



Modelowanie Systemu Informatycznego

prezentacja 4

Modelowanie zachowania – diagram czynności

wersja 1.0

dr inż. Paweł Głuchowski

Wydział Informatyki i Telekomunikacji, Politechnika Wroclawska

Treść prezentacji

1. Diagram czynności
2. Modelowanie czynności
3. Przykłady

1

Diagram czynności

Diagram czynności /activity diagram/ (niepoprawnie ale często zwany diagramem aktywności)

- Składa się z wykonywanych operacji: czynności i akcji.
- Modeluje algorytm i dynamikę systemu lub jego części:
 - **przepływy sterowania** – bezpośrednie przejścia między operacjami;
 - **przepływy obiektów** – przejścia między operacjami z przekazaniem obiektów między nimi.
- Na podstawie słownej specyfikacji wymagań oraz diagramów, np. przypadków użycia, można wykonać diagram czynności modelujący:
 - wykonanie przypadku użycia (scenariusz),
 - wykonanie przypadku testowania,
 - przebieg procesu systemowego,
 - algorytm operacji klasy.

Diagram czynności

Czynność /activity/

- **Złożona procedura realizacji określonego zachowania**, np.:
 - w procesie biznesowym: złożone zachowanie systemu (zadanie wykonywana przez modelowany system i/lub jego aktora);
 - w algorytmie programu: operacja klasy.
- Może się składać z:
 - czynności (innych lub tej samej – rekurencja);
 - akcji (niepodzielna część czynności).

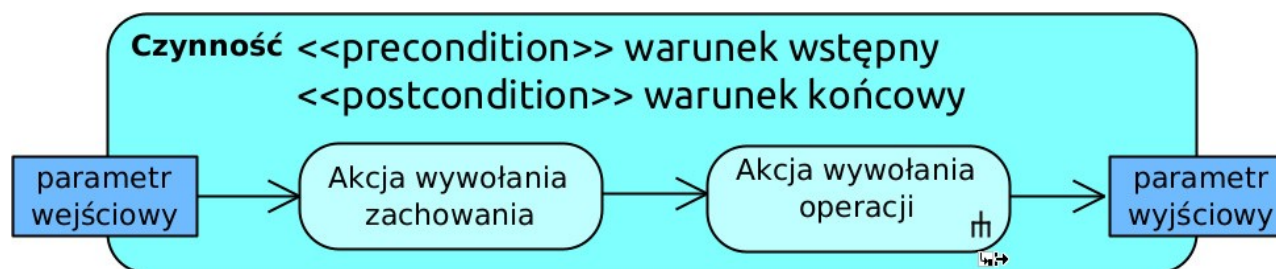


Diagram czynności

Czynność

- Może mieć **warunki wstępne** «*precondition*» – co musi być spełnione zanim czynność się zacznie.
- Może mieć **warunki końcowe** «*postcondition*» – co staje się spełnione do zakończenia czynności.
- Może mieć **parametry wejściowe** /input parameter node/ – obiekty danych do przetwarzania przez czynność (zwykle umieszczane na lewej krawędzi czynności).
- Może mieć **parametry wyjściowe** /output parameter node/ – obiekty danych wytworzonych przez czynność (zwykle umieszczane na prawej krawędzi czynności).

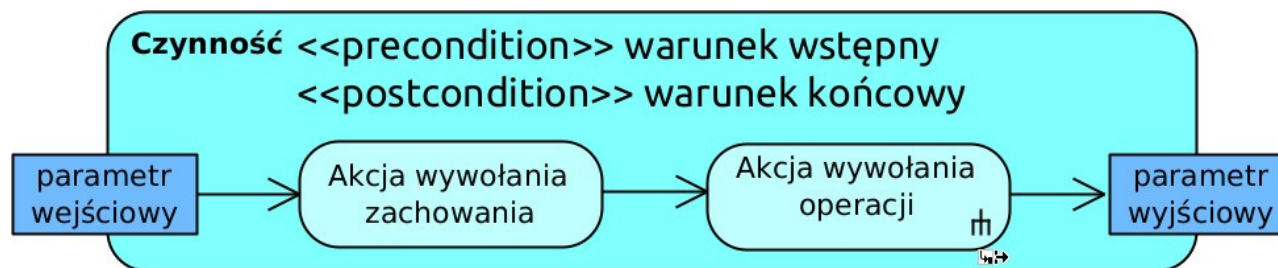


Diagram czynności

Akcja /action/

- Prosta i niepodzielna część czynności.
 - Czynność może zawierać akcje.
 - Akcja nie może zawierać czynności i akcji – jest atomowa:
 - nie podlega dekompozycji;
 - nie podlega przerwaniu.
- **Akcja wywołania zachowania** – przypisane jej zachowanie wykonuje sama,
- **Akcja wywołania operacji** (oznaczona „widłami”) – przypisane jej zachowanie przekazuje do wykonania czynność o tej samej nazwie.
 - Tę czynność przedstawia osobny diagram.

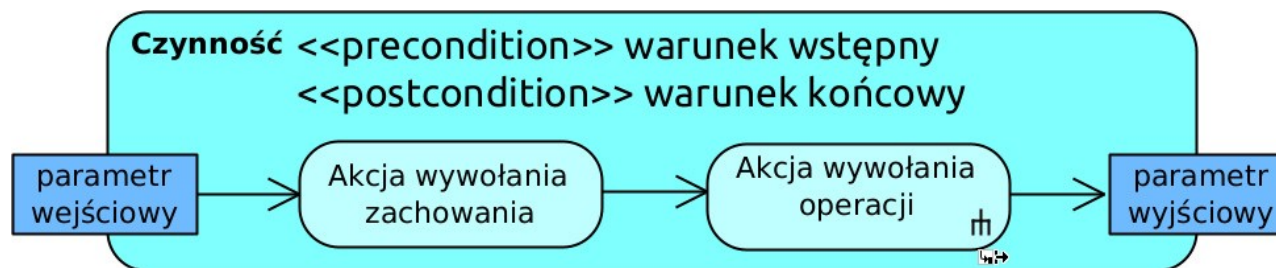


Diagram czynności

Akcja

- **Akcja oczekiwania na zdarzenie** /accept event action/
(prostokąt z trójkątnym wcięciem)
 - oczekuje na zajście zdarzenia, którego opis lub nazwę zawiera.
- **Akcja oczekiwania na czasowe zdarzenie** /accept time event action/
(klepsydra)
 - oczekuje na zajście zdarzenia czasowego, którego opis lub nazwę zawiera: kiedy lub jak często ono zachodzi.
- **Akcja wysłania sygnału** /send signal action/
(prostokąt z trójkątnym uwypukleniem)
 - tworzy i wysyła sygnał powodujący zdarzenie (np. wykonanie czynności lub stanu).

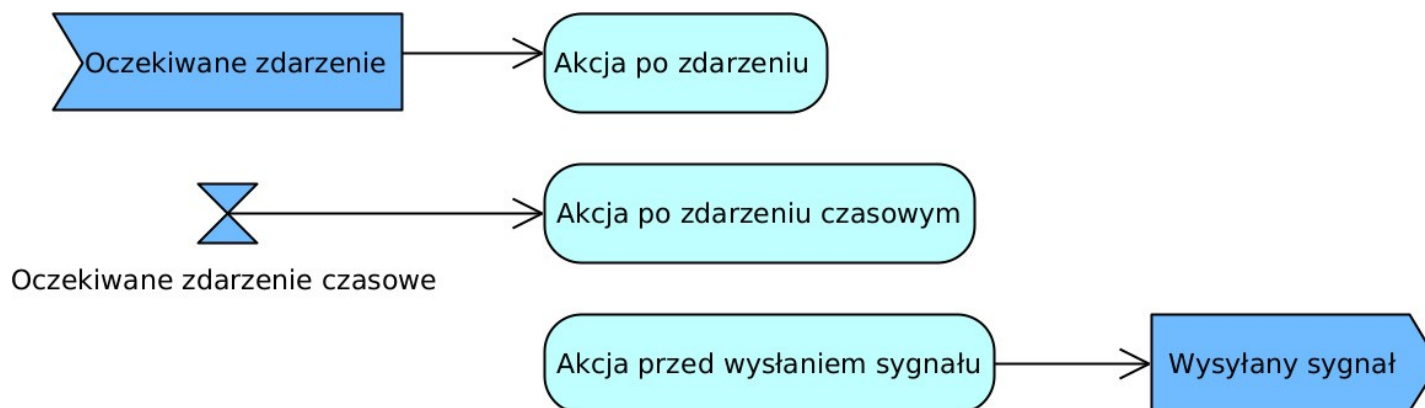


Diagram czynności

Przykład sekwencji czynności i akcji

- Przepływ sterowania fragmentu procesu wypłaty pieniędzy z bankomatu:
 - czynność **Wprowadź kwotę do wypłaty**,
 - akcja wysłania sygnału **Żądanie wypłaty**,
 - akcja oczekiwania na zdarzenie **Zgoda banku**,
 - czynność **Zabierz pieniądze**.

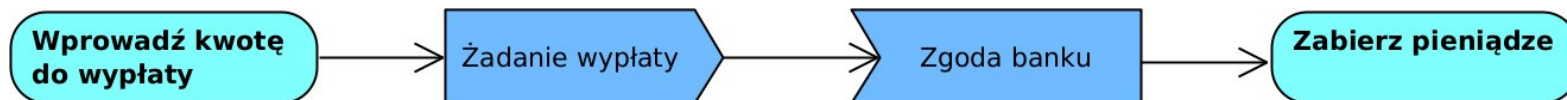


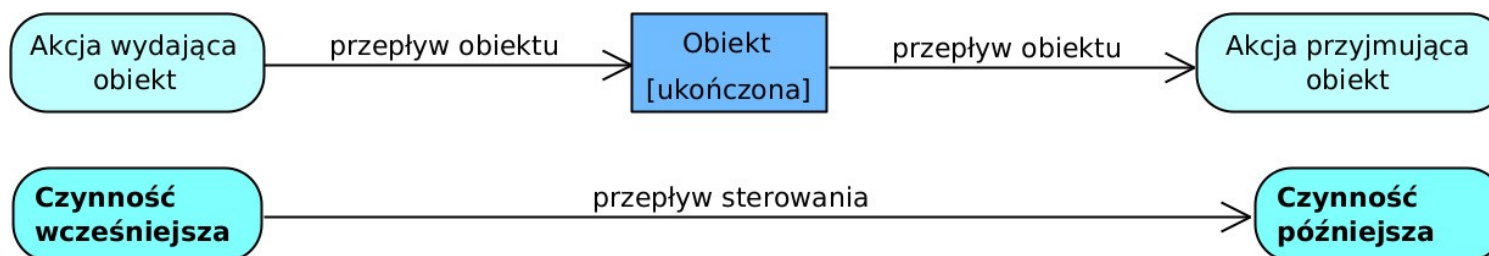
Diagram czynności

Obiekt /object/

- **Artefakt** (np. przetwarzana informacja, instancja klasy), który może być przedmiotem czynności i akcji jako ich dane wejściowe lub dane wyjściowe.
- Jest przekazywany między akcjami i czynnościami (przepływ obiektów).
- Bieżący **stan obiektu** podaje się w nawiasach [].
- Może być kontenerem obiektów.

Przepływ /flow/

- **Przepływ sterowania** /control flow/ – bezpośrednie przejście między dwiema czynnościami, dwiema akcjami lub czynnością a akcją.
- **Przepływ obiektu** /object flow/ – przejście między operacjami z przekazaniem obiektu między nimi.



Kontener

- **Repozytorium**, które przyjmuje, przechowuje i wydaje obiekty (np. kolekcja, baza danych).
 - Umieszczenie obiektu w kontenerze – przez wchodzący do niego przepływ obiektów.
 - Pobranie obiektu z kontenera – przez wychodzący z niego przepływ obiektów, opcjonalnie z parametrem *weight* i liczbą obiektów do pobrania, np.:
 - $\{weight = all\}$ – pobranie wszystkich obiektów,
 - $\{weight = *\}$ – pobranie dowolnej liczby obiektów.
- **Bufor centralny** (obiekt ze stereotypem «**centralBuffer**»)
 - Wydawany obiekt jest usuwany z kontenera.
- **Magazyn danych** (obiekt ze stereotypem «**datastore**»)
 - Wydawany obiekt NIE jest usuwany z kontenera.
 - Wydawana jest jego kopia lub referencja do niego.

Diagram czynności

Przykład bufora centralnego

na podst. [Unified Modeling Language \(UML\)](#)

- Dwie fabryki wytwarzają obiekty **Część**.
- Czynności **Wykonaj część w fabryce 1** i **Wykonaj część w fabryce 2** przekazują swój obiekt **Część** w stanie **ukończona** do bufora **Część**.
 - Bufor gromadzi te obiekty.
- Czynności **Pakuj części** i **Wykorzystaj części** pobierają z tego bufora obiekt **Część**:
 - każdy obiekt **Część** trafia tylko do jednej z tych czynności (losowo),
 - tylko niepobraný obiekt **Część** pozostaje w buforze.

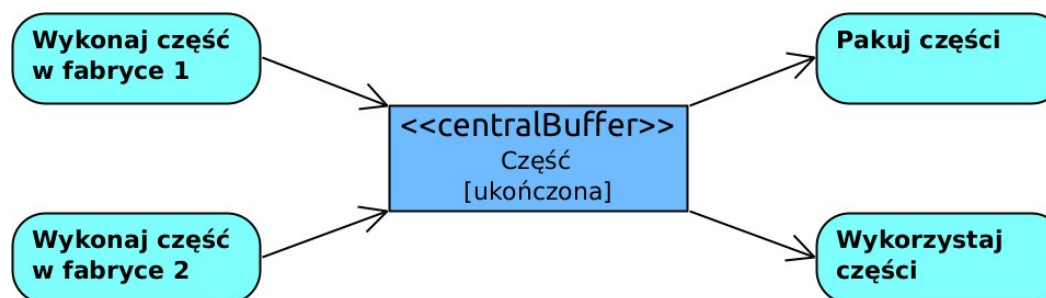


Diagram czynności

Przykład magazynu danych

na podst. [Unified Modeling Language \(UML\)](#)

- Czynność **Zatrudnij pracownika** umieszcza obiekty **Pracownik** w magazynie **Pracownik**.
 - Magazyn gromadzi te obiekty.
- Czynność **Przypisz pracownika do stanowiska** pobiera z magazynu obiekt **Pracownik**, jeśli ten obiekt nie ma przypisanego stanowiska («selection»)
 - pobrany obiekt **Pracownik** pozostaje w buforze.
- Czynność **Oceń pracownika** pobiera z magazynu dowolną liczbę (*{weight=*}*) obiektów **Pracownik** raz na rok (zdarzenie czasowe).
 - pobrany obiekt **Pracownik** pozostaje w buforze.

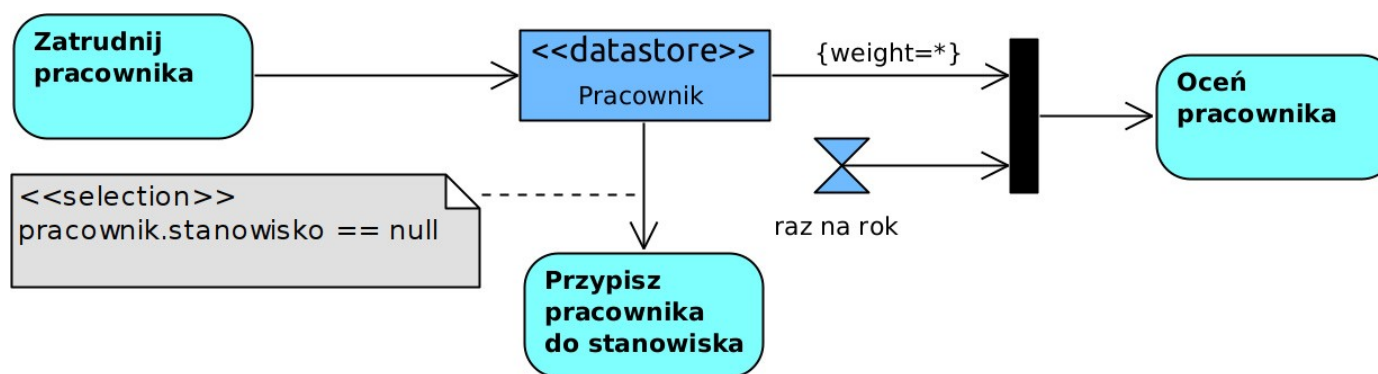


Diagram czynności

Węzeł decyzji

- Rozpoczyna i/lub kończy przebiegi alternatywne względem siebie.
- **Węzeł decyzji** /decision/ – wychodzą z niego alternatywne przebiegi.
 - Każda krawędź wychodząca ma warunek wyboru w nawiasach [].
 - Węzeł może mieć (jeśli to konieczne) opis decyzji: tekst przy węźle decyzji lub w notatce ze stereotypem «*decisionInput*».
 - Warunek i opis decyzji nie tworzą informacji, ale ją sprawdzają.
- **Węzeł złączenia** /merge/ – wchodzi do niego alternatywne przebiegi:
 - wychodzący z niego przebieg jest ich identycznym ciągiem dalszym.
- Węzeł złączenia jest też węzłem decyzji, gdy ma kilka wejść i kilka wyjść.

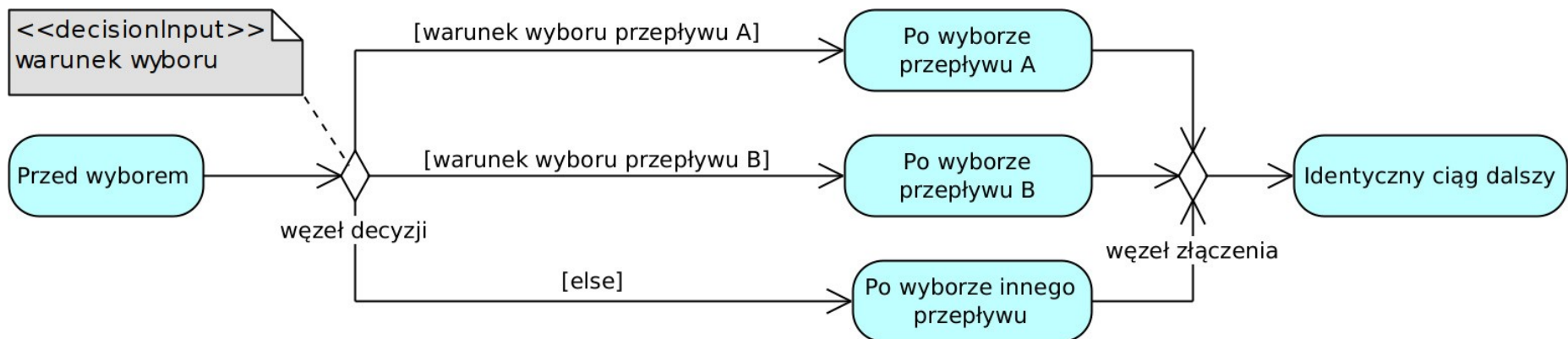


Diagram czynności

Węzeł współbieżności

- Rozpoczyna i/lub kończy przepływy współbieżne ze sobą.
- **Węzeł współbieżności** /fork/ – wychodzą z niego współbieżne przepływy.
- **Węzeł złączenia** /join/ – wchodzi do niego współbieżne przepływy.
 - Wychodzący z niego przepływ:
 - jest ich wspólnym ciągiem dalszym,
 - zaczyna się dopiero po skończeniu wszystkich wchodzących przepływów współbieżnych – węzeł synchronizacji.
- Węzeł złączenia jest też węzłem współbieżności, gdy ma kilka wejść i kilka wyjść.

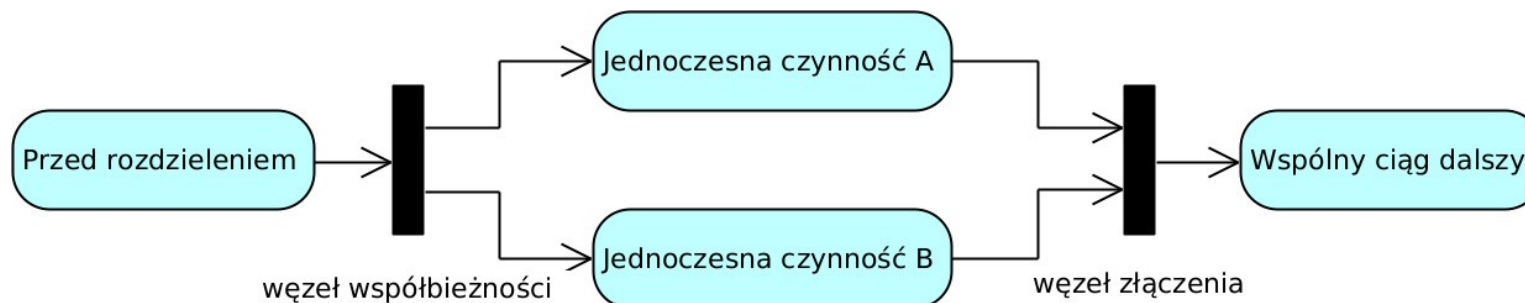


Diagram czynności

Węzeł początkowy /initial/

- Element diagramu lub czynności zawierającej akcje:
 - rozpoczyna jego/jej przepływ sterowania ,
 - wskazuje jego początkową czynność lub akcję.
- Diagram lub czynność zwykle ma tylko 1 węzeł początkowy.
 - Jeśli diagram lub czynność ma n węzłów początkowych, to wejście w ten diagram lub tę czynność rozpoczyna n współbieżnych przepływów sterowania.

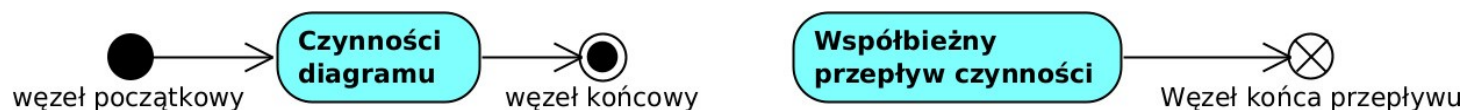


Diagram czynności

Węzeł końcowy /activity final/

- Element diagramu lub czynności zawierającej akcje:
 - kończy jego/jej przepływ sterowania,
 - przerywa jego/jej pozostałe nieukończone współbieżne przepływy.
- Diagram lub czynność może mieć wiele węzłów końcowych.

Węzeł końca przepływu /flow final/

- Element diagramu lub czynności zawierającej akcje:
 - kończy jego/jej przepływ sterowania,
 - NIE przerywa jego/jej pozostałych nieukończonych współbieżnych przepływów.
- Diagram lub czynność może mieć wiele węzłów końca przepływu.



Diagram czynności

Przykład węzłów końcowych i pętli

na podst. [Unified Modeling Language \(UML\)](#)

- Tworzenie aplikacji składa się z dwóch przepływów sterowania:
 - przepływ 1 z akcją **zbuduj komponent**,
 - przepływ 2 z akcjami **zainstaluj komponent** i **dostarcz aplikację**.
- Zapętłony przepływ 1 zawiera akcję **zbuduj komponent**.
- Przepływ 1 kończy się, gdy **brak komponentów do zbudowania**.
 - nie przerywa to przepływu 2 (rozpoczętego w poprzedniej iteracji pętli).

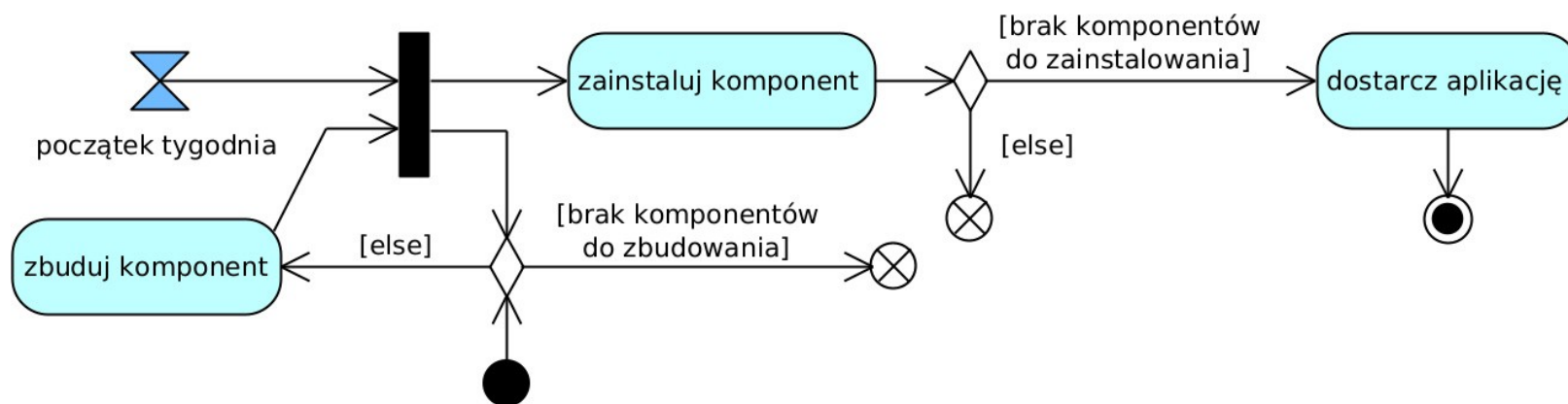


Diagram czynności

Przykład węzłów końcowych i pętli

na podst. [Unified Modeling Language \(UML\)](#)

- Po zejściu **początku tygodnia** i po zbudowaniu komponentu rozpoczyna się przepływ 2 zawierający akcje **zainstaluj komponent** i (opcjonalnie) **dostarcz aplikację**.
- Przepływ 2 kończy się akcją **dostarcz aplikację**, gdy **brak komponentów do zainstalowania**:
 - przerywa to przepływ 1 (bieżącą iteracją pętli).
- W innym wypadku przepływ 2 kończy się bez tej akcji:
 - nie przerywa to przepływu 1 (bieżącej iteracji pętli).

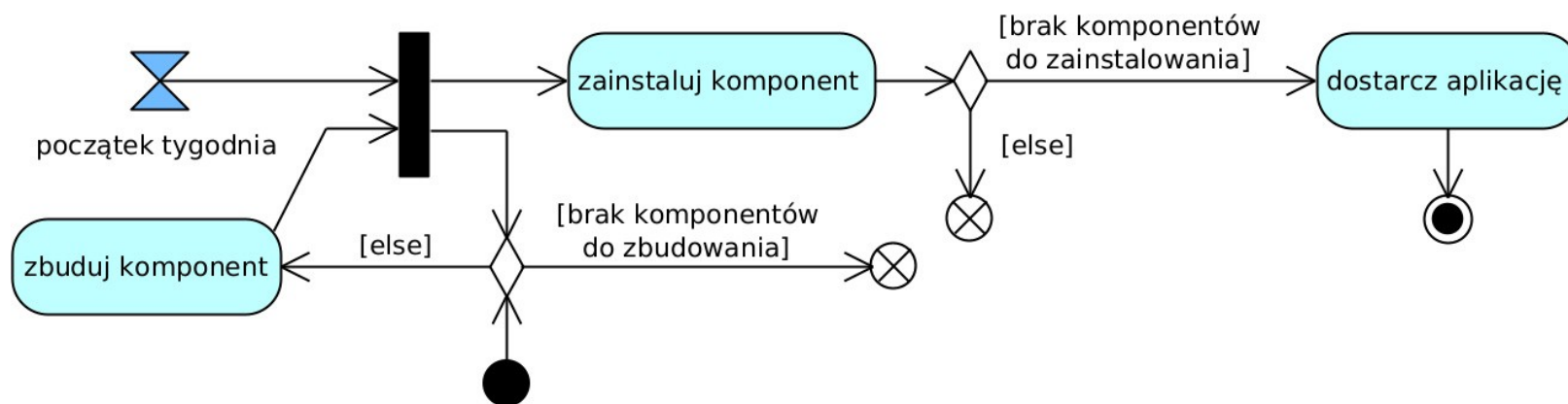


Diagram czynności

Partycja /partition/
(dawniej zwana **torem** /swimlane/)

- Segregacja czynności i akcji (NIE dotyczy obiektów):
 - na różnym poziomie abstrakcji
(np. na podstawie udziału w nich aktorów, miejsca ich wykonania, ich implementacji).
- Każda czynność i akcja musi być w konkretnej partycji.
- Obiekt może być w konkretnej partycji lub na granicy między nimi.
- Podział na partycje może być pionowy lub poziomy.
- Partycje mogą być zagnieżdżone.

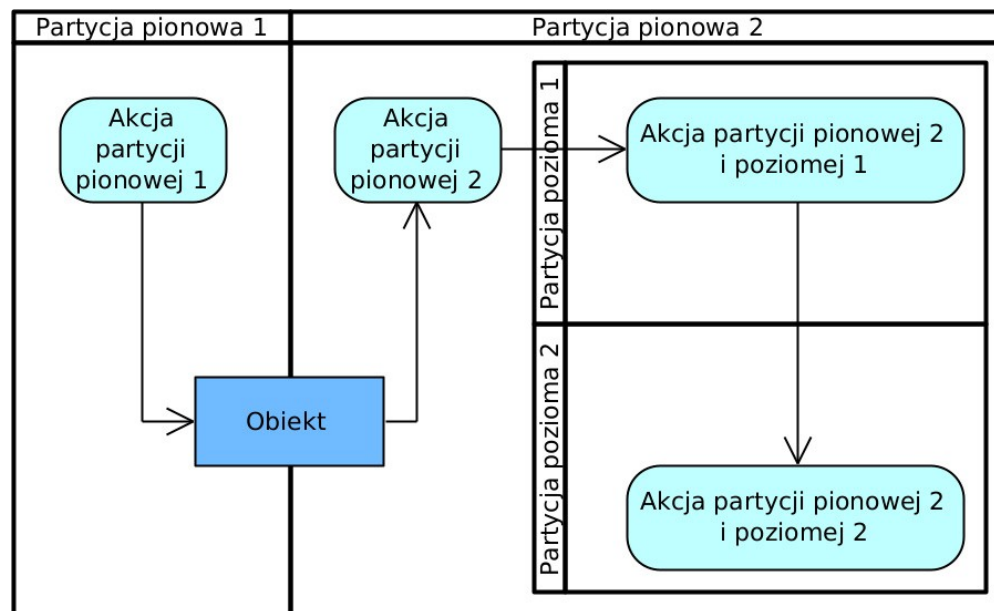


Diagram czynności

Przykład partycji i węzłów decyzji i współbieżności

na podst. [Unified Modeling Language \(UML\)](#)

- Czynność **Przetwarzaj zamówienie** zaczyna się akcją **Przyjmij zamówienie** z parametrem wejściowym **Zamówienie**.
- Akcja **Wyślij paczkę** jest współbieżna z przepływem z akcjami: **Wyślij fakturę**, **Zatwierdź fakturę** i **Zatwierdź opłatę**.
- Akcja **Wyślij fakturę** przekazuje obiekt **Faktura** do akcji **Opłać fakturę**.
- Akcja **Opłać fakturę** wykonywana jest przez **Klienta**, inne – przez **Sklep**.

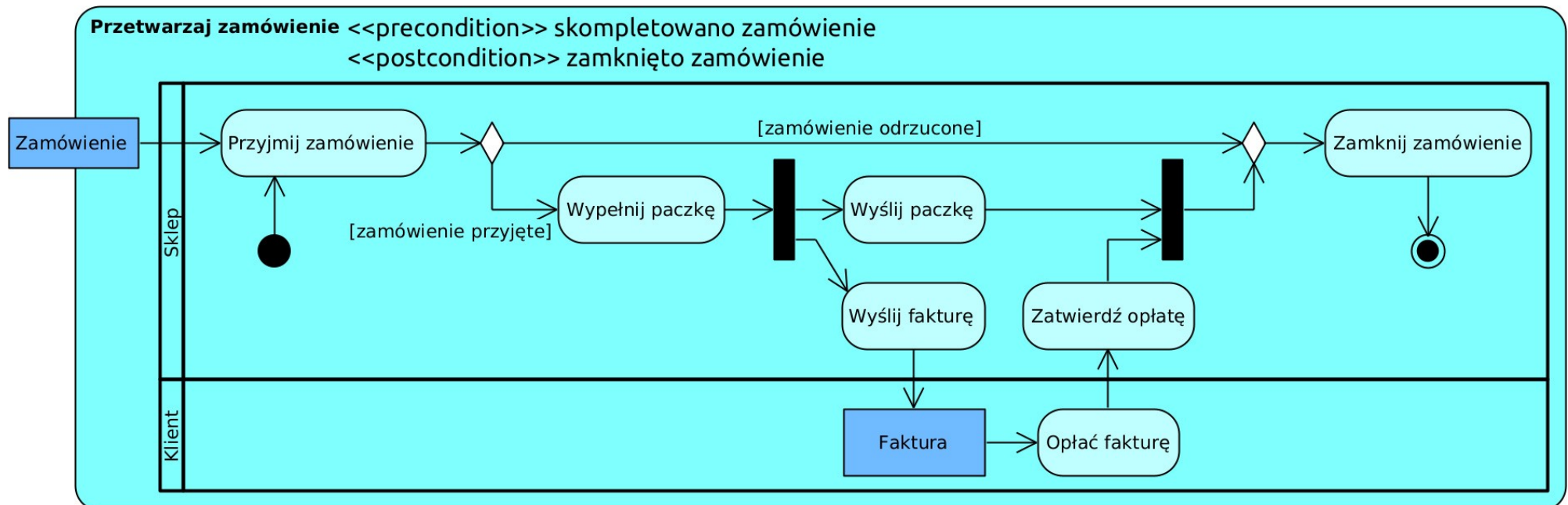


Diagram czynności

Obszar przerywalny /interruptible region/

- **Sekcja krytyczna i obsługa zdarzenia przerwania lub wyjątku.**
- Zawiera akcję oczekiwania na to zdarzenie.
 - Wychodzi z niej przepływ obsługi przerwania /exception handler/ (ma kształt pioruna lub obok symbol pioruna).
 - Przepływ ten wchodzi do czynności lub akcji obsługi przerwania.
- Zawiera czynności i akcje, w czasie których może zająć to zdarzenie.
 - Innych czynności i akcji NIE zawiera.

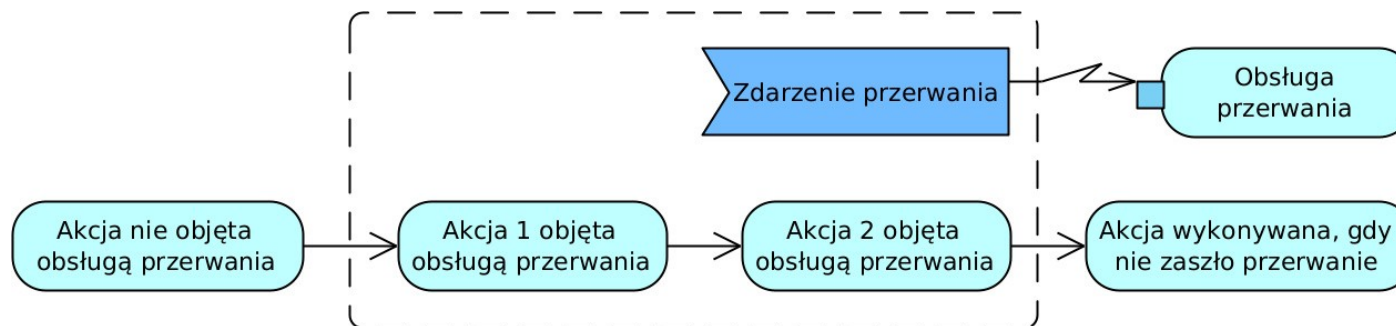


Diagram czynności

Obszar rozszerzenia /expansion region/

- Modeluje **przetwarzanie każdego obiektu z kolekcji obiektów**.
- Zawiera przepływy z czynnościami i akcjami przetwarzania 1 obiektu z wykorzystaniem odpowiednich węzłów.
- Przepływ wchodzący do tego obszaru wchodzi przez wierzchołek wejściowy /input expansion node/ z nazwą wejściowej kolekcji obiektów.
- Przepływ wychodzący z tego obszaru wychodzi przez wierzchołek wyjściowy /output expansion node/ z nazwą wyjściowej kolekcji obiektów.
- Może być kilka wierzchołków wejściowych i/lub wyjściowych.

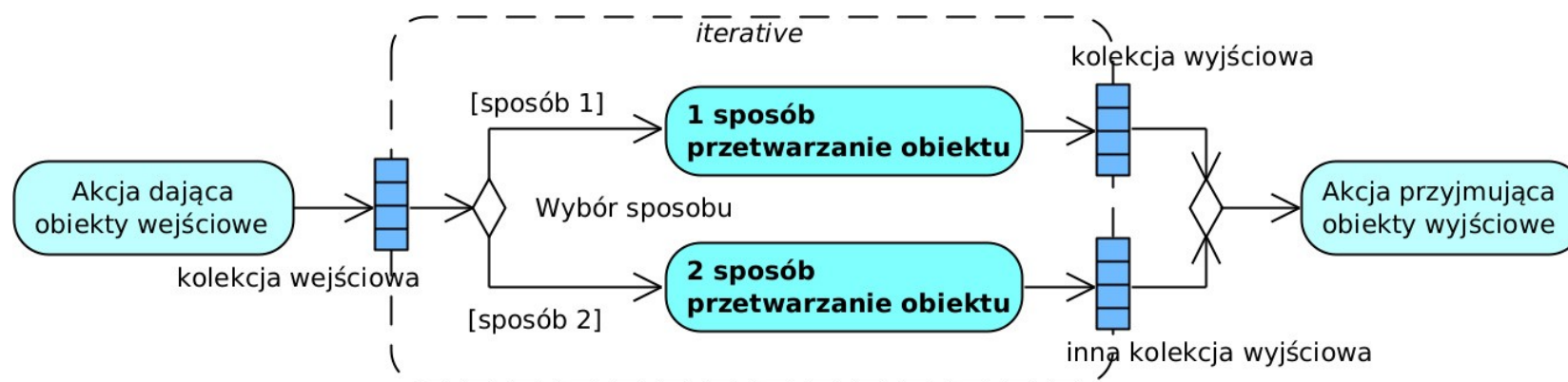


Diagram czynności

Obszar rozszerzenia

- Sposób przetwarzania obiektów podaje się u góry obszaru:
 - **szeregowo («iterative»)** – otrzymanie całej kolekcji obiektów, a następnie kolejne ich przetworzenie: pierwszy, drugi itd.;
 - **strumieniowo («stream»)** – kolejne przetworzenie obiektów: pierwszy, drugi itd., w miarę ich otrzymywania;
 - **współbieżnie («parallel»)** – otrzymanie całej kolekcji obiektów, a następnie współbieżne ich przetworzenie: pierwszy, z drugim itd.

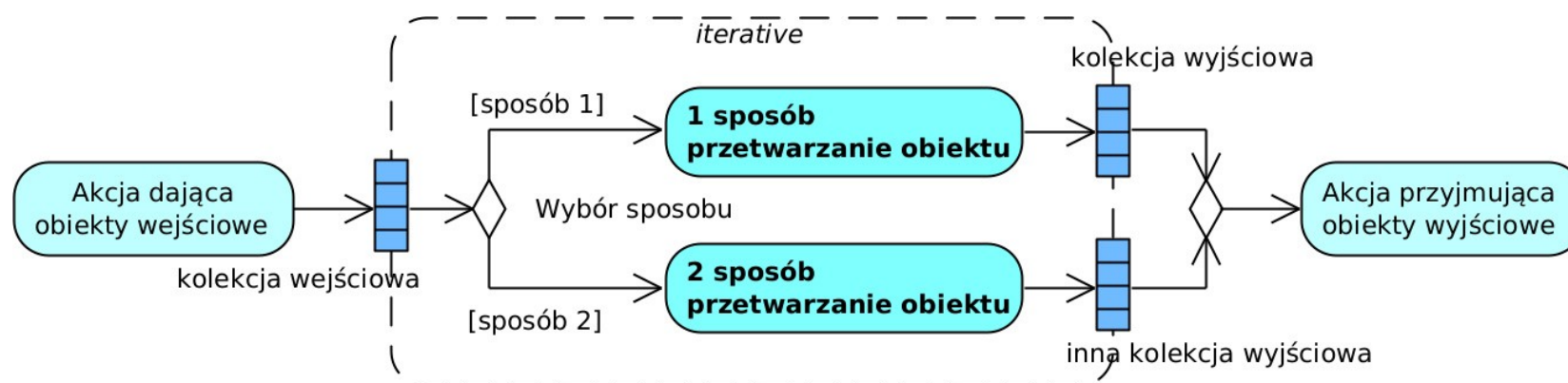
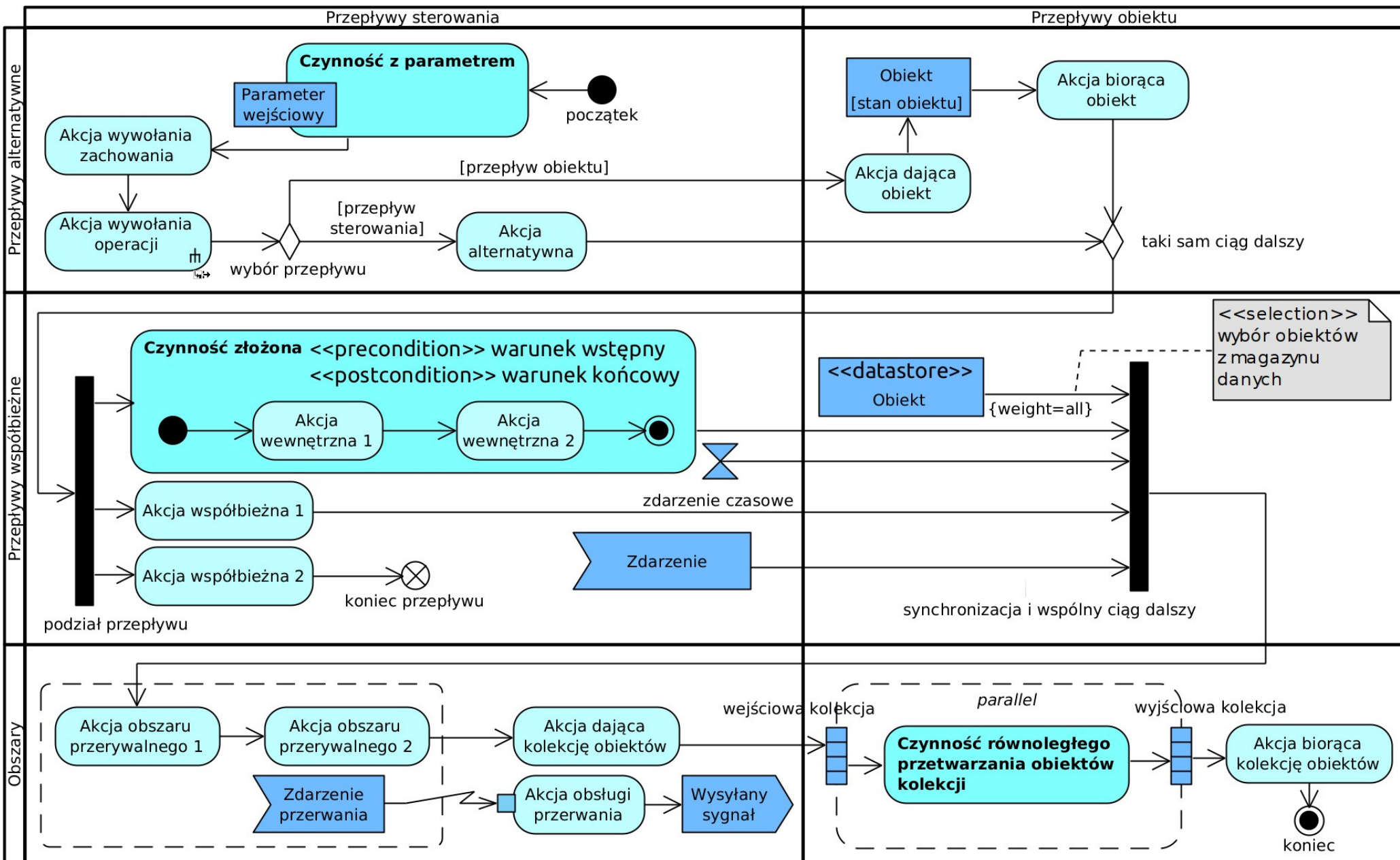


Diagram czynności

Podsumowanie



2

Modelowanie czynności

Elementy modelowania diagramu czynności

1. Zdefiniuj podstawowe czynności:

- Np. na podstawie scenariuszy przypadków użycia.
- Pomiń inne czynności (np. z innych przypadków użycia).
- Zdefiniuj stan początkowy i stan końcowy czynności.
- Kolejne czynności definiuj, idąc od stanu początkowego do końcowego.

2. Zdekomponuj czynności na ich składowe czynności i akcje.

- Małą czynność złożoną:
 - zdefiniuj jako czynność zawierającą jej czynności i akcje składowe.
- Dużą lub powtarzającą się czynność złożoną:
 - pokaż jak prostą czynność (bez czynności i akcji składowych),
 - jej czynności i akcje składowe pokaż na osobnym diagramie.

3. Zdefiniuj obiekty przetwarzane przez czynności i kontenery danych.

- Wykonaj analizę zmienności stanu obiektu pod wpływem jego przetwarzania przez zdefiniowane czynności.

Elementy modelowania diagramu czynności

4. Połącz czynności i akcje w przepływy sterowania:

- Wykonaj analizę zależności między czynnościami i akcjami.
- Zastosuj odpowiednie węzły i regiony do budowy przepływów alternatywnych i współbieżnych:
 1. Zbuduj sekwencyjne przepływy sterowania.
 2. Gdzie trzeba, połącz przepływy sterowania w przepływy alternatywne.
 3. Gdzie trzeba, połącz przepływy sterowania w przepływy współbieżne.

5. Dołącz przepływy obiektów:

- Połącz obiekty z przetwarzającymi je czynnościami i akcjami.
- Uwzględnij bieżący stan obiektu, jeśli się zmienia.

Elementy modelowania diagramu czynności

6. Zdefiniuj partycje:

- Np. dla procesu biznesowego:
 - dla aktorów wykonujących czynności i akcje;
 - dla miejsc wykonywania czynności i akcji.
- Np. dla opisu algorytmu programu:
 - dla klas wykonujących czynności i akcje;
 - dla warstw oprogramowania wykonujących czynności i akcje.

7. Rozmieść czynności i akcje w dotyczących ich partycjach.

- Zagnieżdżaj partycje różnego przeznaczenia.

Elementy modelowania diagramu czynności (dla modelowania algorytmu operacji klasy)

- **Uwzględnij:**
 - parametry przekazywane do tej operacji,
 - atrybuty klasy tej operacji,
 - klasy obiektów związanych z tą operacją.
- **Zdefiniuj warunki wstępne i końcowe tej operacji.**
- **Zdefiniuj wartości, których ta operacja nie zmienia.**
- **Stosuj przepływy alternatywne do modelowania:**
 - instrukcji warunku,
 - instrukcji pętli.
- **Stosuj przepływy współbieżne do modelowania:**
 - dla klasy aktywnej (kontroluje swoje wykonanie) i w wielowątkowości,
 - do modelowania instrukcji współbieżnych.

Uwaga!

- **Definiowanie czynności i akcji musi być zrozumiałe, ustandaryzowane i wewnętrznie spójne:**
 - W procesie biznesowym (np. scenariusz przypadku użycia):
 - tryb oznajmujący, np. *Analiza jakości produktu*;
 - tryb rozkazujący, np. *Przeanalizuj jakość produktu*.
 - W opisie algorytmu programu (np. operacji klasy):
 - pseudokod w języku naturalnym,
 - pseudokod lub kod w języku programowania.
- **Diagram czynności powinien być możliwie jak najprostszy i wizualnie czytelny:**
 - W procesie biznesowym: mała szczegółowość.
 - W opisie algorytmu programu: szczegółowość zależy od etapu modelowania.

3

Przykłady

Przykłady

zobacz: [Inżynieria Oprogramowania](#), Z. Kruczkiewicz, PWr, wykład 3

1. Diagramy czynności. System sporządzania rachunków. (strony 20—29).
2. Przykład diagramów czynności UML – modelowanie operacji. (strony 31—33).

Temat następnej prezentacji

Modelowanie zachowania – diagram stanów