządzanie powiadomieniami*".
- Jeśli **znajomi**: przejście do Action node: "*Pinguj znajomego*".
- Jeśli **aplikacja**: przejście do Action node: "*Zarządzanie aplikacją*".
- Action node: "*Zapisz ustawienia*".
- Action node: "*Wyślij powiadomienie*".
- Action node: "*Otrzymaj powiadomienie*".

Dla tego systemu przykład diagramu czynności pokazuje scenariusz użytkownika ustanawiającego nowy nawyk z przypomnieniem. Taki przypadek użycia łączy interakcję człowieka z automatycznymi procesami wewnątrz systemu.

6. Zastosowania diagramów czynności

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

6.1. Analiza i modelowanie wymagań

- Pozwalają dokładnie opisać scenariusze użytkownika.
- Umożliwiają wychwycenie brakujących kroków lub warunków.

6.2. Projektowanie logiki biznesowej

- Wizualizują przepływy danych i operacji.
- Pomagają projektantom określić odpowiedzialności modułów.

6.3. Komunikacja w zespole

- Ułatwiają współpracę między analitykami, projektantami i programistami.
- Eliminują nieporozumienia wynikające z interpretacji słownego opisu.

6.4. Testowanie

- Służą jako podstawa do tworzenia przypadków testowych.
- Pozwalają odwzorować ścieżki alternatywne i sytuacje wyjątkowe.

6.5. Optymalizacja procesów

- Pokazują miejsca, gdzie można wprowadzić asynchroniczność, skrócić proces, lub dodać walidację.

7. Wnioski

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

- pokazanie, jak system zachowuje się w odpowiedzi na działania użytkownika,
- wskazanie zależności czasowych i logicznych między operacjami,
- uchwycenie procesów współbieżnych,
- weryfikacja poprawności zaprojektowanych funkcji jeszcze przed implementacją.

W kontekście projektów zespołowych, takich jak opisywana aplikacja do nawyków (referencja: 5):

- diagramy czynności porządkują i formalizują pracę nad modułami,
- zwiększają spójność projektową,
- redukują ryzyko błędnych interpretacji,
- wspierają testowanie i wdrożenie.

Ostatecznie, w dobrze zaprojektowanym systemie informatycznym diagramy czynności pozwalają na klarowne odwzorowanie procesów biznesowych oraz zapewniają, że implementacja odpowiada rzeczywistym potrzebom użytkownika i założeniom funkcjonalnym.