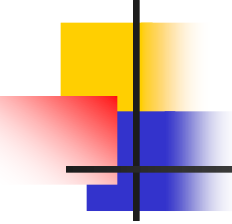


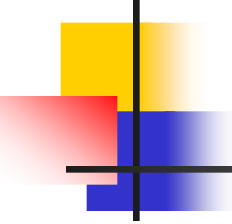
Zalety projektowania obiektowego

- Łatwe zarządzanie
- Możliwość powtórnego użycia klas obiektów – projektowanie/programowanie komponentowe
- W wielu przypadkach występuje stosunkowo proste mapowanie pomiędzy elementami rzeczywistego środowiska systemu a obiektami
- Wsparcie ze strony narzędzi programistycznych oraz istnienie standardu notacji obiektowej (UML)



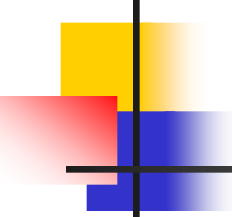
Obiekty - komunikacja

- Konceptyjnie obiekty komunikują się poprzez przesyłanie wiadomości/komunikatów gdzie nazwa wiadomości określa usługę do wykonania. Wiadomość zawiera również informacje niezbędne do wykonania operacji oraz posiada możliwość zwracania wartości będących wynikiem wykonania operacji
- W praktyce przekazywanie wiadomości realizowane jest jako wywołanie metod zdefiniowanych w klasie obiektu:
 - Nazwa metody == nazwa wiadomości
 - Informacja == wartości parametrów metody
 - Wynik operacji == wartość zwracana przez metodę



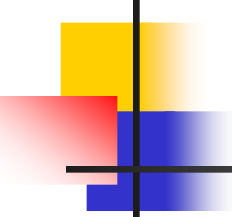
Proces tworzenia modelu obiektowego

- Identyfikacja kontekstu (środowiska) pracy systemu
- Określenie przypadków (modeli) wykorzystania systemu
- Budowa obiektowego modelu architektury systemu:
 - Identyfikacja klas i obiektów
 - Identyfikacja związków klas i obiektów
 - Identyfikacja atrybutów klas
 - Identyfikacja i definiowanie metod i komunikatów – interfejsy obiektów



Przykład - system gromadzenia informacji pogodowych - opis

- System gromadzenia informacji pogodowych (weather forecast system) służy do automatycznej generacji map pogodowych na podstawie danych gromadzonych przez zdalne stacje pogodowe (weather station) i inne źródła informacji pogodowych takie jak satelity i balony obserwacyjne. Każda stacja pogodowa przesyła zgromadzone dane do centralnego komputera w odpowiedzi na jego zapytanie. Komputer centralny weryfikuje otrzymane dane oraz integruje wszystkie dane przesłane z różnych źródeł. Zgromadzone w ten sposób dane przechowywane są w archiwum danych. Na podstawie zapisanych informacji i bazy map system generuje lokalne mapy pogodowe dla różnych obszarów. Stworzone mapy mogą być drukowane na ploterze lub wyświetlane w różnych formatach.

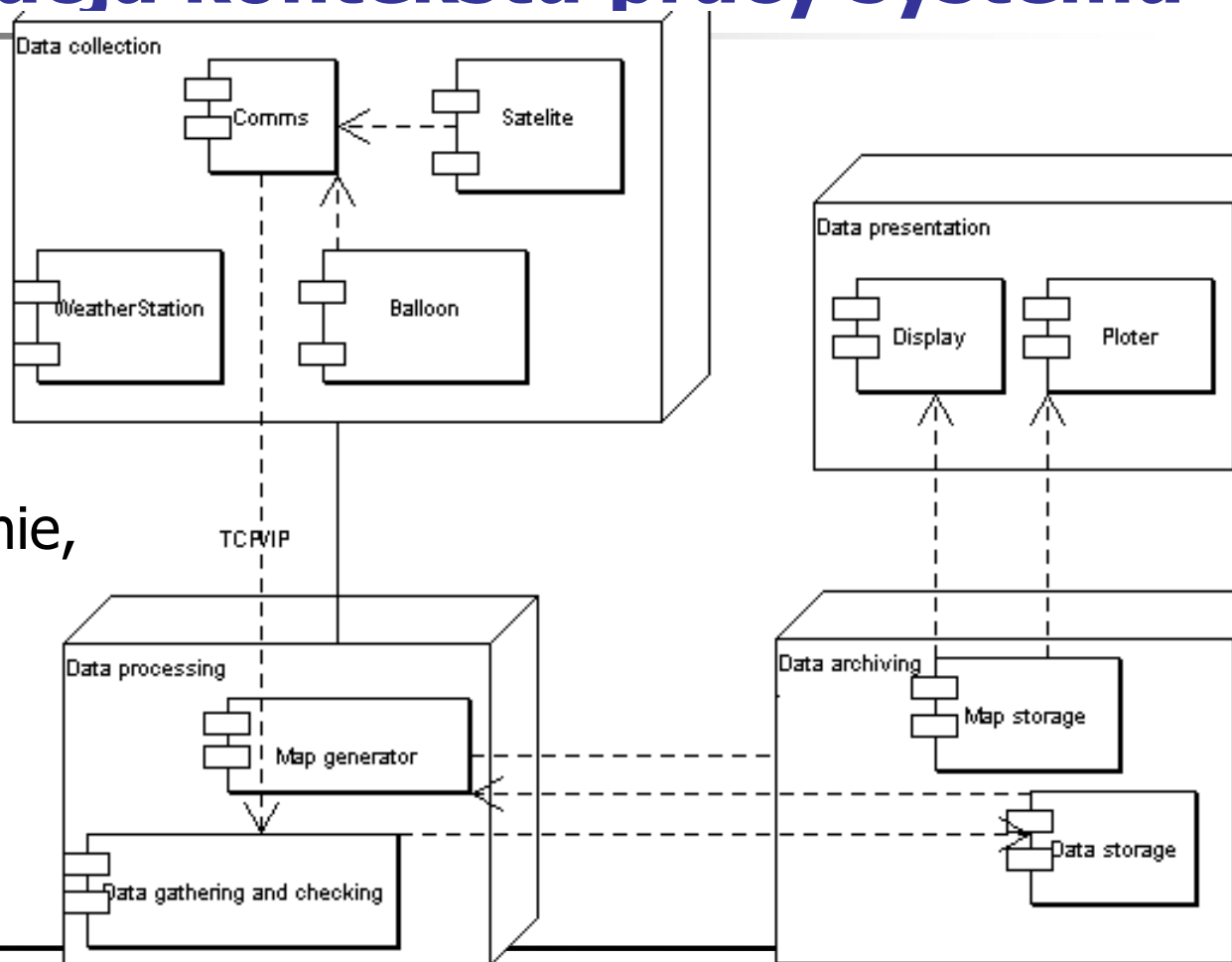


Opis stacji pogodowej (weather station)

- Stacja pogodowa (weather station) jest zbiorem przyrządów pomiarowych kontrolowanych za pomocą specjalizowanego oprogramowania. Przyrządy pomiarowe gromadzą dane pogodowe (np. temperaturę powietrza, wilgotność itp.), które następnie zostają poddane wstępnej analizie i w odpowiedzi na żądanie zostają przesłane do centralnego komputera. W skład przyrządów pomiarowych wchodzi termometry mierzące temperaturę powietrza i ziemi, prędkość wiatru, ciśnienie oraz ilość opadów. Dane z poszczególnych przyrządów odczytywane są w pięciominutowych odstępach. W momencie gdy zostanie odebrane żądanie przesyłu danych, oprogramowanie stacji pogodowej przetwarza zgromadzone dane i przesyła wyniki przetwarzania do komputera centralnego.

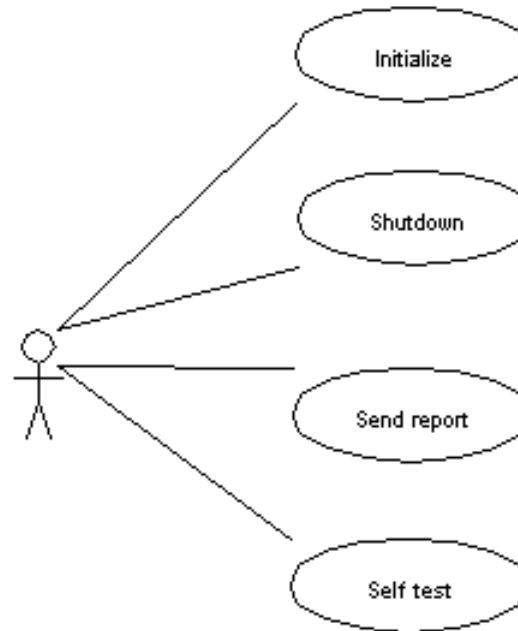
Identyfikacja kontekstu pracy systemu

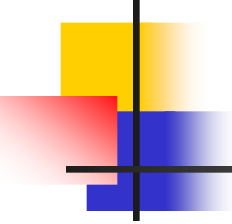
Kontekst pracy systemu określa poszczególne elementy wchodzące w skład systemu lub stanowiące jego otoczenie, z którym system wchodzi w interakcję



Określenie przypadków wykorzystania systemu – przykład stacji pogodowej

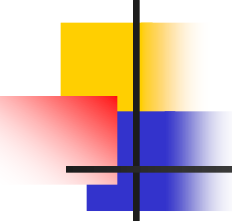
Dynamiczny model określający sposób interakcji systemu z poszczególnymi elementami otaczającego środowiska



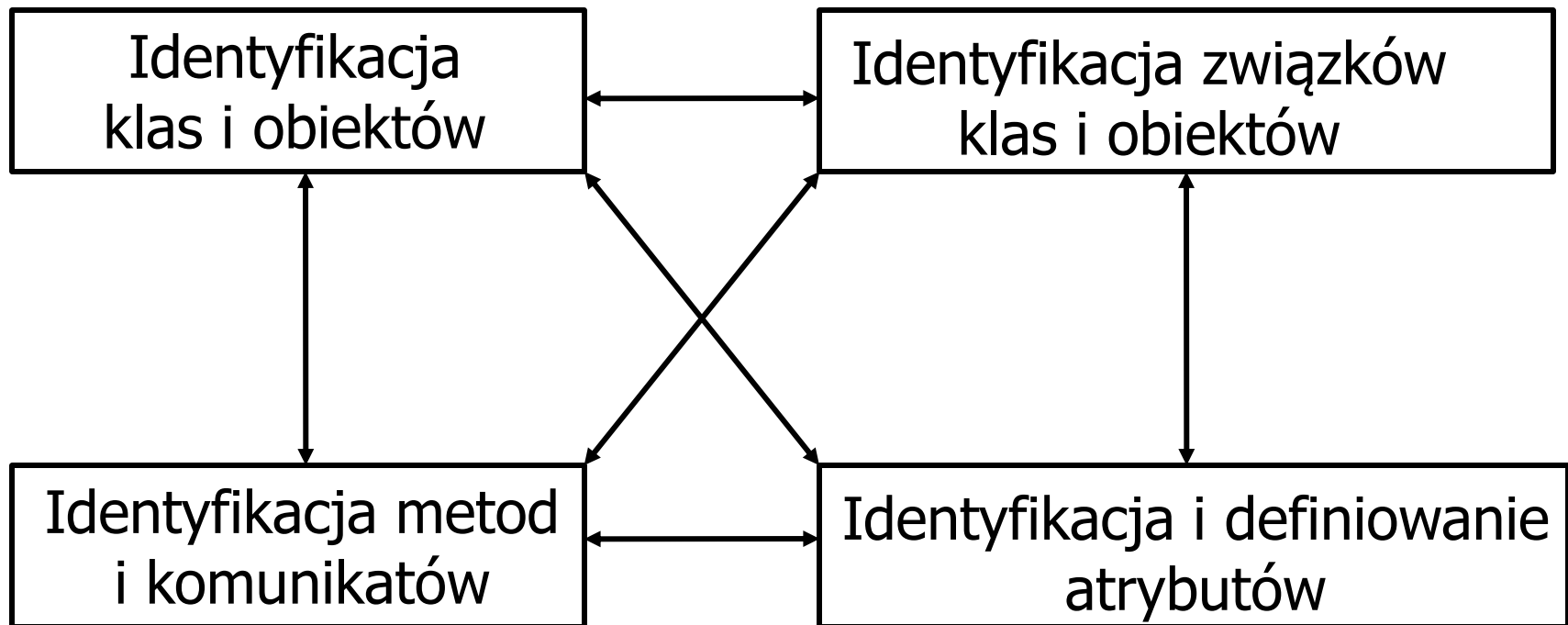


Opis przypadku użycia (ang. use-case)

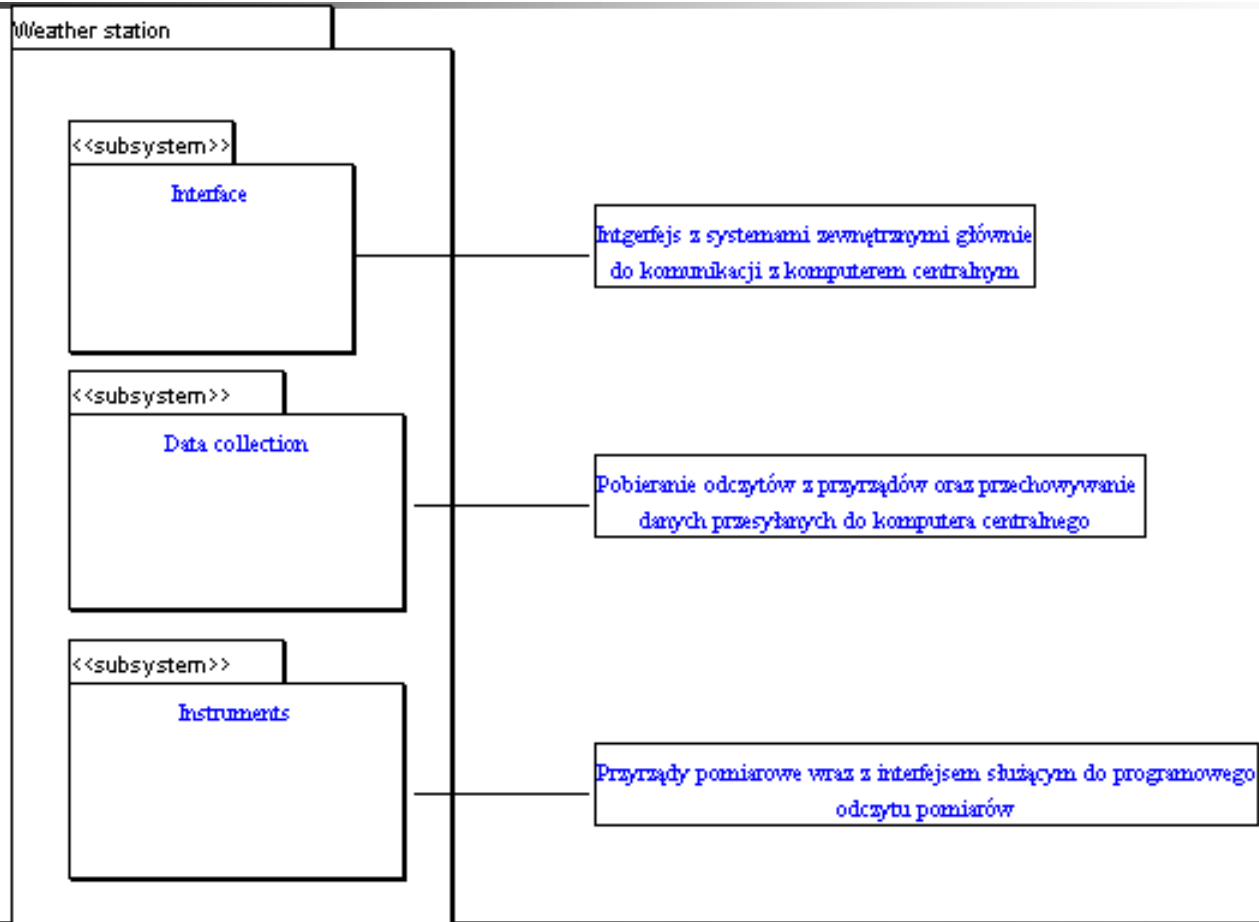
System:	stacja pogodowa (weather station)
Use-case:	send report
Aktorzy:	podsystem data collection
Przetwarzanie:	w odpowiedzi na żądanie stacja pogodowa przesyła dane zgromadzone przez poszczególne przyrządy pomiarowe w ostatnim okresie gromadzenia danych. Przesyłane dane dotyczą minimalnej, maksymalnej i średniej temperatury powietrza i ziemi, minimalnego, maksymalnego i średniego ciśnienia powietrza, minimalnej, maksymalnej i średniej prędkości wiatru, całkowitej ilości opadów
Zdarzenie:	podsystem data collection zestawia modemowe połączenie do stacji pogodowej i przesyła żądanie transmisji zgromadzonych danych
Odpowiedź:	zgromadzone w ostatnim okresie dane zostają przesłane do podsystemu data collection
Komentarz:	podsystem data collection wysyła żądania przesłania raportów do stacji pogodowych w odstępach nie krótszych niż 10 minut

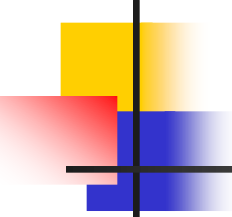


Budowa obiektowego modelu architektury systemu



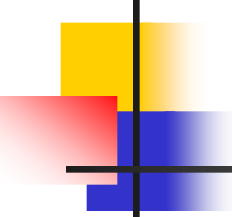
Logiczny model architektury systemu dla stacji pogodowej





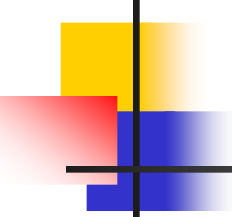
Identyfikacja klas i obiektów – metoda klasyczna (1/2)

- Poszukiwanie klas obiektów polegające na obserwacji, klas i obiektów w innych systemach. Na podstawie analizy listy typowych klas określa się klasy obiektów w analizowanym systemie. Do typowych klas można zaliczyć:
 - przedmiot namacalne (np.: samochód, czujnik)
 - role pełnione przez osoby (np.: pracownik, student, wykładowca)
 - zdarzenia, o których system przechowuje informacje (np.: zamówienie, dostawa)
 - interakcje pomiędzy osobami, systemami, o których system przechowuje informacje (np.: spotkanie, konferencja)



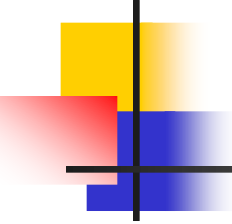
Identyfikacja klas i obiektów – metoda klasyczna (2/2)

- lokalizacje – miejsca przeznaczone dla ludzi lub przedmiotów (np.: magazyn, mieszkanie)
 - grupy przedmiotów namacalnych (np.: czujniki, studenci)
 - organizacje (np.: firma, wydział, uczelnia)
 - koncepcje (np.: miara jakości)
 - dokumenty (np.: faktura, prawo jazdy)
 - interfejsy dla systemów zewnętrznych lub urządzeń
-
- Należy zwrócić uwagę, że pewne elementy mogą posiadać dwojakie znaczenie w zależności od interpretacji np.: pracownik może oznaczać klasę opisującą jeden z elementów systemu (np.: w systemie finansowo-księgowym) jak również system zewnętrzny – użytkownika systemu
-



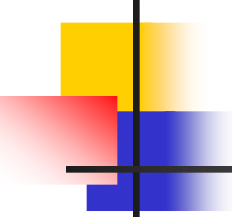
Identyfikacja klas i obiektów – metoda analizy opisu w języku naturalnym

- Na podstawie sporządzonego w fazie analizy wymagań opisu systemu określa się klasy obiektów oraz związane z nimi operacje. W opisie wyróżnia się rzeczowniki (wraz z opisującymi je przymiotnikami) oraz czasowniki. Rzeczowniki traktuje się jako potencjalne klasy, obiekty lub atrybuty. Czasowniki to potencjalne metody lub związki pomiędzy klasami.
- Ze względu na niejednoznaczność i wieloznaczność języka naturalnego może powstać wiele czasami niepotrzebnych klas obiektów lub łączących je relacji.



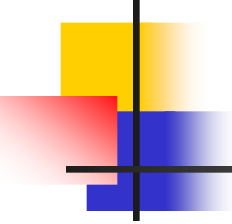
Identyfikacja klas i obiektów – metoda analizy funkcji (przypadków użycia)

- Analizie poddawane są kolejne use-case'y stworzone w fazie analizy wymagań. Na podstawie opisu skojarzonego z każdym use-case'm tworzony jest scenariusz interakcji obiektów jednocześnie wprowadzając niezbędne klasy i metody.



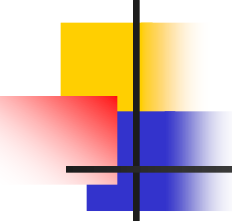
Weryfikacja klas i obiektów (1/2)

- Weryfikując konieczność wprowadzenia danej klasy należy wziąć pod uwagę następujące czynniki:
 - Brak atrybutów i metod – z reguły oznacza to, że klasa znajduje się poza zakresem odpowiedzialności systemu
 - Nieliczne pojedyncze atrybuty i metody – istnieje możliwość, że pola i metody mogą zostać umieszczone w innej klasie



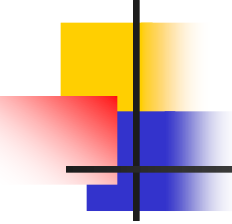
Weryfikacja klas i obiektów (1/2)

- Tylko jeden obiekt danej klasy – w pewnych przypadkach może to oznaczać zbyt rozbudowaną hierarchię dziedziczenia. Przykładowo dobrą klasą jest „samochód” ze specjalizacją np.: „samochód ciężarowy” natomiast błędną specjalizacją jest np.: „samochód studenta X”
- Brak związków z innymi klasami – z reguły oznacza to, że klasa znajduje się poza zakresem odpowiedzialności systemu



Identyfikacja atrybutów klas (1/3)

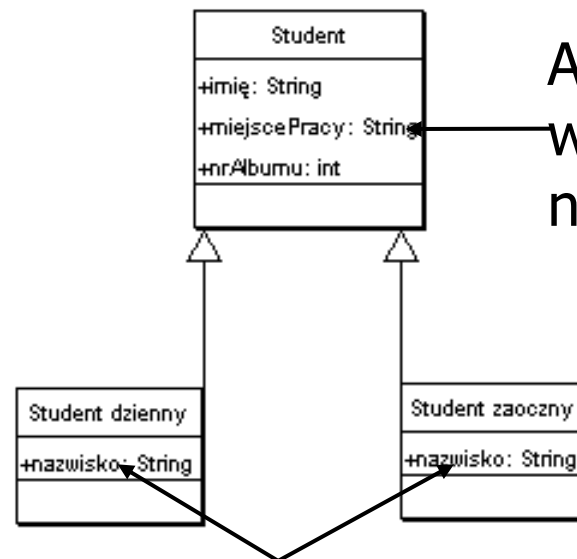
- Identyfikując pola klas należy spróbować odpowiedzieć na następujące pytania:
 - Jakie informacje potrzebne są do opisu klasy w ramach dziedziny problemu (np. klasa student wymaga atrybutów określających imię, nazwisko, nr albumu itp.)
 - Jakie dane będą potrzebne obiektom danej klasy do realizacji zadań (np.: klasa określająca grupę studentów musi posiadać kolekcję przechowującą np.: identyfikatory studentów należących do danej grupy)



Identyfikacja atrybutów klas (2/3)

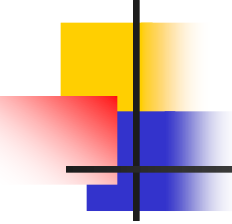
- Jakie pola należy wprowadzić, aby opisać stany w jakich mogą znajdować się obiekty danej klasy (np.: klasa opisująca wiadomości email może posiadać atrybut określający stan jako wysłany, odebrany, zwrócony)
- Na podstawie opisu w języku naturalnym można poprzez analizę występowania rzeczowników określić podstawowe atrybuty klas
- Interfejs GUI zaakceptowany przez użytkownika może dostarczyć bardziej szczegółowych informacji na temat wprowadzanych, edytowanych i prezentowanych danych, które powinny być przechowywane jako atrybuty odpowiednich klas

Identyfikacja atrybutów klas (3/3) – przykład błędów w umieszczaniu atrybutów w hierachii klas



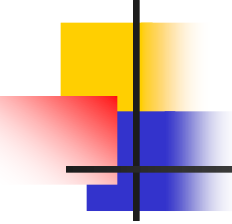
Atrybut umieszczony zbyt wysoko w hierachii gdyż nie każdy student pracuje

Atrybuty umieszczony zbyt nisko w hierachii przez co niepotrzebnie powtarzają się w definicji klas pochodnych pomimo tego, że określają tę samą właściwość



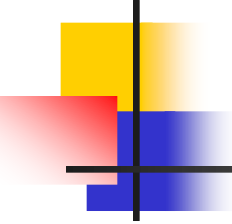
Identyfikacja metod i komunikatów klas

- Metody klas możemy zgrubnie podzielić na dwie kategorie:
 - Algorytmicznie proste
 - Algorytmicznie złożone



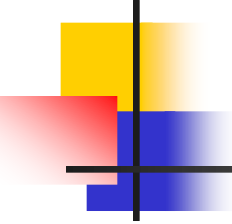
Identyfikacja metod i komunikatów klas

- Metody algorytmicznie proste:
 - Konstruktory/destruktory oraz metody inicjalizujące stan obiektów klasy
 - Metody służące do pobierania wartości publicznych atrybutów klasy
 - Metody służące do ustawiania wartości atrybutów klas
 - Metody służące do implementacji związków pomiędzy klasami (np.: agregacji)



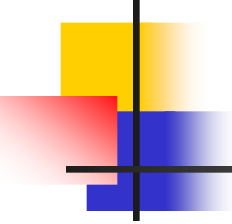
Identyfikacja metod i komunikatów klas

- Metody algorytmicznie złożone:
 - Metody służące do realizacji obliczeń
 - Metody służące do monitorowania pracy systemów i urządzeń zewnętrznych
 - (wszystkie inne nie będące metodami prostymi) 😊



Identyfikacja metod i komunikatów klas – analiza przypadków użycia (1/3)

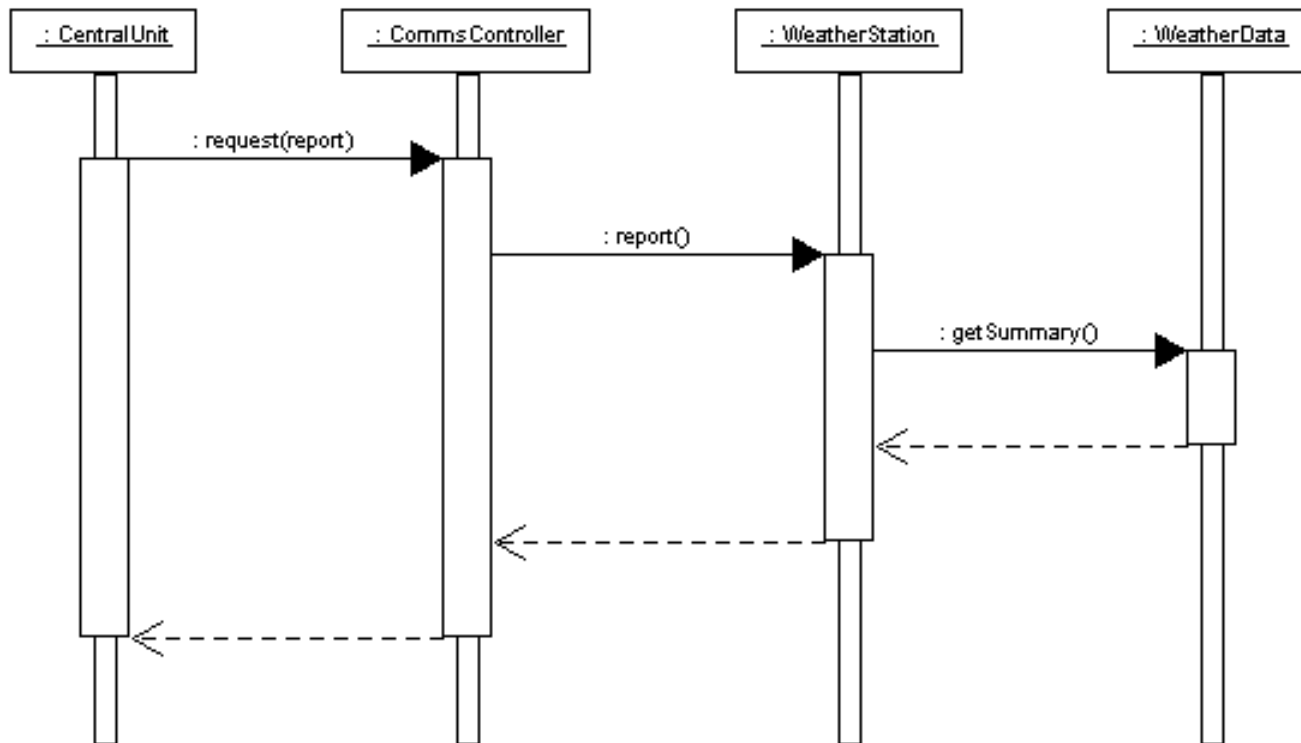
- Definiowanie metod klas poprzez analizę sposobu realizacji funkcji systemu wynikających z analizy poszczególnych use-case'ów
- Scenariusz przepływu komunikatów (wywołań metod) między obiektami systemu tworzony jest według następującego schematu:
 - Wybranie jednego z komunikatów otrzymywanych przez system (zwykle jeden z use-case'ów)

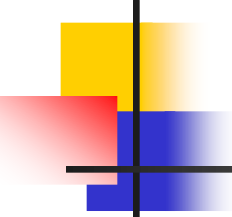


Identyfikacja metod i komunikatów klas – analiza przypadków użycia (2/3)

- Określenie klasy, która otrzyma komunikat (jeżeli klasa jeszcze nie istnieje należy ją stworzyć)
- Wybranie metody, która będzie obsługiwała komunikat lub stworzenie nowej
- Określenie czy do realizacji funkcji wystarczy jedna metoda. Jeżeli metoda nie jest elementarna należy określić jakie obiekty i odpowiednie ich metody będą brały udział w jej realizacji (krok ten może zostać wykonany w kolejnej iteracji)

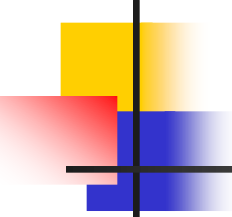
Identyfikacja metod i komunikatów klas – analiza przypadków użycia (3/3)





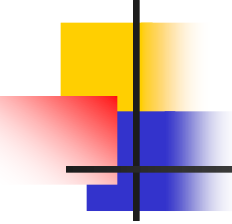
Diagramy interakcji (ang. interaction diagrams)

- Diagramy interakcji:
 - **Sequence** diagram
 - **Collaboration** diagram
- Diagramy interakcji należą do grupy pięciu diagramów wykorzystywanych w UML do modelowania dynamicznych aspektów systemów
- Każdy diagram interakcji prezentuje dynamiczne zależności pomiędzy obiektami wchodzącymi w skład systemu, które wzajemnie wymieniając komunikaty (ang. message) współdziałają w realizacji określonej funkcjonalności systemu.



Diagramy interakcji - komunikaty

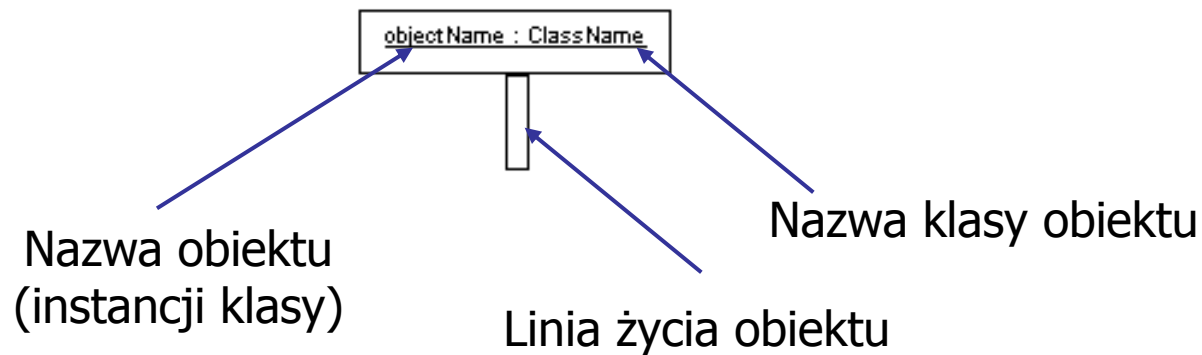
- Komunikat stanowi jedyną metodę wymiany informacji pomiędzy obiektami
- Komunikat wysłany do obiektu pewnej klasy oznacza żądanie wykonania jednej z metod tej klasy
- Komunikat może być wysłany przez system zewnętrzny lub przez obiekt jednej z klas systemu
- Wysłanie komunikatu może wiązać się z przekazaniem pewnych danych wejściowych do wywoływanej metody oraz z pobraniem danych wyjściowych zwracanych przez metodę
- Nazwa komunikatu jest nazwą wywoływanej metody



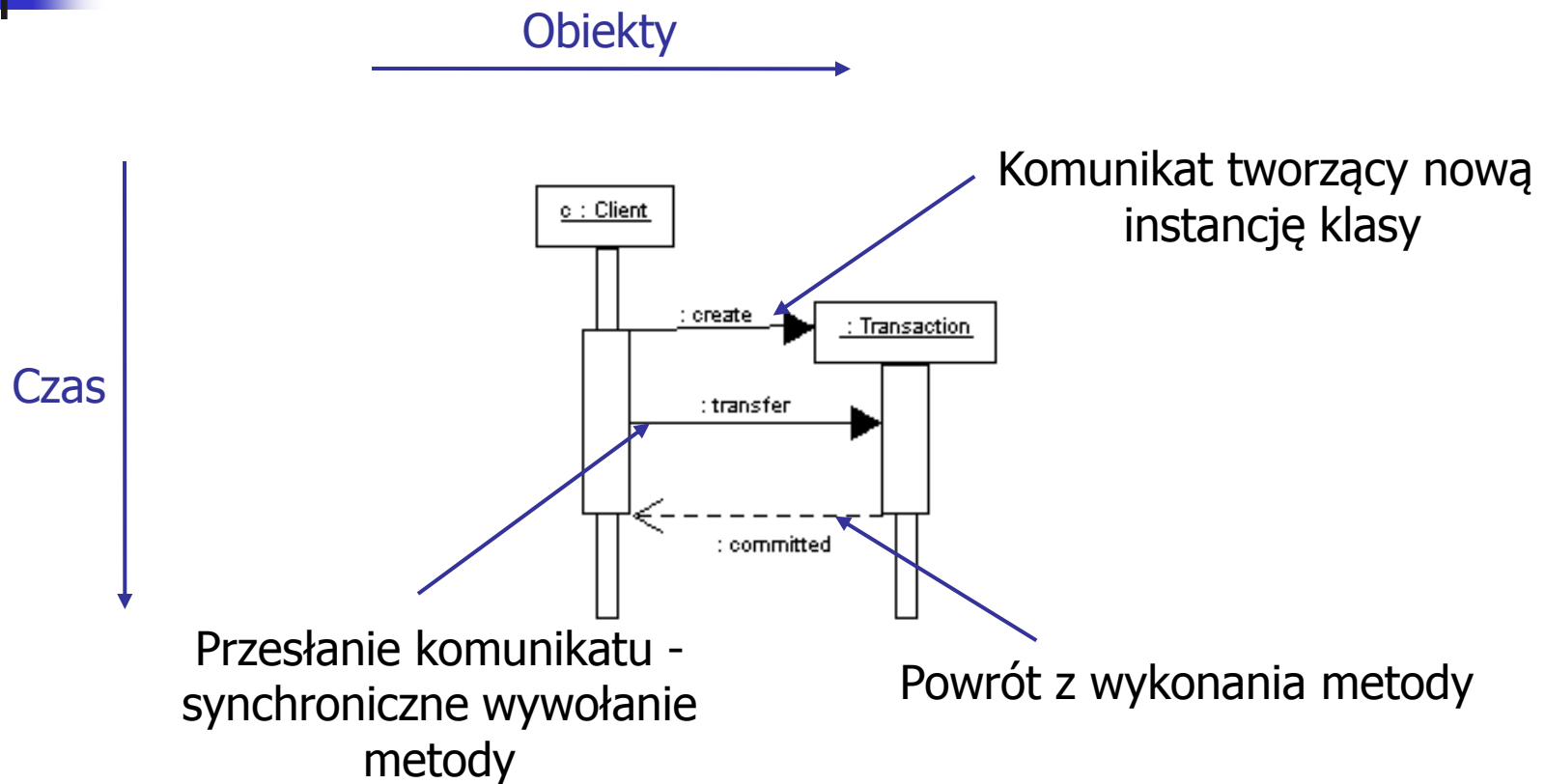
Diagramy typu sequence (1/5)

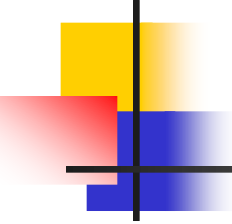
- Diagramy typu sequence prezentują przepływ **komunikatów** pomiędzy **współdziałającymi obiektami** zwracając uwagę na kolejność komunikatów w **czasie**
- Zazwyczaj stosowane są do opisywania szczegółów zachowania obiektów systemu i systemów zewnętrznych w ramach pojedynczych use case'ów
- Diagramy sequence zawierają następujące elementy:
 - Obiekty
 - Komunikaty

Diagramy typu sequence (2/5)



Diagramy typu sequence (3/5)





Diagramy typu sequence (4/5)

- Rodzaje komuniaków:
 - Synchroniczne
 - Asynchroniczne
 - Tworzące nowe instancje klas
 - Usuwające istniejące instancje klas
 - Powroty z wywołania metod

Diagramy typu sequence (5/5)

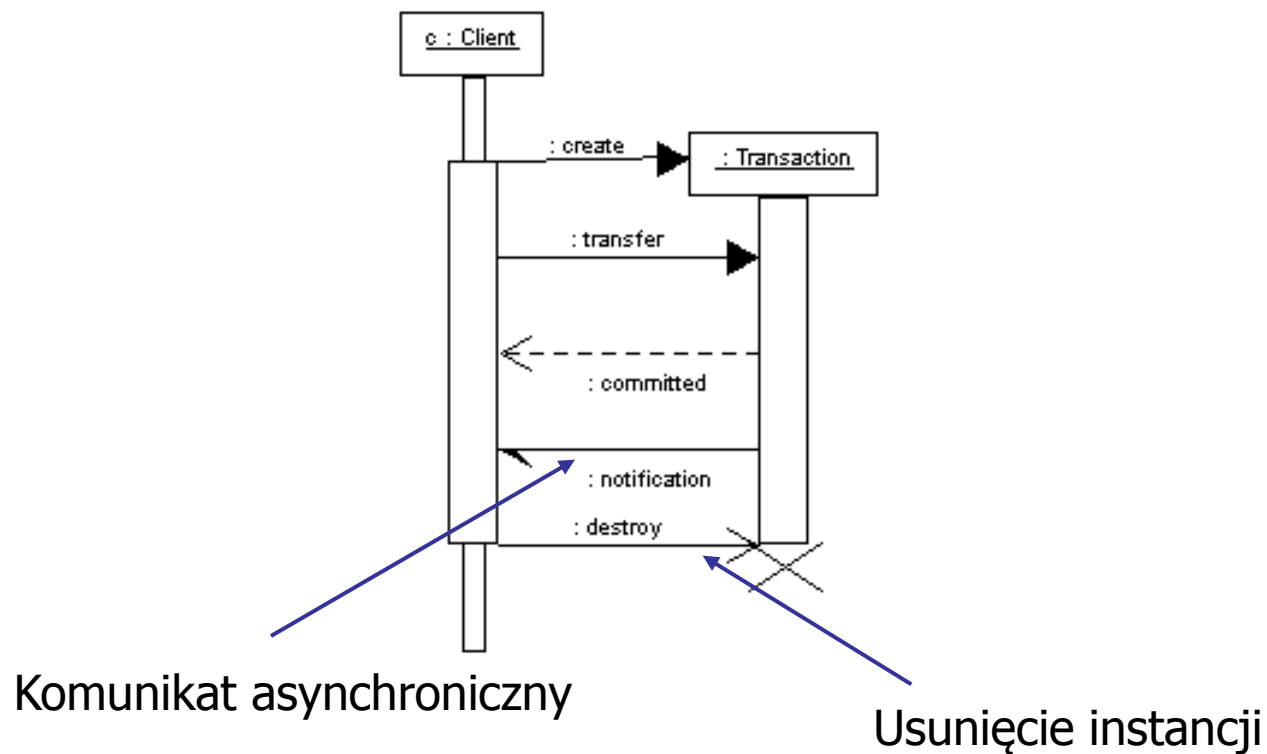


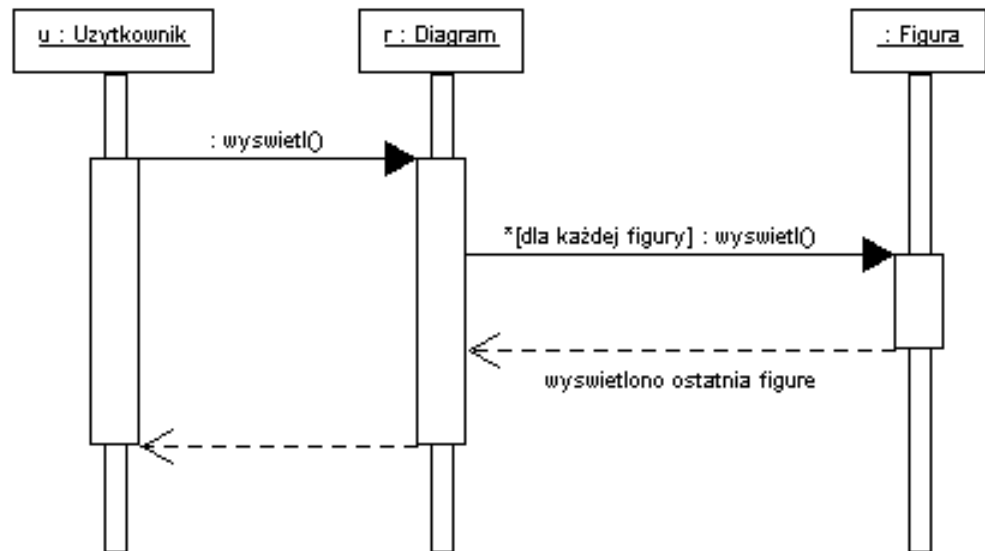
Diagram sequence opisany skryptem

Użytkownik żąda wyświetlenia diagramu

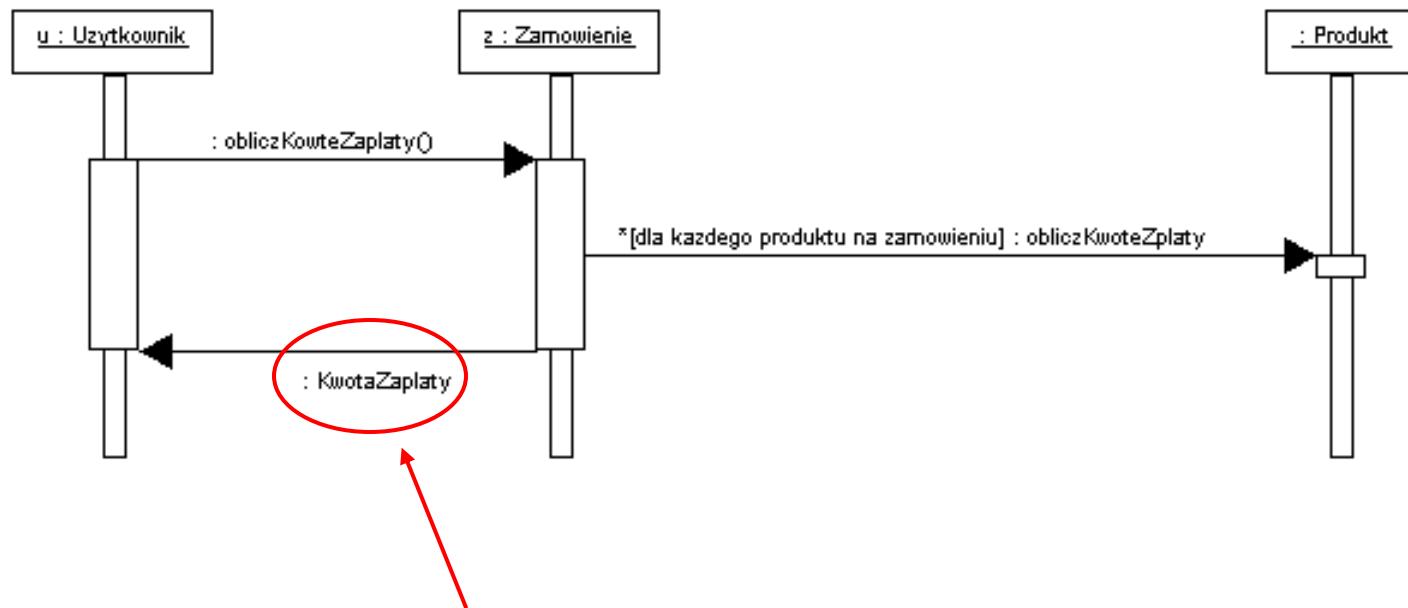
Wyświetlane są wszystkie figury,
z których składa się diagram

Użytkownik może wydać kolejne
polecenie po zakończeniu realizacji
wyświetlania diagramu

Skrypt

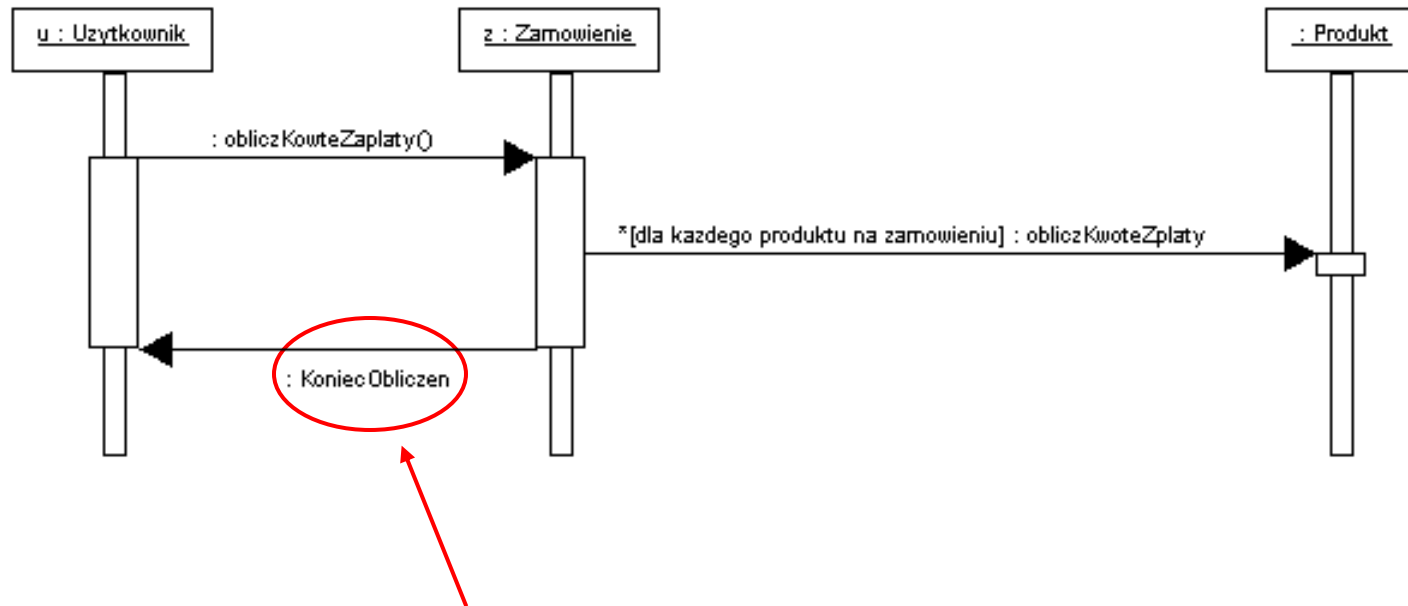


Podstawowe błędy popełniane w diagramach typu sequence



Symbol komunikatu błędnie użyty do zobrazowania przepływu danych

Podstawowe błędy popełniane w diagramach typu sequence



Symbol komunikatu błędnie użyty do zobrazowania zdarzenia