

Podstawy projektowania systemów komputerowych

Diagramy sekwencji

Widok logiczny



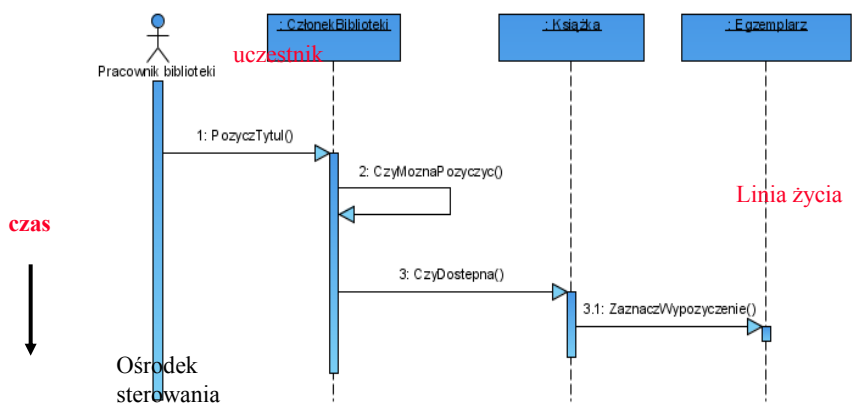
- Używany do modelowania części systemu oraz sposobów, w jaki one ze sobą współdziałają.
- Ten widok zazwyczaj tworzą diagramy:
 - **Klas**,
 - Obiektów,
 - Maszyny stanowej,
 - Interakcji:
 - **Sekwencji**,
 - Komunikacji,
 - Czasowe,
 - Przeglądowe diagramy interakcji.

Diagramy interakcji

UML zawiera cztery rodzaje diagramów interakcji modelujące interakcje zachodzące w czasie działania systemu pomiędzy jego częściami (obiektami), które to części wchodzą w skład widoku logicznego modelu:

- **Diagramy sekwencji** – opisująymi interakcje pomiędzy częściami systemu w postaci sekwencji komunikatów wymienianych między nimi
- **Diagramy komunikacji** – specyfikujące strukturalne związki pomiędzy biorącymi udział w interakcji częściami oraz wymianę komunikatów pomiędzy tymi instancjami
- **Diagramy czasowe** – reprezentują na osi czasu zmiany dopuszczalnych stanów, jakie może przyjmować uczestnik w interakcji
- **Przeglądowe diagramy interakcji** – udostępniają wysokiego poziomu widok wzajemnej współpracy kilku interakcji wykorzystywanych w celu implementacji pewnej części systemu, na przykład danego przypadku użycia.

Diagramy sekwencji

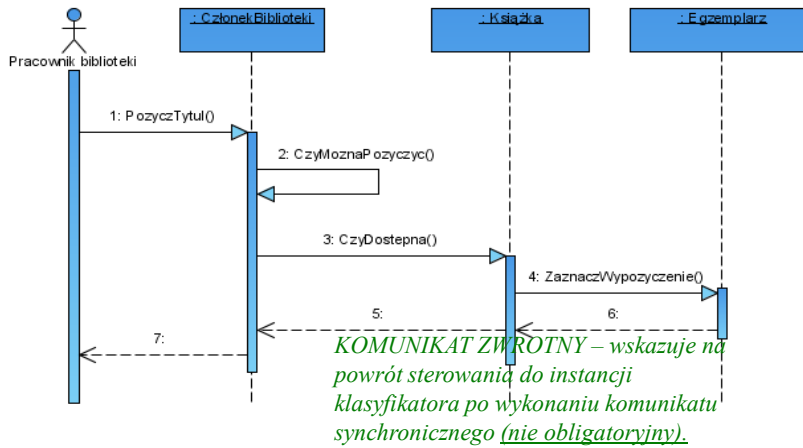


Komunikaty synchroniczne – oznaczają przekazanie sterowania do odbiorcy.

W momencie przesłania tego komunikatu aktualny przepływ sterowania nadawcy ulega przerwaniu. Jest on wznowiany dopiero po wykonaniu przez odbiorcę operacji inicjowanej przez ten komunikat.

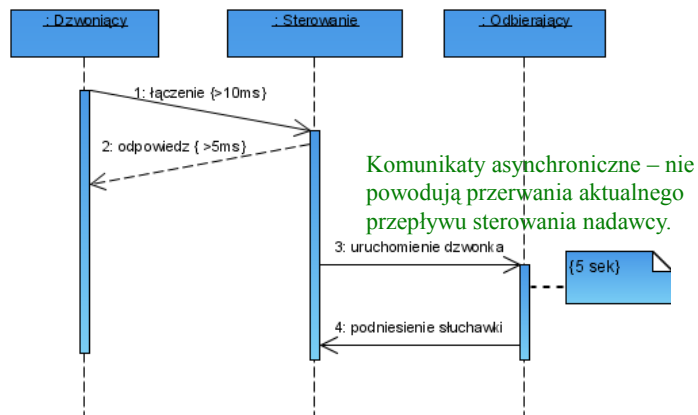
Ilustracja przekazywania sterowania

KOMUNIKAT (sygnał) – należy traktować jako zdarzenie przekazywane od nadawcy w celu wymuszenia u odbiorcy wykonania pewnej czynności.

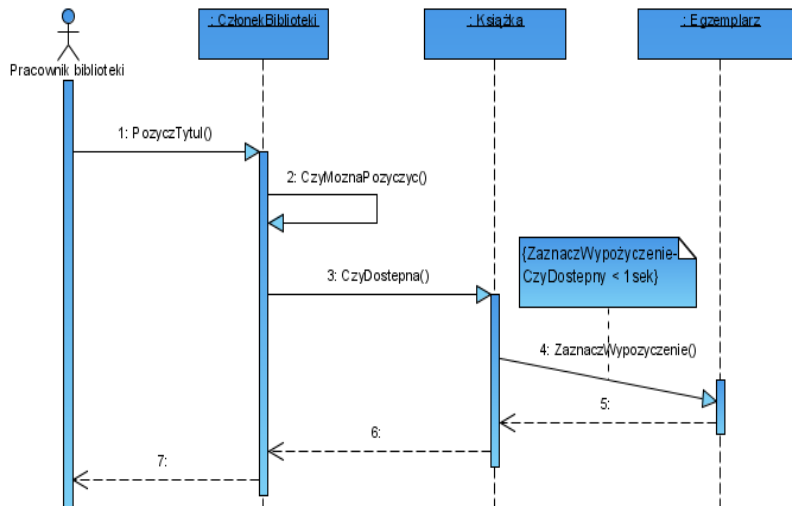


Nakładanie ograniczeń na przepływ czasu (1)

Główna przewaga diagramów sekwencji nad diagramami komunikacji przejawia się w ich zdolności do **graficznego prezentowania przepływu czasu**, a nawet do podawania ograniczeń czasowych, czy też skali czasowej. Taka możliwość może mieć duże znaczenie dla opisu systemów czasu rzeczywistego.



Nakładanie ograniczeń na przepływ czasu (2)

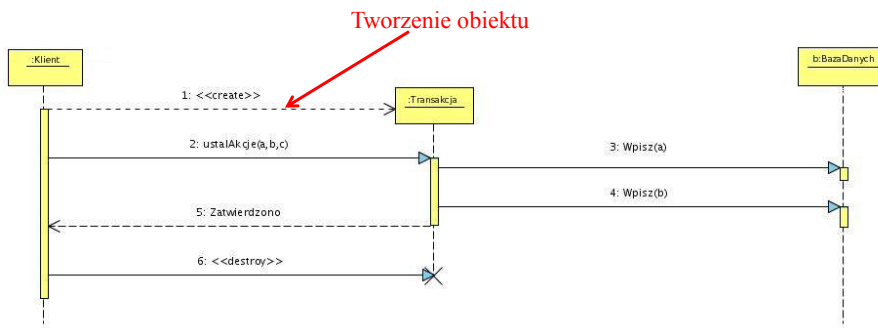


Sygnatury komunikatów

**Atrybut = nazwa_sygnału_lub_komunikatu
(argumenty) : typ_zwracany**

np. **myVar = doSomething(num:Number):ReturnClass** –
Komunikat nosi nazwę doSomething, przyjmuje jeden argument klasy Number oraz zwraca obiekt klasy ReturnClass, który jest następnie przypisywany do atrybutu myVar nadawcy.

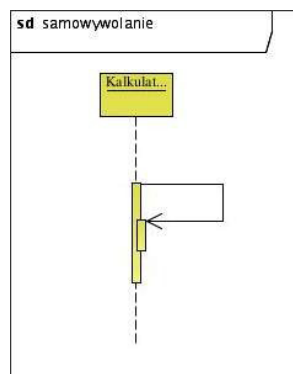
Tworzenie, usuwanie obiektów



Projekt musi specyfikować, kto jest odpowiedzialny za usuwanie obiektów, aby zapobiec tzw. “wyciekaniu pamięci”.

Niektóre języki, takie jak np. Java czy SmallTalk, posiadają wbudowane mechanizmy zbierania nieużytków (garbage collectors). Z grubsza, polega to na usuwaniu (w jakimś czasie) wszystkich obiektów, do których nie ma żadnych referencji w systemie.

Samowywołanie



Fragmenty wyodrębnione

Fragment wyodrębniony jest to logicznie spójny obszar interakcji, część diagramu sekwencji charakteryzująca się specyficznymi właściwościami określonymi przez operator interakcji.

Stosowanie fragmentów wyodrębnionych umożliwia bardziej precyzyjne zobrazowanie istoty interakcji, co jest szczególnie ważne w przypadku systemów czasu rzeczywistego oraz wspomagających skomplikowane procesy biznesowe. Specyfika fragmentu wyodrębnionego jest ściśle uzależniona od charakteryzującego go **operatora interakcji** (ang. *interaction operator*).

Operator interakcji stanowi sprecyzowanie funkcjonalności realizowanej przez fragment wyodrębniony.

Fragmenty wyodrębnione

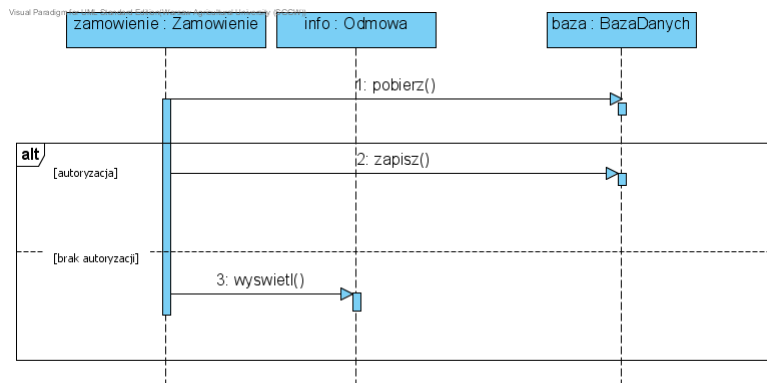
Fragment wyodrębniony występuje w formie obramowanej z wyszczególnieniem na główka. W nagłówku obligatoryjnie umieszczony jest operator charakteryzujący specyfikę danego fragmentu wyodrębnionego wraz z ewentualnymi parametrami, zapisany w następującej konwencji:

<operator-interakcji> [<parametry>].

Mamy następujące operatory interakcji:

- alt* — alternatywa,
- opt* — opcja,
- break* — przerwanie,
- loop* — iteracja,
- neg* — funkcjonalność nieprawidłowa,
- par* — współbieżność,
- critical* — obszar krytyczny,
- assert* — formuła,
- consider* — istotność,
- ignore* — nieistotność,
- stricte* — ściśle uporządkowanie,
- seq* — słabe uporządkowanie.

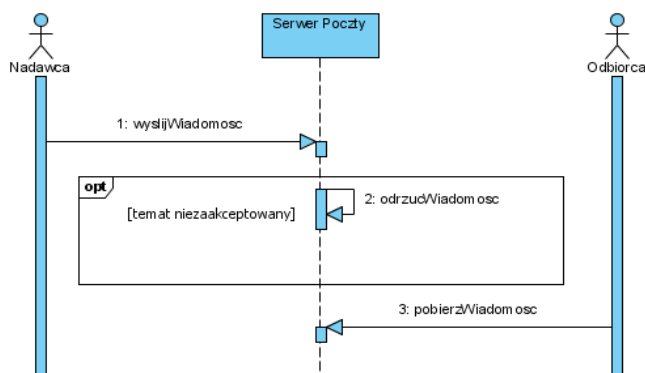
Alternatywa - alt



Oznacza możliwość wyboru jednego i tylko jednego spośród wszystkich operandów danego fragmentu wyodrębnionego. **Wybór jest dokonywany na podstawie warunku przypisanego do operandu.**

Przypomina instrukcję if(...) else w kodzie programu.

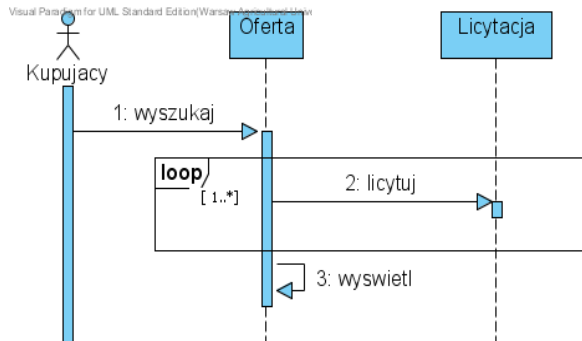
Opcja - opt



Interakcje zawarte w tego rodzaju fragmencie zostaną wykonane jedynie w przypadku, gdy warunek będzie mieć wartość logiczną prawdą.

Przypomina prostą instrukcję if(...).

Iteracja - loop

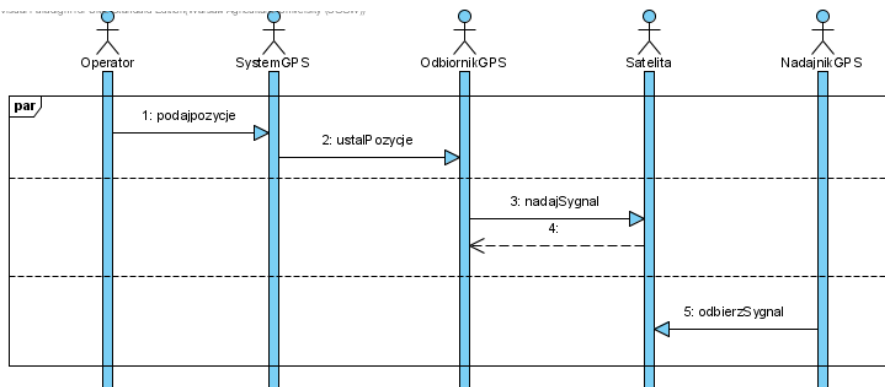


Oznacza powtarzanie operandu określoną ilość razy. Dolna i górna liczba iteracji może być wskazana w formie parametru, jak poniżej:

`loop ["(" <minint> [" , " <maxint>] ")"]`

Parametr *minint* jest nieujemną liczbą całkowitą, natomiast *maxint* — liczbą całkowitą większą od *minint*. W przypadku wyspecyfikowania tylko jednego parametru pętla zostanie wykonana wskazaną ilość razy. Pominięcie parametrów jest domyślnie traktowane jako wykonanie pętli od zera do nieskończoności. Jest również możliwość podania warunku – w tej sytuacji interakcje będą powtarzane dopóty warunek będzie mieć wartość prawda.

Współbieżność - par



Wskazuje na równoległe wykonywanie wszystkich operandów danego fragmentu wyodrębnionego.

Tworzenie diagramów sekwencji

- Każdy diagram sekwencji obrazuje jedną interakcję.
 - Interakcja może być związana z jednym przypadkiem użycia.
 - Jeśli więc dla projektu „sklep internetowy” mamy trzy przypadki użycia: *zakup*, *zwrot*, *dostawa towaru*, to należy sporządzić trzy diagramy sekwencji.
 - Diagramy te będą pokazywać interakcje zachodzące pomiędzy obiektami w systemie, gdy będzie dokonywany:
 - Zakup,
 - Zwrot,
 - Dostawa towaru.