

Model-Driven Engineering

Radostaw Klimek
2015-22

<http://home.agh.edu.pl/rklimek>

Diagram aktywności

Aktywności (czynności)

Diagram aktywności przedstawia przepływ sterowania od czynności do czynności.

Czynność jest wieloetapowym działaniem wykonanym na maszynie stanowej. Wynikiem jest **akcja**, składająca się z niepodzielnych obliczeń prowadzących do zmiany stanu, lub przekazania wartości.

Wystąpienie (instancja) zdarzenia charakteryzuje się aktualnymi wartościami poszczególnych parametrów. Wartości te są dostępne dla akcji przypisanej do przejścia wyzwalanego przez to zdarzenie.

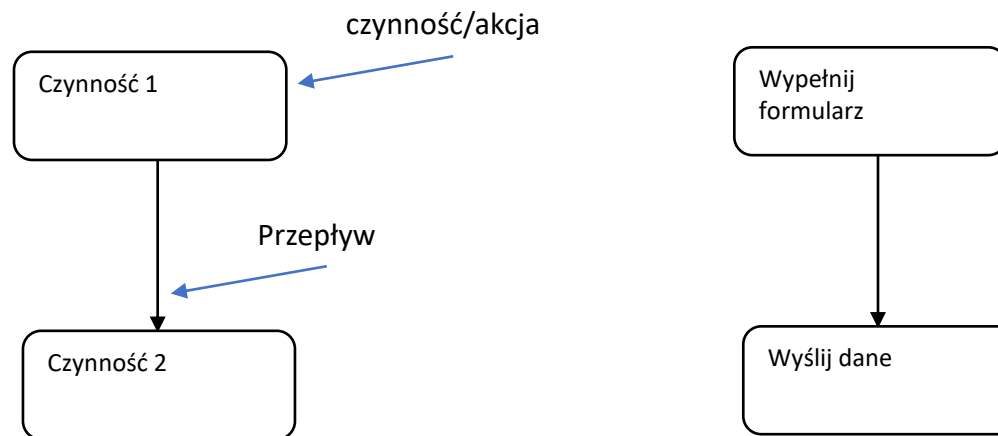
Przykłady akcji: wywołanie operacji, wysłanie sygnału, utworzenie lub zniszczenie obiektu, wyznaczenie wartości wyrażenia arytmetycznego. Innymi słowy diagram czynności jest schematem blokowym przedstawiającym czynności wykonywane w miarę upływu czasu (pewna analogia do sieci PERT).

Zawartość diagramów

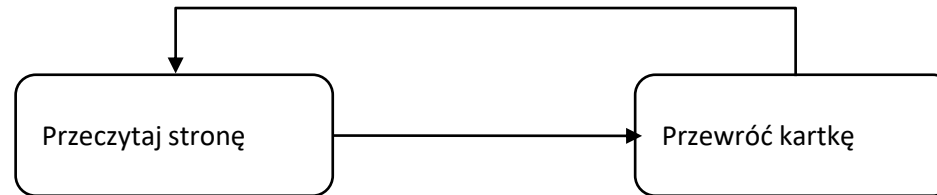
1. Stany akcji i czynności
2. Przejścia
3. Obiekty

Akcje są niepodzielne, czynności wieloetapowe i mogą być realizowane przez podmaszyny (inny podgraf czynności).

Przepływ sterowania w diagramie czynności składa się z wielu zdarzeń. Zdarzeniem może być obliczenie wartości wyrażenia (wartość przekazana do otoczenia lub wartość atrybutu), wywołanie operacji obiektu, wysłanie sygnału do obiektu, utworzenie lub zniszczenie obiektu.

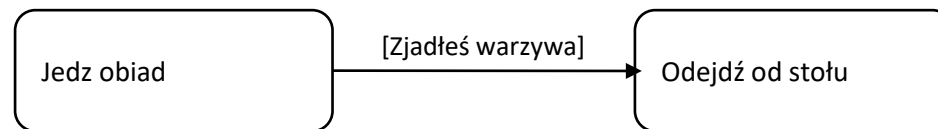


Przykłady przepływów



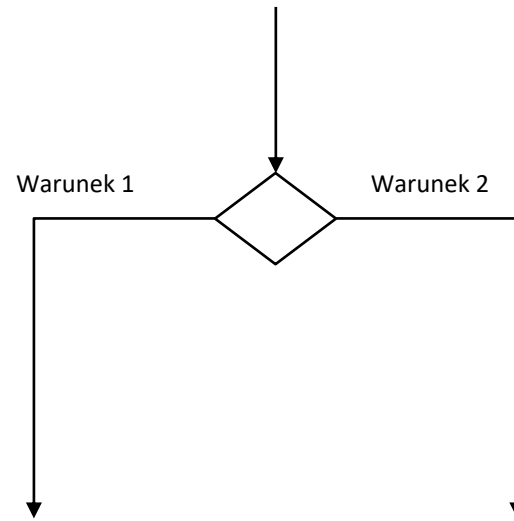
Akcje (lub czynności) łączone są przepływem sterowania (lub przejściem, ang. *transition*). Diagram aktywności jest serią akcji (lub czynności) połączonych przepływami.

Do przejścia dochodzi tylko i wyłącznie po ukończeniu poprzedzającej akcji. Przykładowo, przewrócenie kartki nastąpi po przeczytaniu strony.



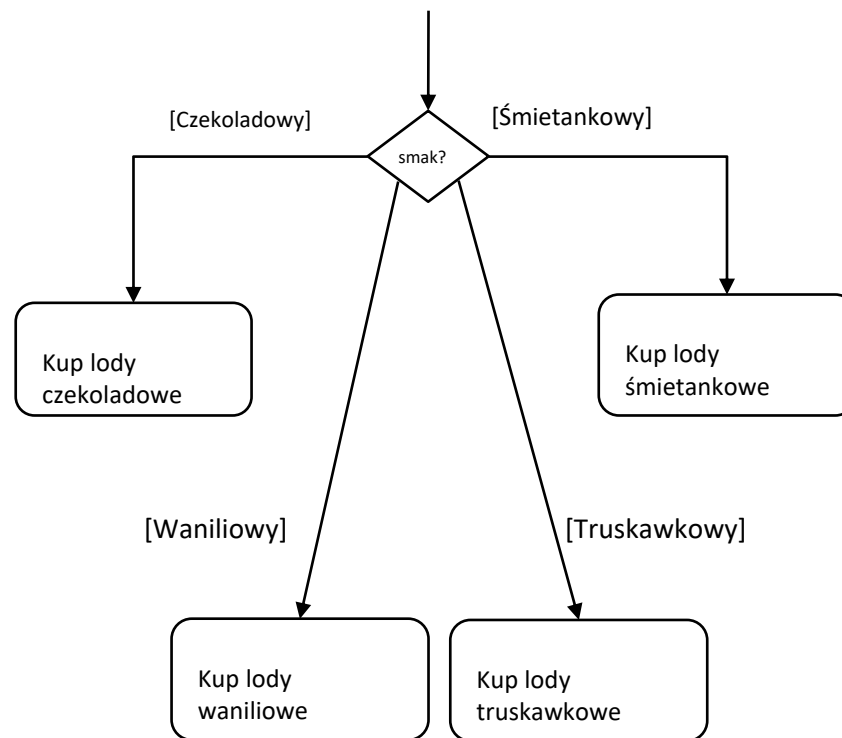
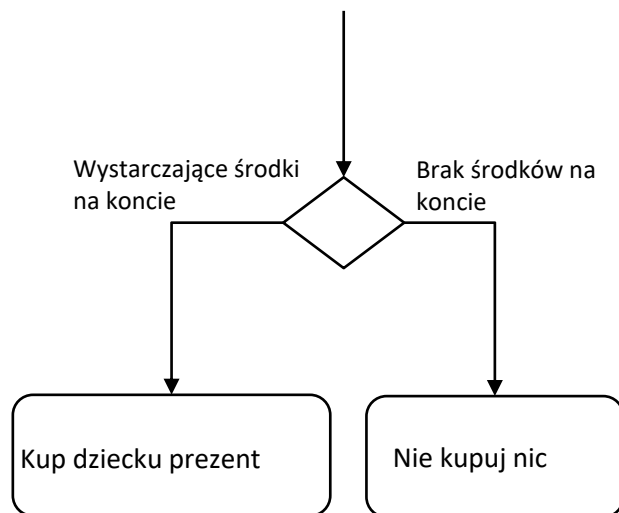
Czasami dalszy przepływ sterowania powinien nastąpić tylko i wyłącznie po nastąpieniu pewnego warunku.

W celu zamodelowania takiego zachowania nad strzałką należy umieścić warunek w kwadratowych nawiasach. Musi być to wyrażenie o wartości logicznej prawda lub fałsz.



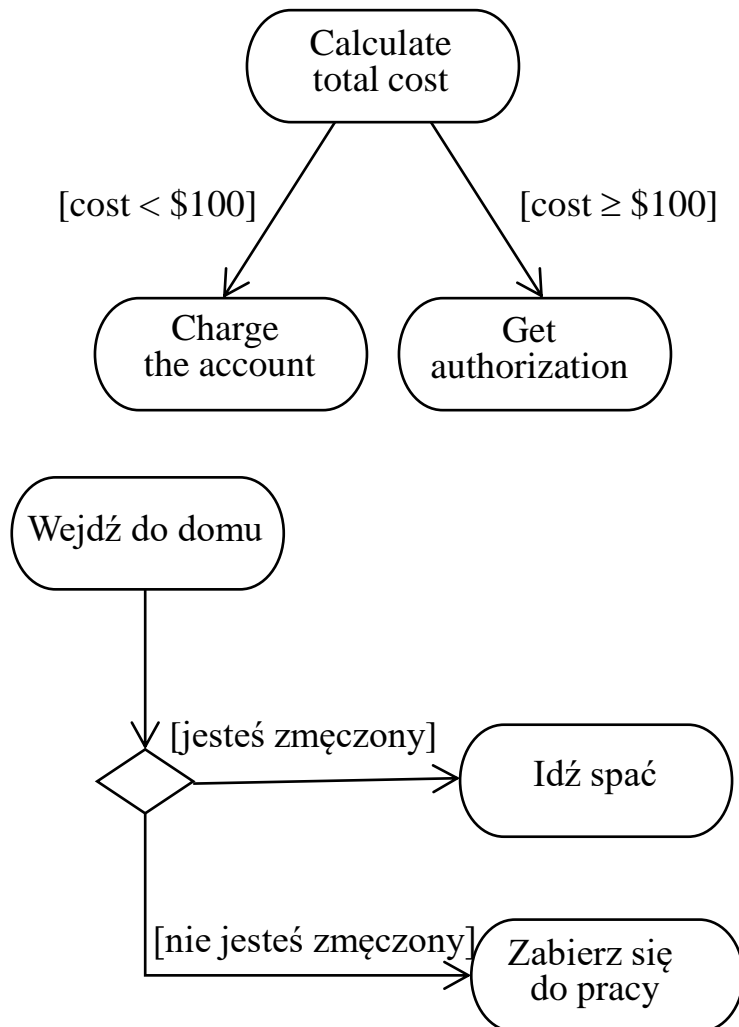
Podjęte decyzje są reprezentowane poprzez ikonę rozgałęzienia. Każdy z wychodzących przepływów musi posiadać nad sobą warunek. Wszystkie warunki razem muszą zawierać cały możliwy zbiór rozwiązań oraz wykluczać się wzajemnie.

Złożone decyzje



Z bloku decyzyjnego może wychodzić większa ilość krawędzi oraz warunki mogą być złożone, jednak ogół możliwości musi być wszystkimi możliwymi rozwiązaniami.

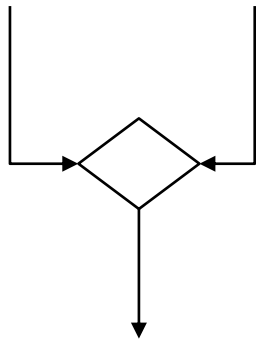
Decyzje a warunki



Rozgałęzienie przepływów sterowania może zostać zamodelowane również bez użycia bloków decyzyjnych, używając jedynie warunków.

Różnica polega na tym, że rozgałęzienie symbolizuje proces decyzyjny (np. oczekiwanie na decyzję z użytkownika).

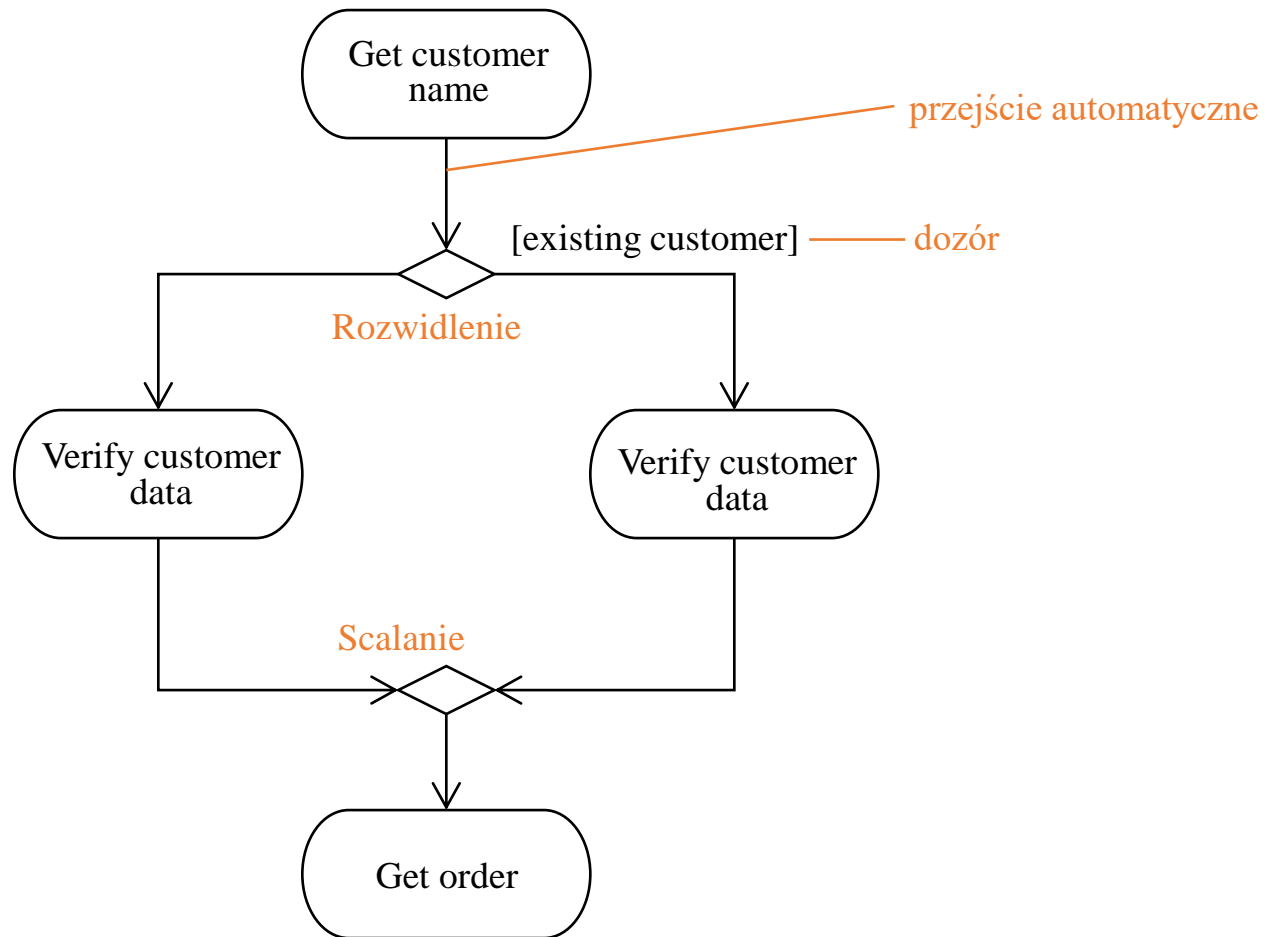
W przypadku, kiedy informacje potrzebne do podjęcia decyzji są znane wcześniej (jak np. w przykładzie – stan konta jest jawny), nie należy używać bloków decyzyjnych

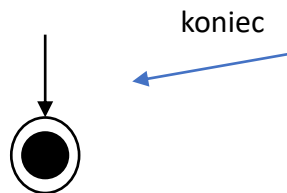
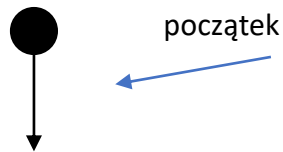


Dokładnie ten sam symbol co symbolizuje rozgałęzienie symbolizuje również scalenie.

Służy on do łączenia dwóch różnych przepływów sterowania w jeden.

Jego głównym zastosowaniem jest poprawa czytelności diagramów, gdyż w przeciwnym przypadku należałoby powielać te same czynności dla każdego przepływu.





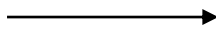
UML dostarcza również symbol początku i zakończenia diagram.

Na jednym diagramie, ze względu na obecny proces decyzyjny, może być więcej niż jedno zakończenie.

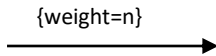
Symbol zakończenia oznacza natychmiastowe zakończenie wszystkich trwających czynności na diagramie.



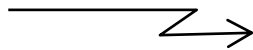
Stan początkowy



Przepływ sterowania – krawędź diagram aktywności



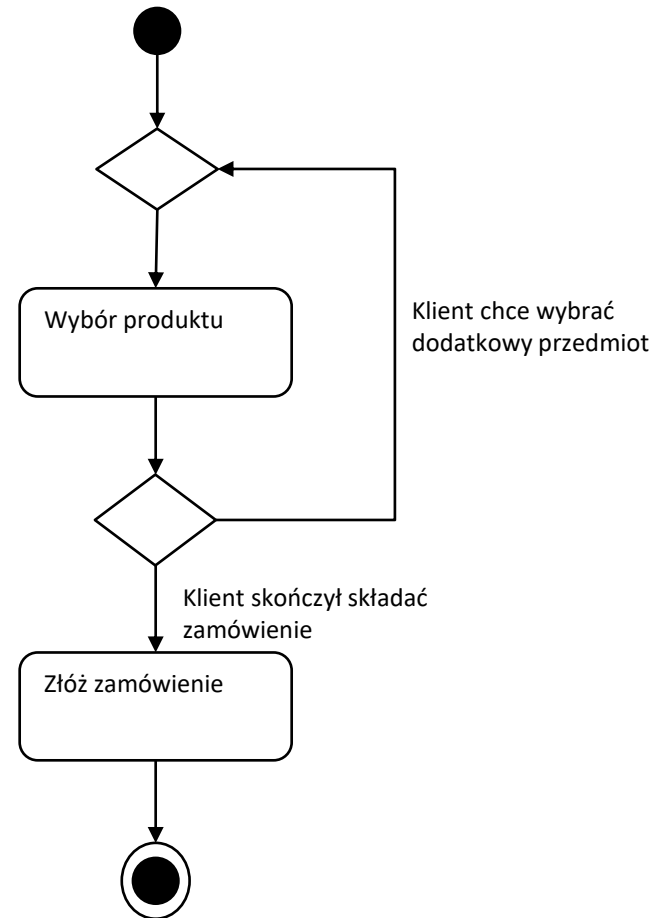
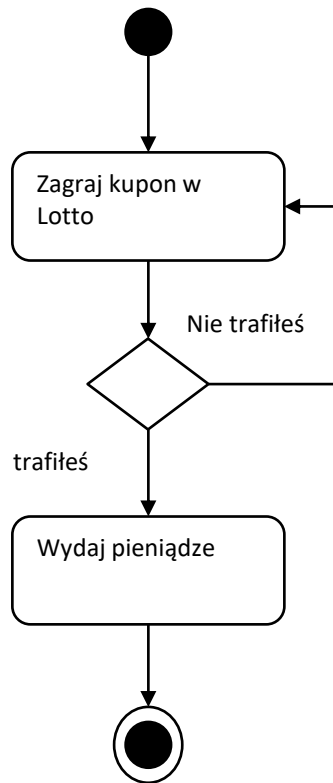
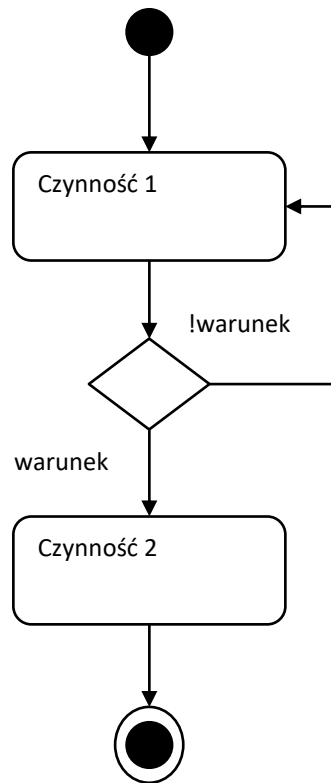
Przepływ sterowania z wagą



Krawędź w obszarach, w których może zaistnieć przerwanie

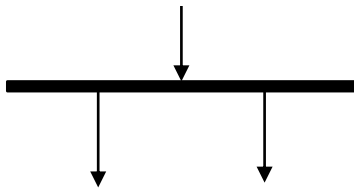


Stan końcowy

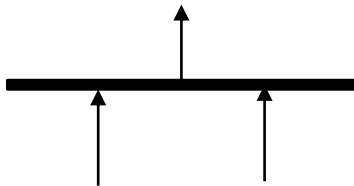


Współbieżność

Rozwidlenie - fork



Scalenie - join

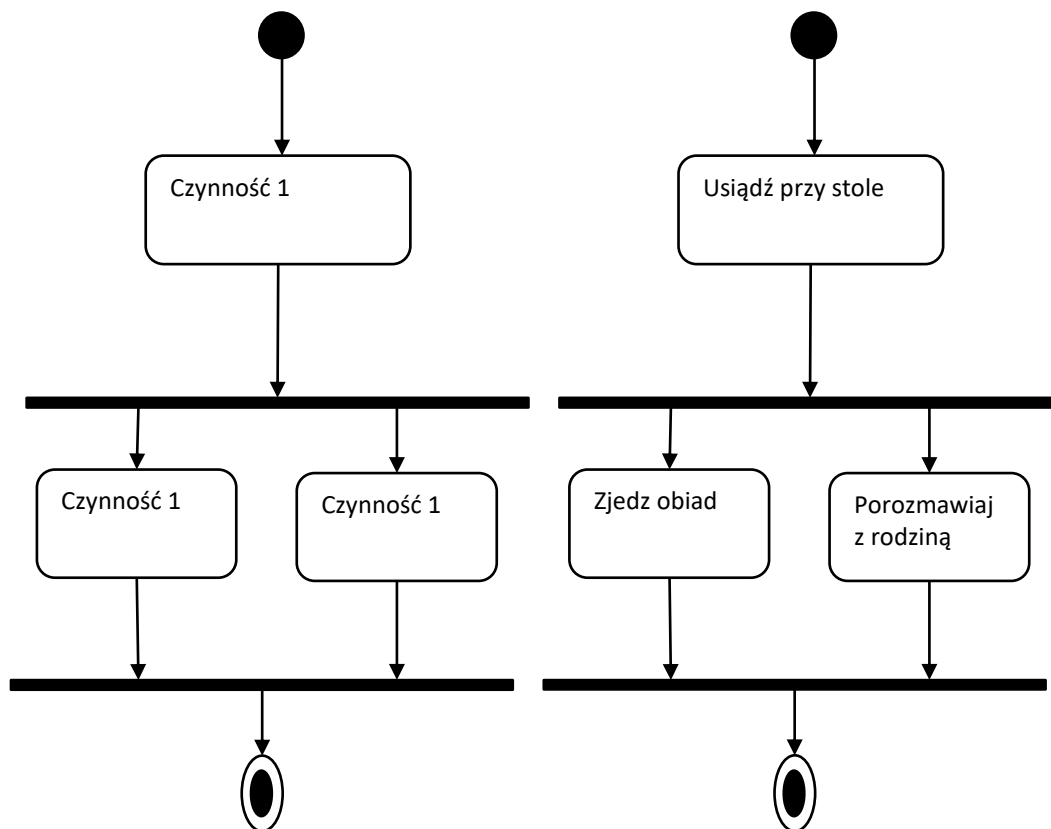


UML dostarcza również modelowania procesów wykonywanych współbieżnie przy pomocy prostego paska nazywanego rozwidleniem lub podziałem sterowania.

Oznacza on, że jeden proces tworzy wiele wątków (lub procesów) wykonujących się równolegle.

Synchronizację zapewnia scalenie – zatrzymuje pozostałe przepływy do czasu dotarcia pozostałych.

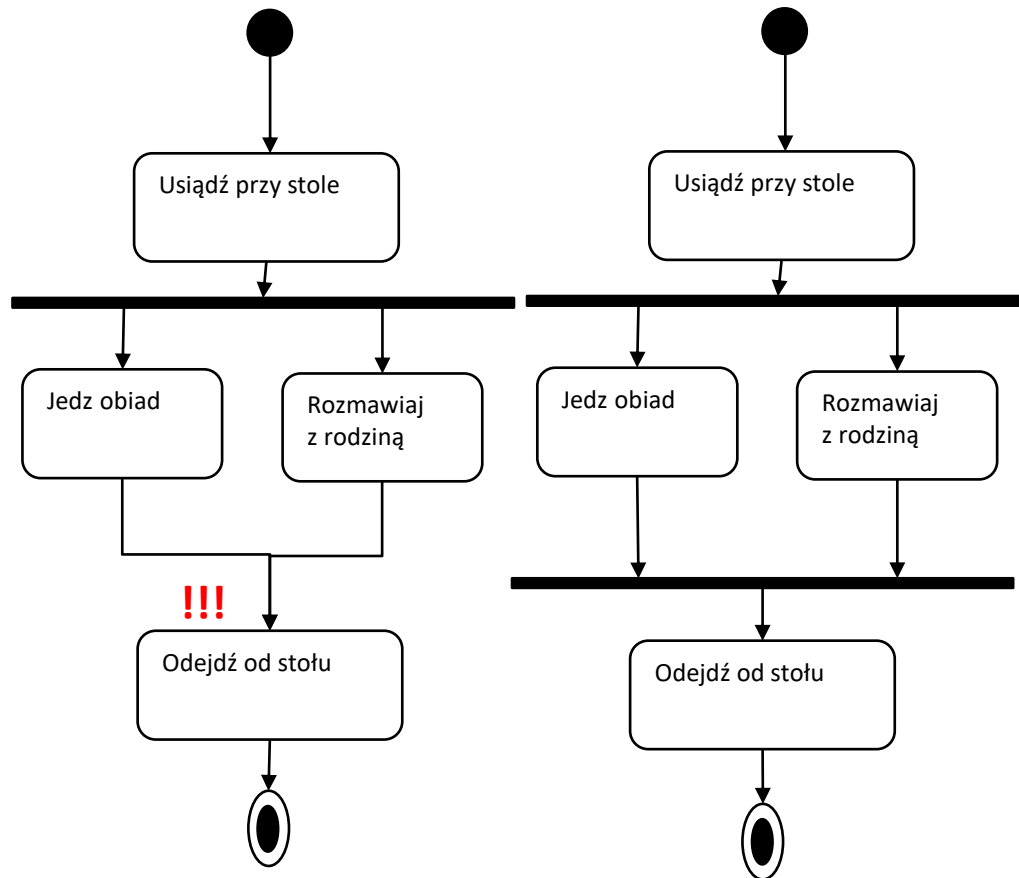
Współbieżność



Możliwość zrównoleglenia procesu, nie oznacza zawsze, że jest ono konieczne. Większość równoległych procesów może być również zaimplementowana sekwencyjnie.

W dużych aplikacjach wąskim gardłem przepływu wąskim gardłem przepływu często okazują się zewnętrzne czynniki (np. szybkość zapisu do bazy danych). W takich sytuacjach korzyść ze zrównoleglenia dwóch operacji działających w pamięci operacyjnej jest stosunkowo niewielka, a może powodować duże skomplikowanie kodu programu.

Przykłady przepływów – współbieżność

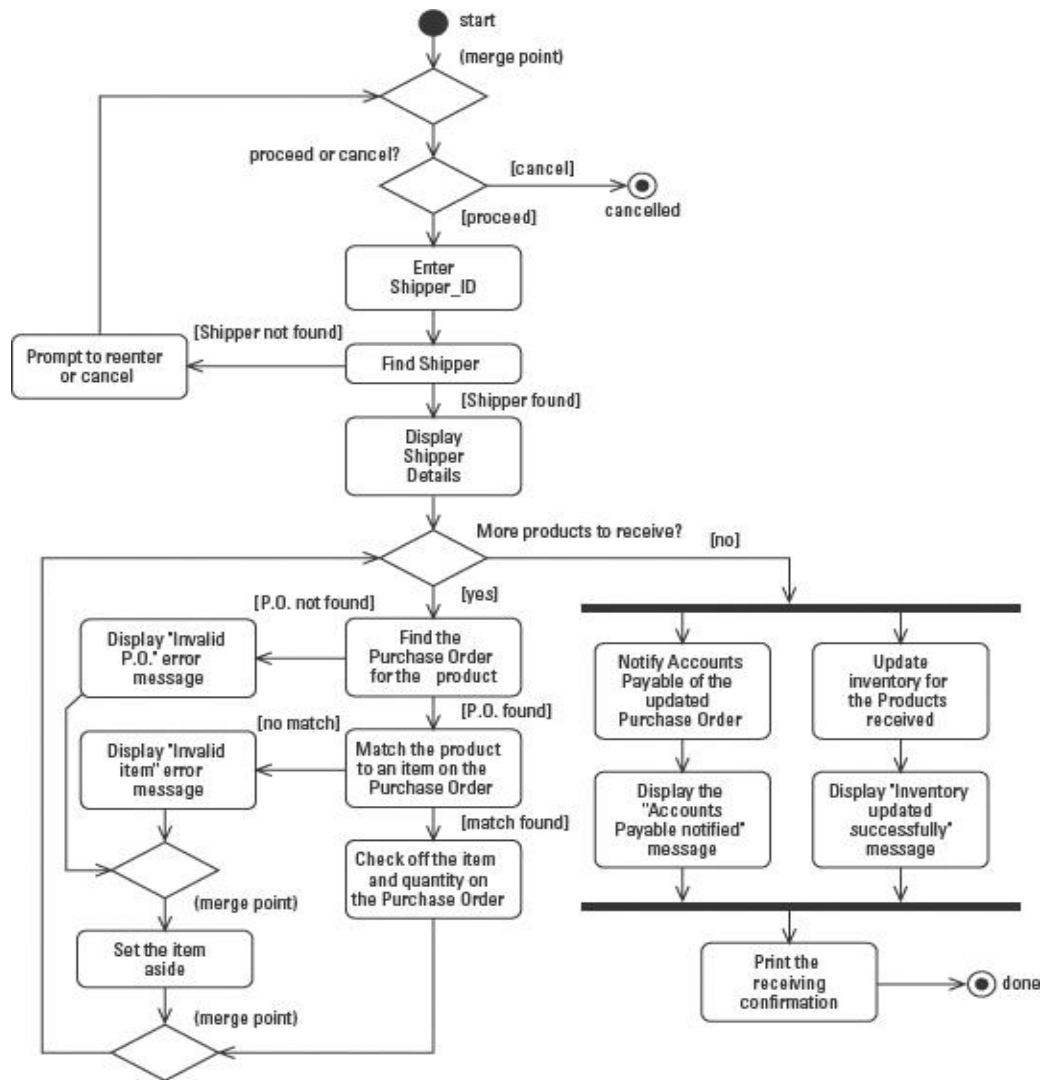


W przypadku niezastosowania scalenia (jak na rysunku pierwszym), nie ma gwarancji wykonania czynności obu wątków.

W dodatku, nie wiadomo, który z przepływów został wykonany.

Jedyną wiadomością to fakt, wykonania jednej z czynności.

Przykłady przepływów



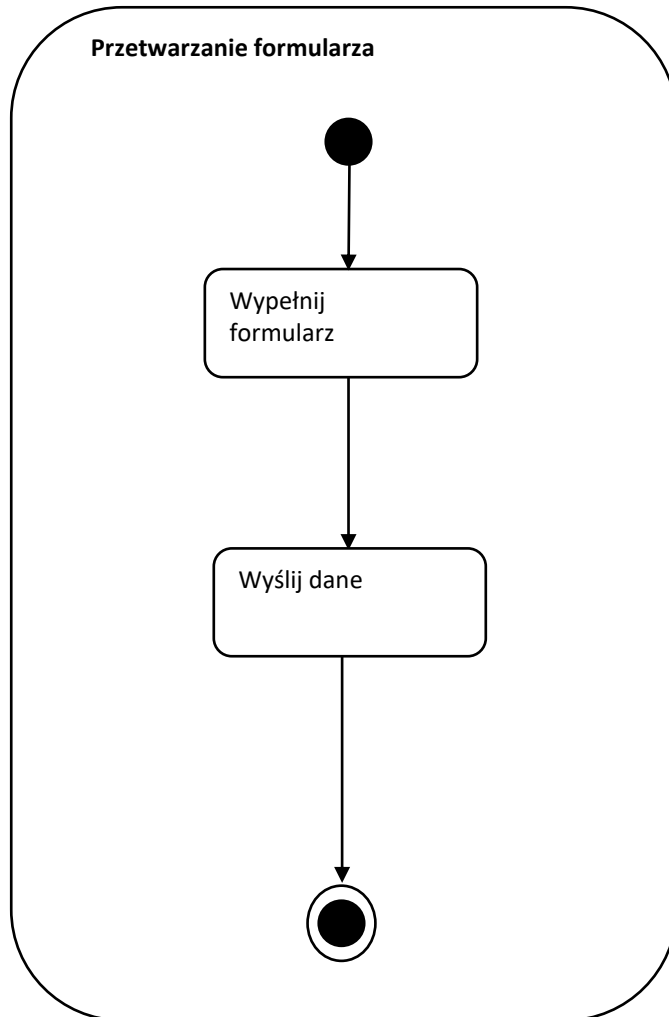
Przykładowy diagram aktywności sporządzony dla odbioru produktu.

Znajdują się na nim wszystkie wcześniej opisane elementy.

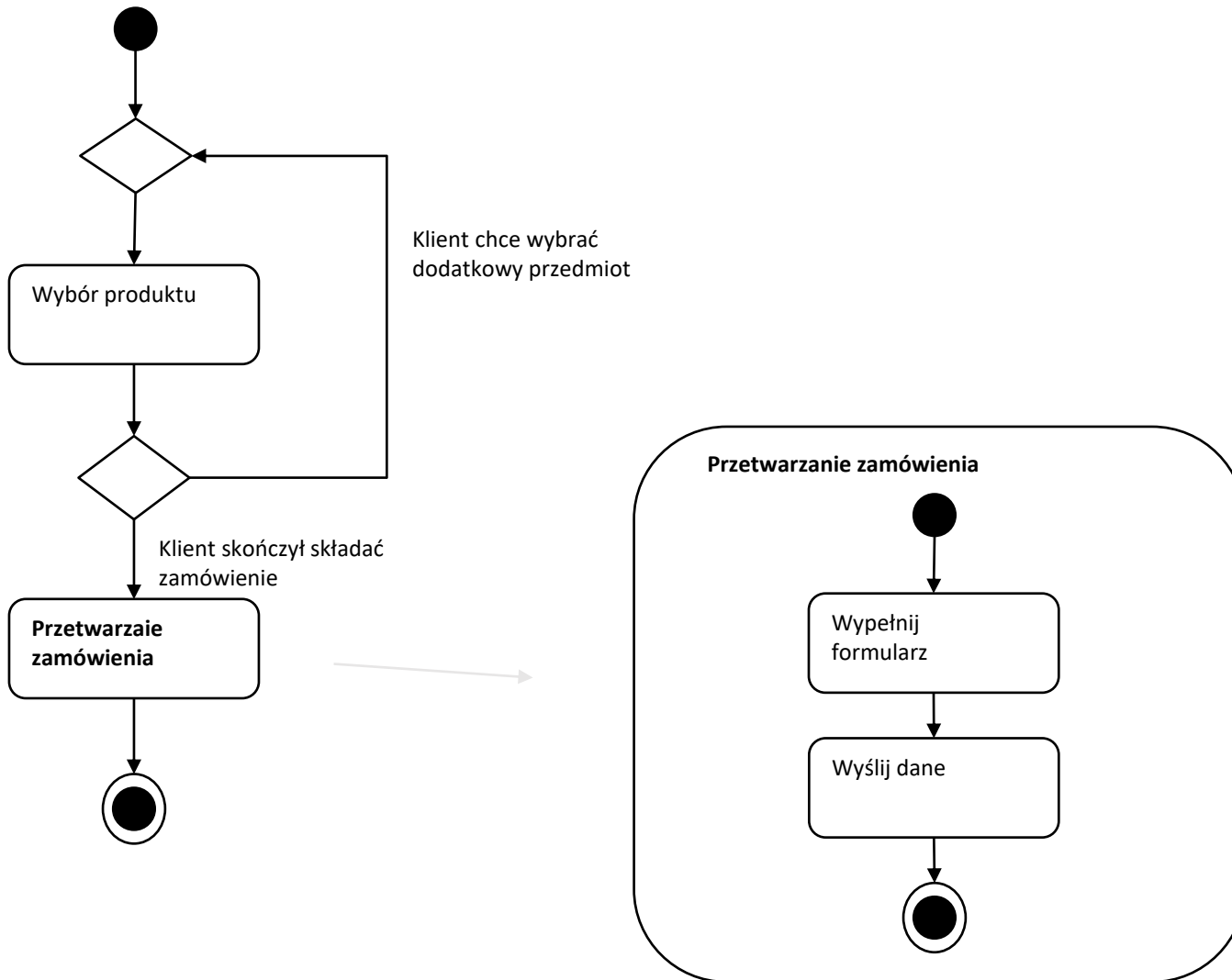
Zamodelowano również równoległe procesy, które także mogłyby zostać przedstawione sekwencyjnie.

Diagramy są kompatybilne ze specyfikacją techniczną języka **UML 1.4**.

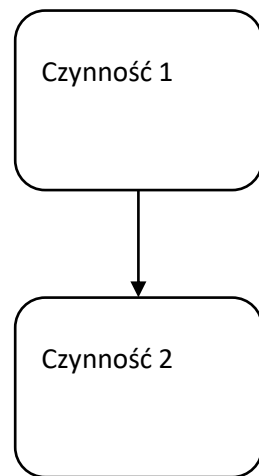
Specyfikacja **UML 2.0** wprowadza dodatkowe element do diagramów aktywności.



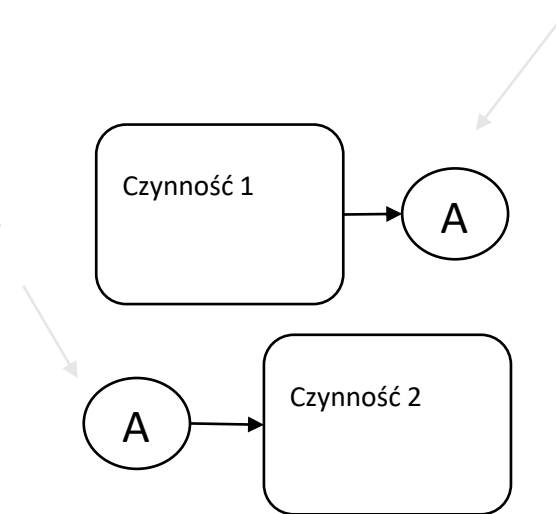
Czynność pozwala na grupowanie akcji. Akcja jest niepodzielnym zachowaniem wykonywanym w kontekście aktywności. W aktywności wykonywany jest jeden lub więcej przepływów sterowania.



Konektory



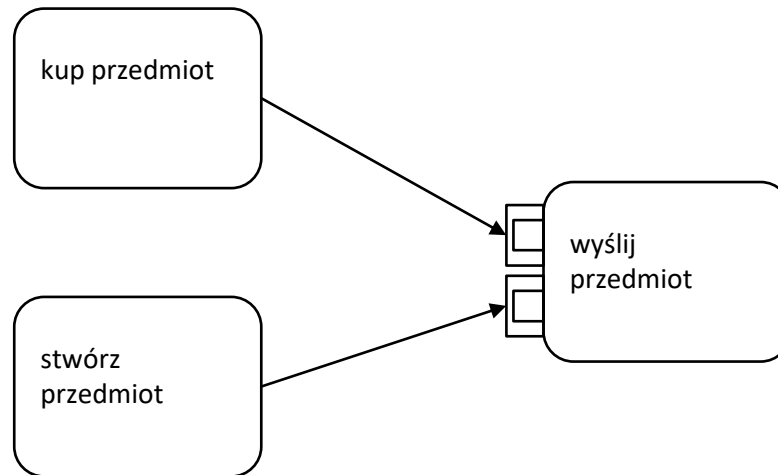
Konektor



W celu przeniesienia przepływu w inne miejsce na płaszczyźnie, można użyć konektorów.

Konektory zawsze występują w parach.
Jeden z krawędzią wchodzącą, drugi z wychodzącą.

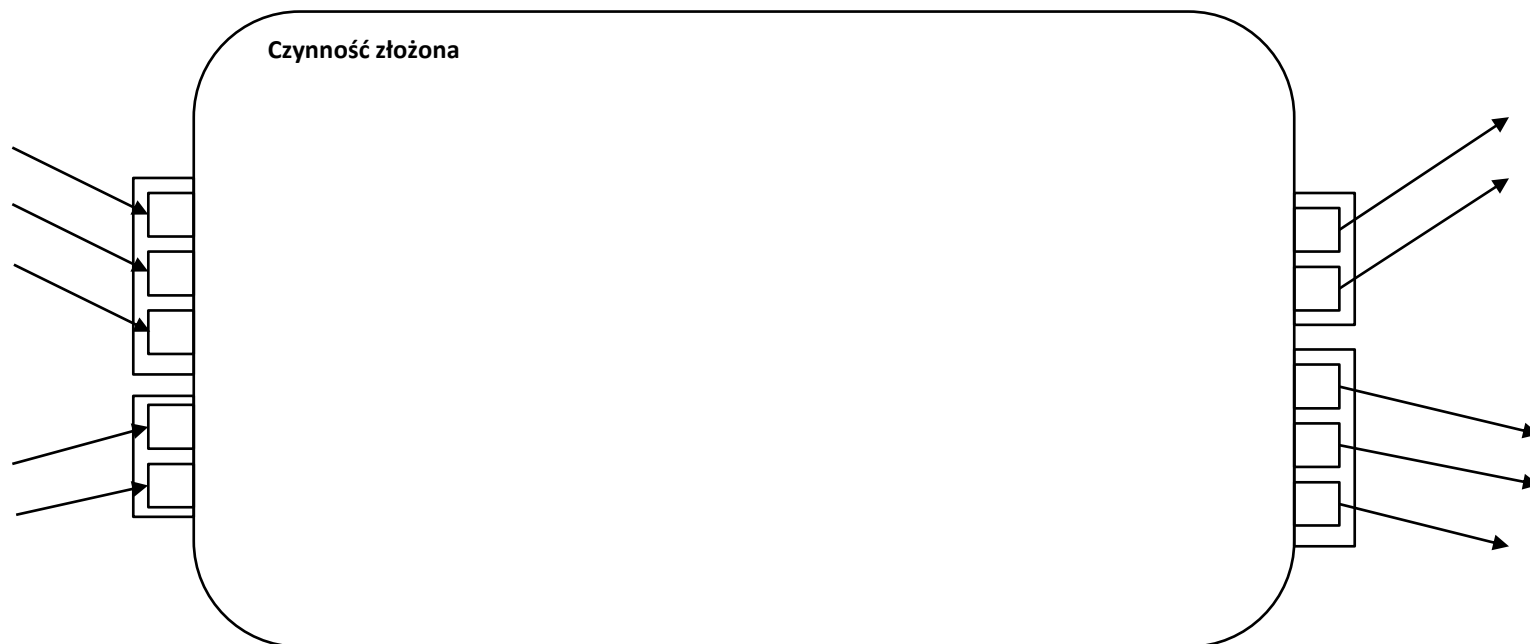
Wejścia/wyjścia czynności



Czynności mogą posiadać wejścia alternatywne.

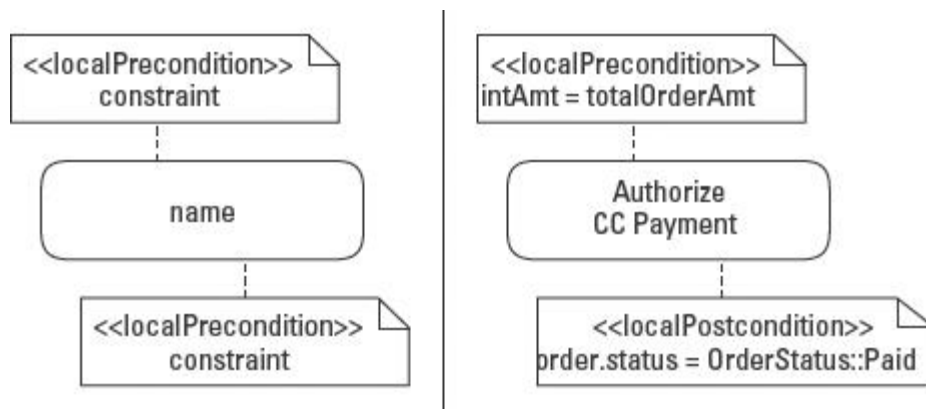
Czynność “wyślij przedmiot” zostanie wykonana po czynności “kup przedmiot” LUB “stwórz przedmiot”

Alternatywna czynność z wieloma wejściami/wyjściami



Czynność może posiadać wiele wejść i wyjść. Zazwyczaj są one traktowane jako warunek "and". Czasami jednak można zamodelować grupy alternatywnych parametrów (warunek "or").

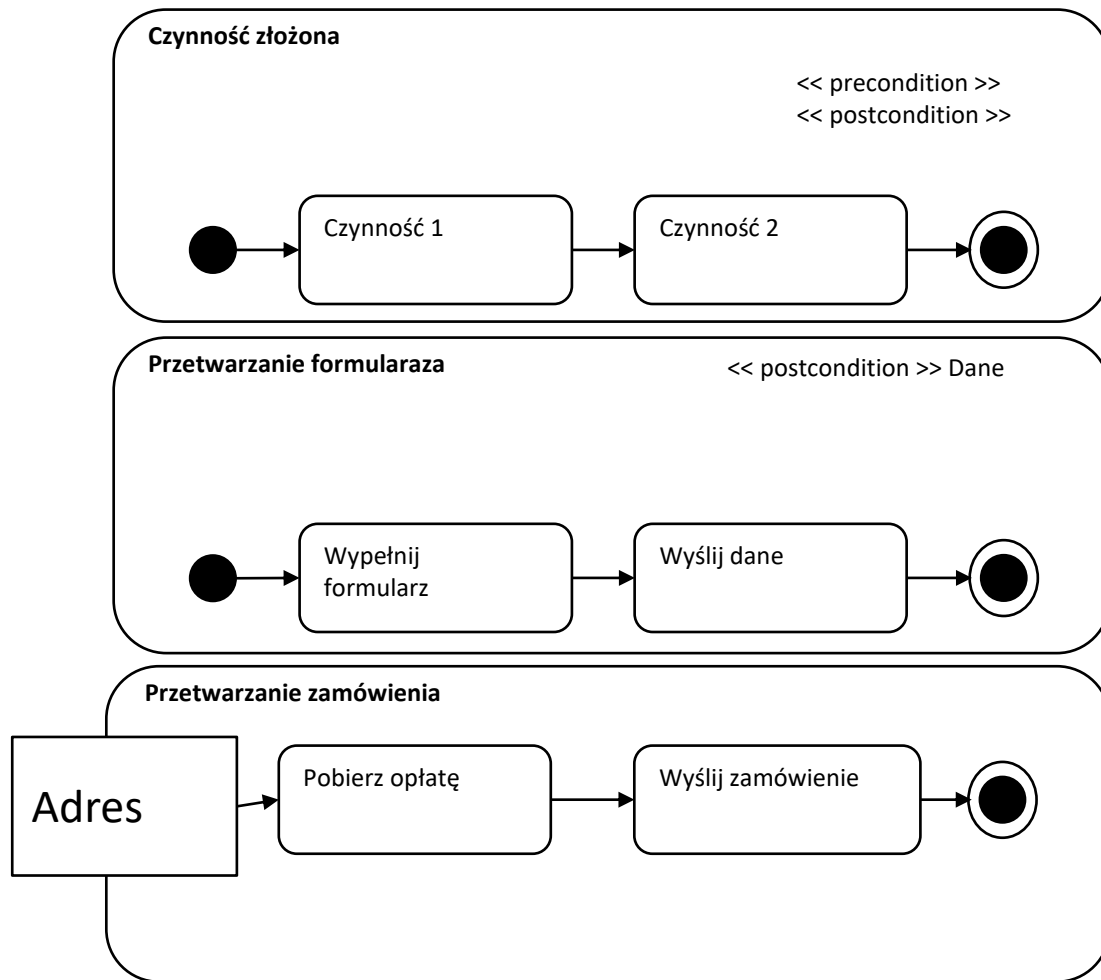
Warunki początkowe i końcowe



UML umożliwia definiowanie lokalnych warunków początkowych i końcowych, jakie muszą zaistnieć, aby akcja została wykonana.

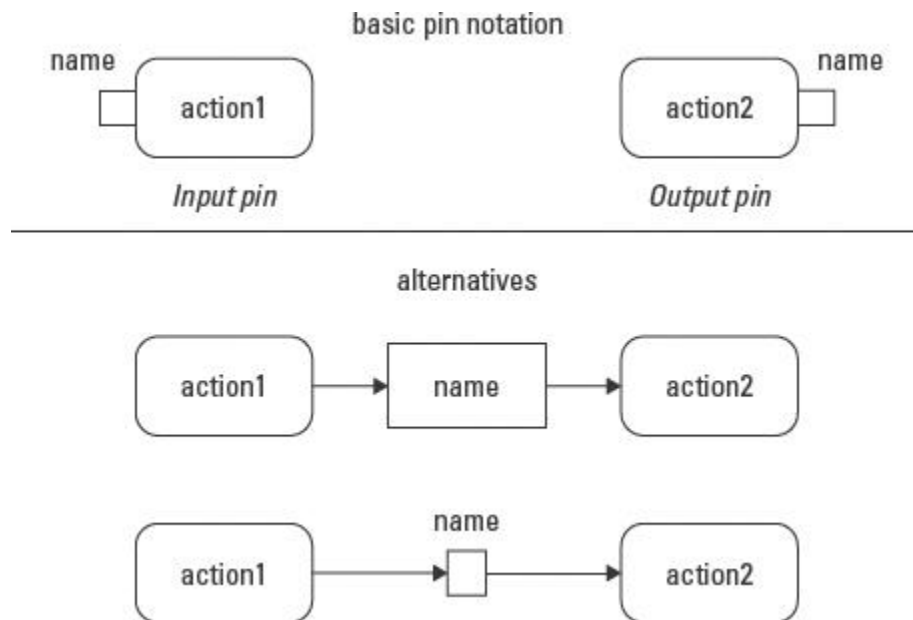
Tego typu podejście nie definiuje sposobu obsłużenia błędu wynikającego z niespełnienia warunków przez system. Ta odpowiedzialność spoczywa na wykonawcy implementacji.

Parametry/dane wejściowe i wyjściowe czynności



Czynności mogą również posiadać dane wejściowe oraz warunki potrzebne do ich wykonania.

Na rysunku obok przedstawiono alternatywną notację do poprzedniego slajdu.

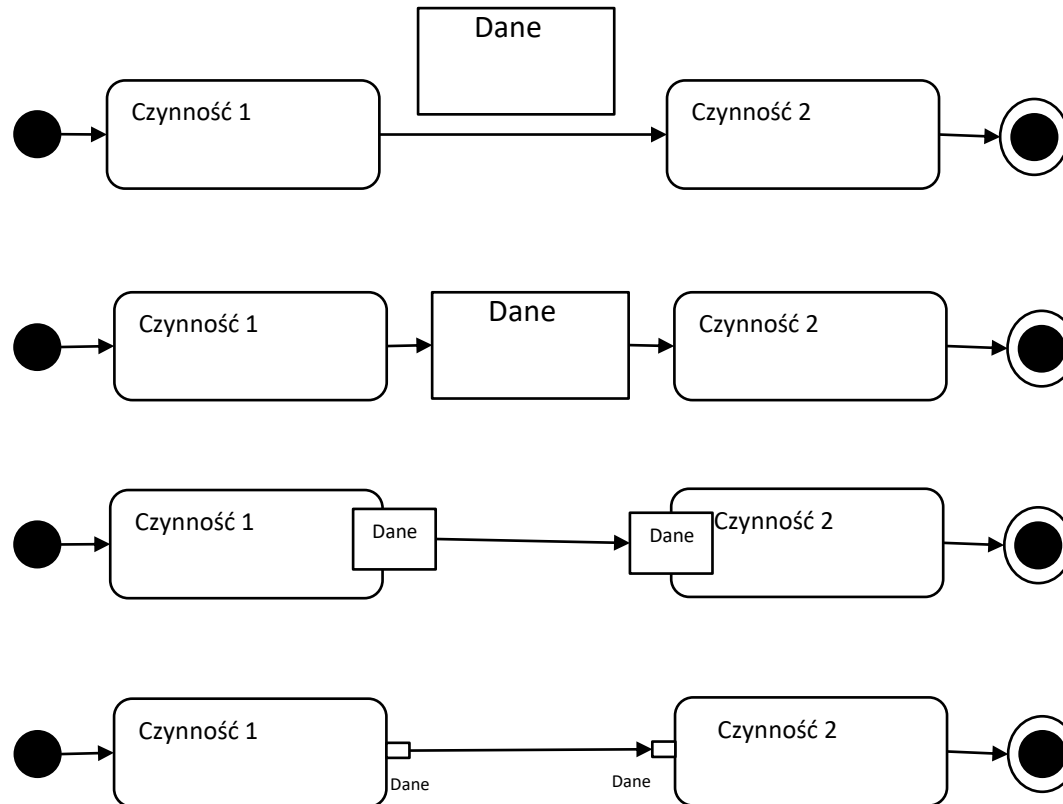


Dane akcji są modelowane przez piny.

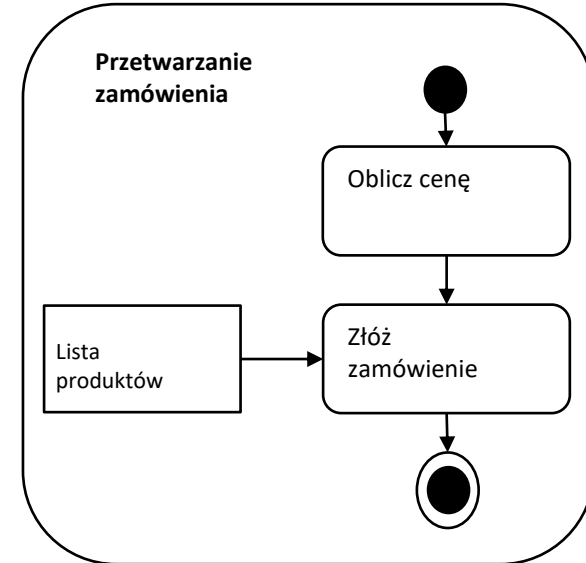
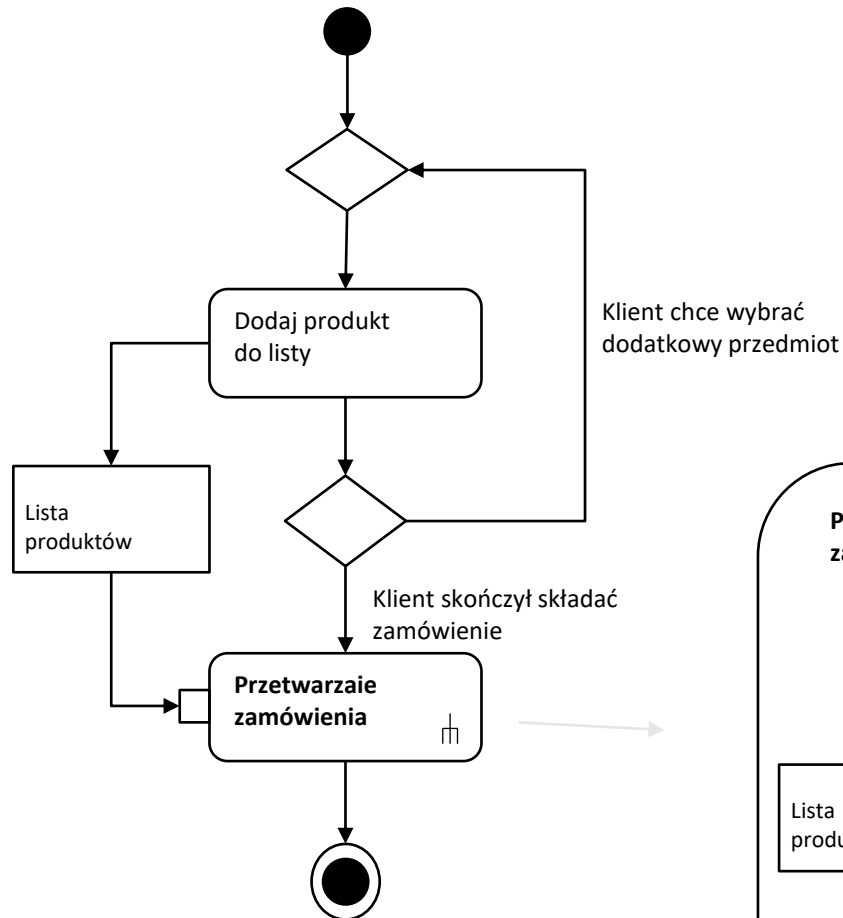
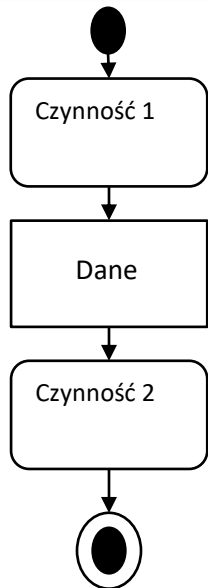
Akcje mogą mieć piny wejściowe i wyjściowe, każdy z nich zawiera pojedynczą wartość.

Kiedy piny następujących akcji odpowiadają sobie nawzajem możliwa jest notacja alternatywna.

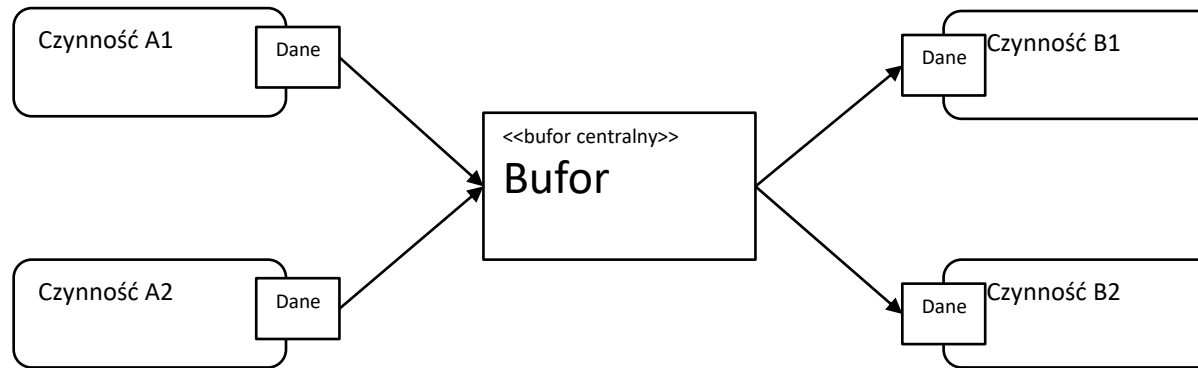
Różne sposoby oznaczania przepływu danych



Przepływy danych w procesie



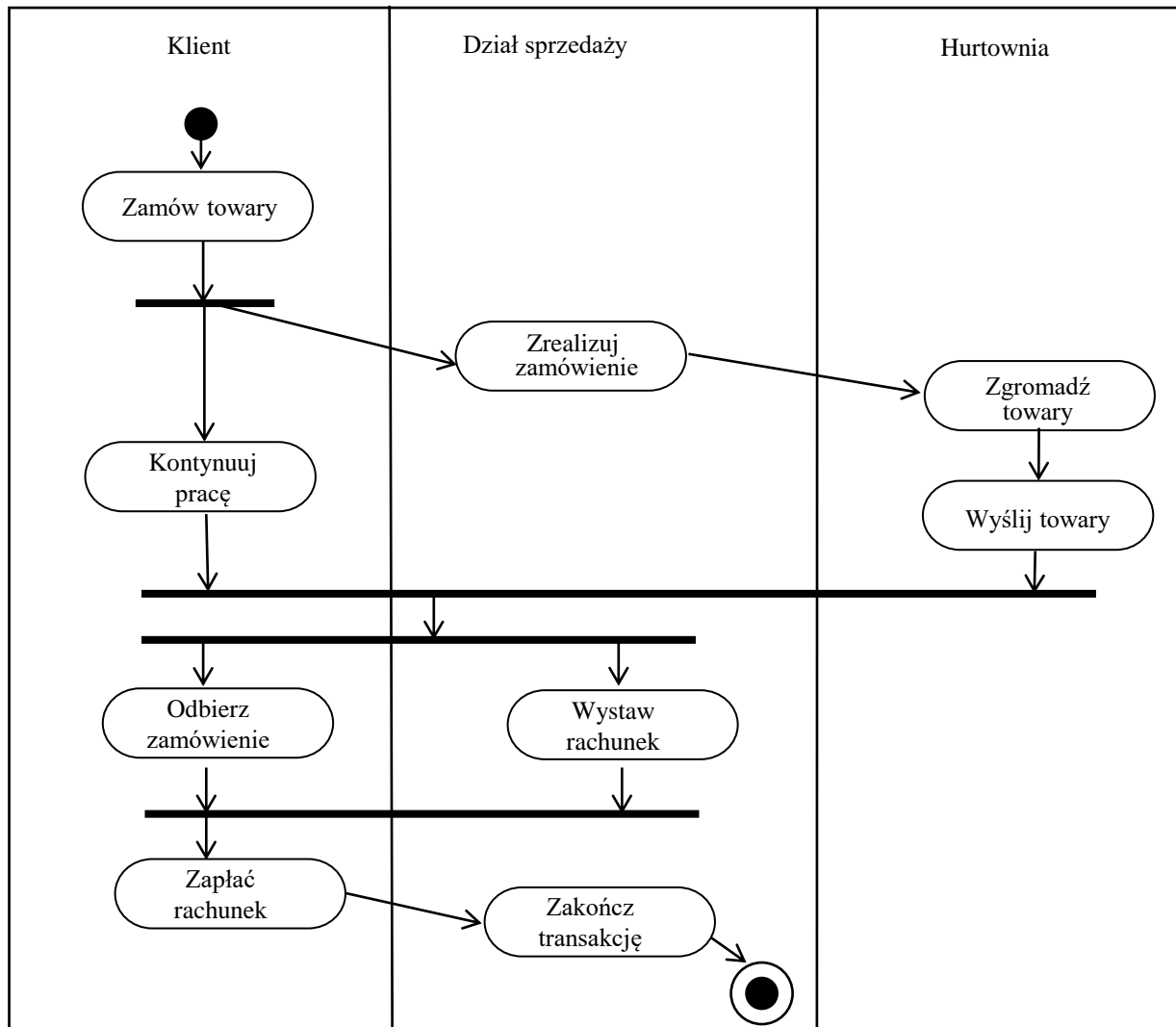
Bufor centralny



Bufor centralny może otrzymywać dane z dowolnej liczby pinów wyjściowych i zapewniać je dowolnej ilości wejść

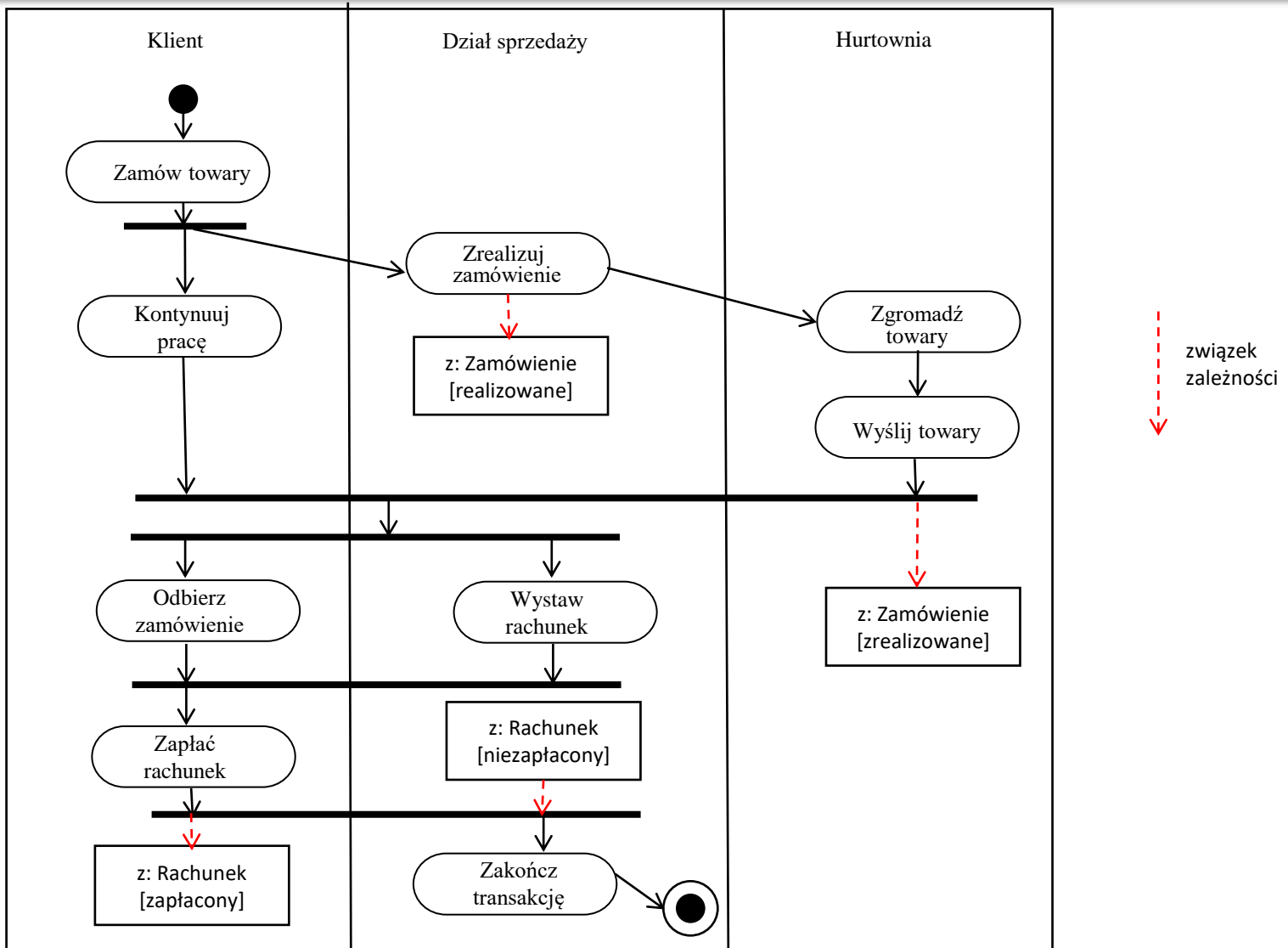
Tor (ang. *swimlanes*) na diagramach aktywności ma charakter organizacyjny, może odpowiadać pewnemu bytowi świata zewnętrznego (np. dział w przedsiębiorstwie), w którym są wykonywane określone czynności. Tory umożliwiają tym samym symboliczne rozlokowanie tych czynności między poszczególne elementy (działy przedsiębiorstwa).

Każdy tor ma nazwę, unikatową w obrębie jednego diagramu. Tor może zatem odpowiadać pewnemu bytowi świata rzeczywistego i może być modelowany przez co najmniej jedną klasę.

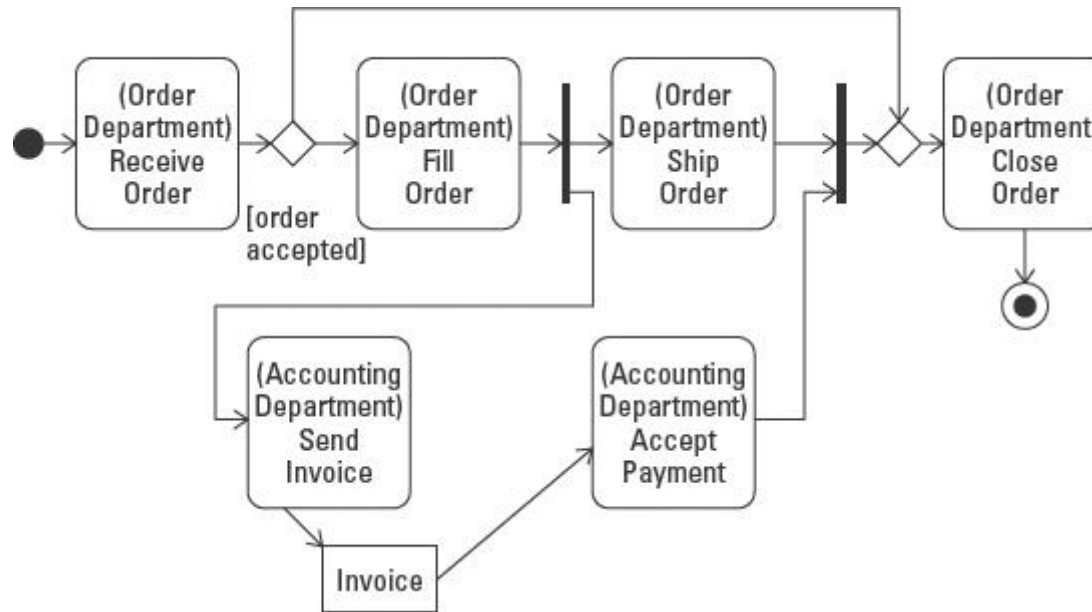


- Na diagramie aktywności można umieścić obiekty, łącząc je związkiem zależności z czynnością lub przejściem.
- Na diagramie można również umieścić zmianę roli, stanu i wartości każdego obiektu.
- Stan obiektu jest zapisywany w nawiasach kwadratowych tuż pod nazwą obiektu.
- Podobnie można wpisać wartości atrybutów - w dodatkowej sekcji poniżej nazwy.

Przeptyw obiektów

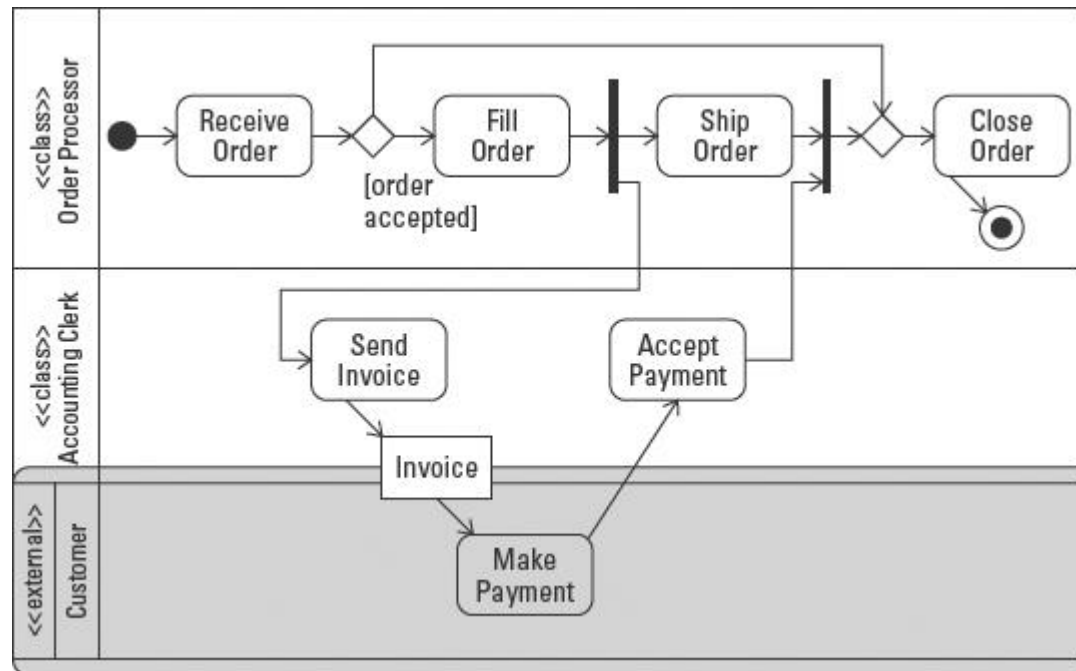


Notacja alternatywna dla torów



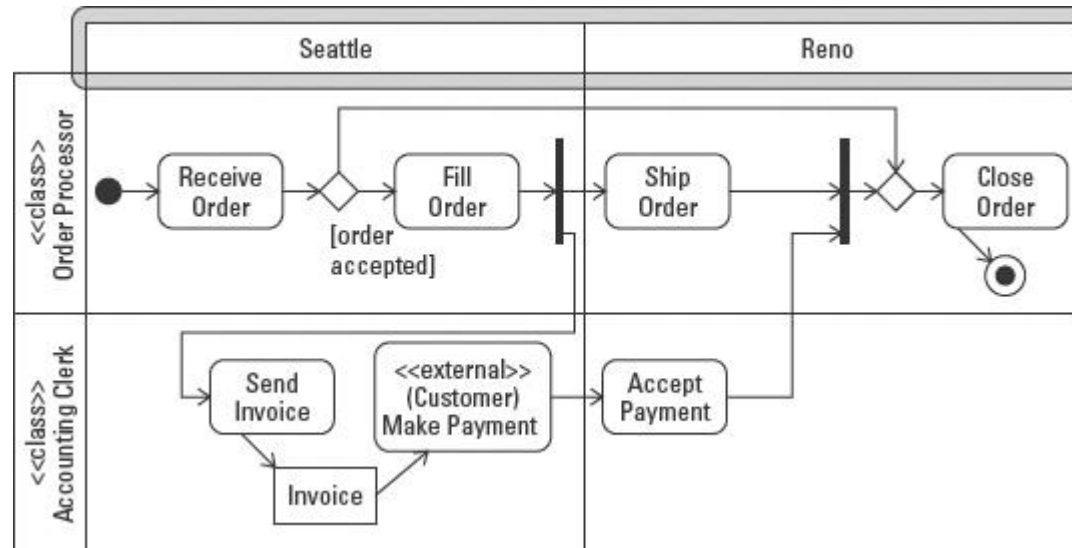
Alternatywnym sposobem notacji do torów jest poprzedzanie czynności odpowiednimi informacjami w okrągłych nawiasach, jest to jednak mniej czytelne.

Czynności zewnętrzne



Często zdarza się, że część procesu jest wykonywana przez podmiot zewnętrzny (jak np. klient na powyższym diagramie). Tego typu aktywność wymaga modelowania, jednak jest poza obszarem system, żeby to zaznaczyć warto użyć zewnętrznego toru.

Wielowymiarowe procesy



Dodatkowo tory umożliwiają wielowymiarowe modelowanie procesów. Przykładowo powyżej, poza funkcjami zamodelowane zostało jeszcze miejsce.

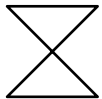
Sygnały asynchroniczne



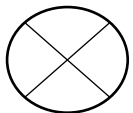
Wysyłanie



Odbieranie

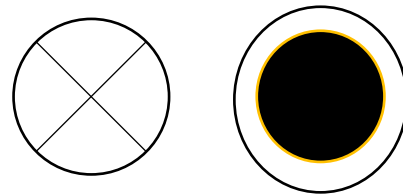


Czas

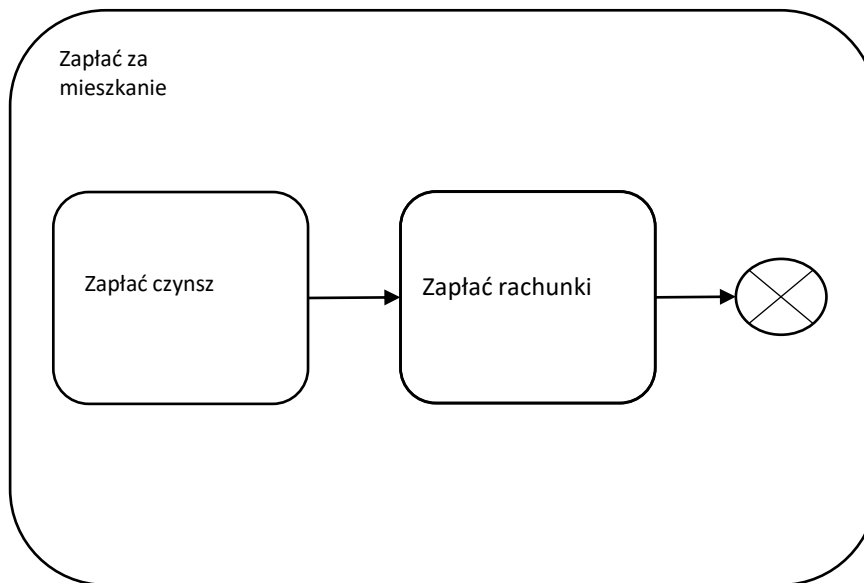


Koniec danego sygnału sygnału

Sygnały zakończenia

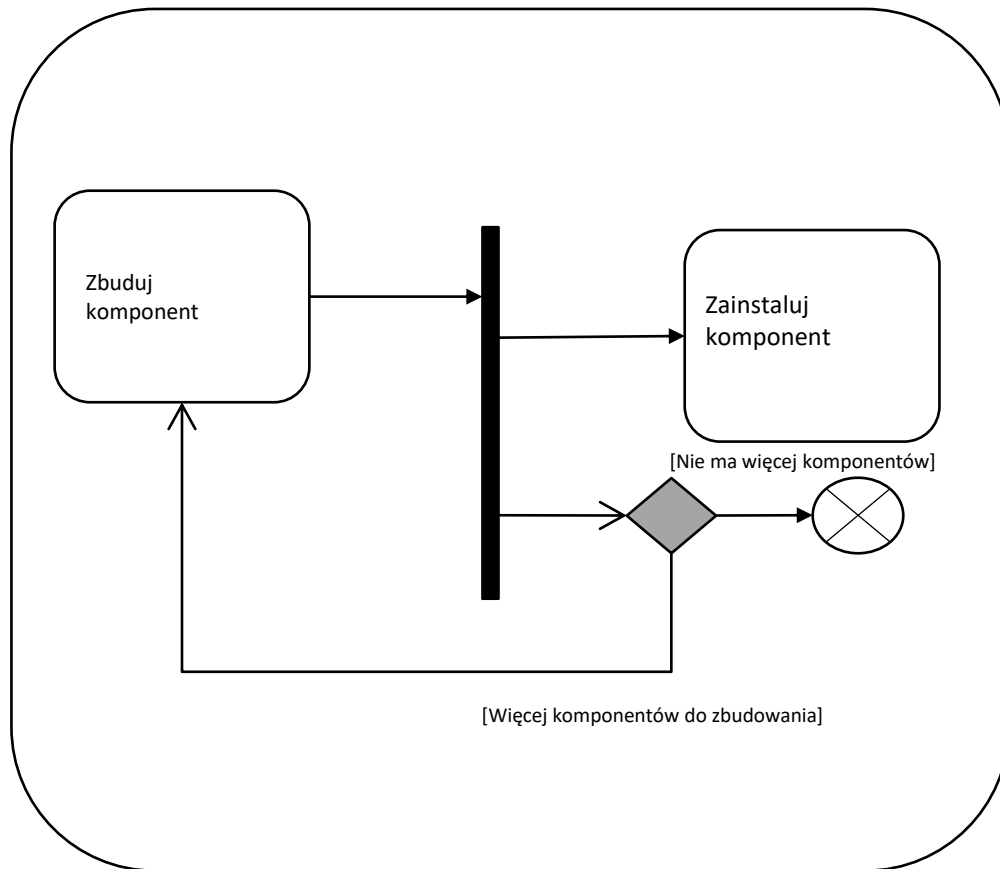


Pierwszy symbol oznacza koniec danego sygnału sterowania, nie kończy jednak całego procesu. W przypadku dotarcia do drugiego symbol następuje przerwanie wszystkich sterowań.



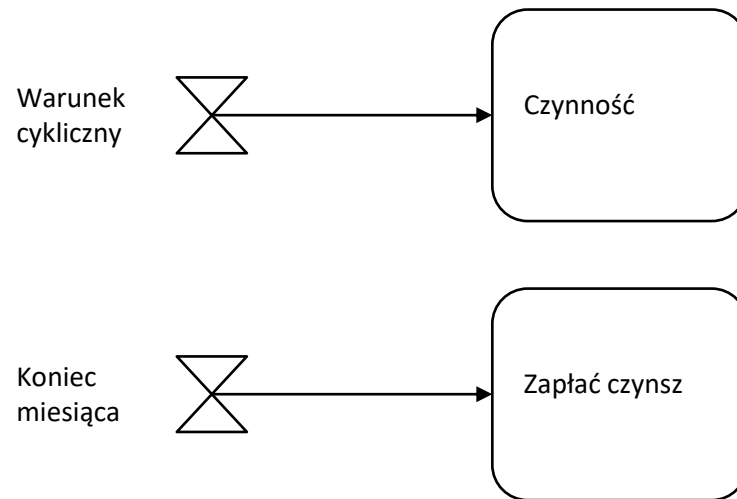
Czasami aktywność może mieć również swoje zakończenie, nie musi kończyć się wychodzącym przepływem sterowania, ani danymi.

Zakończenie pojedynczego przepływu

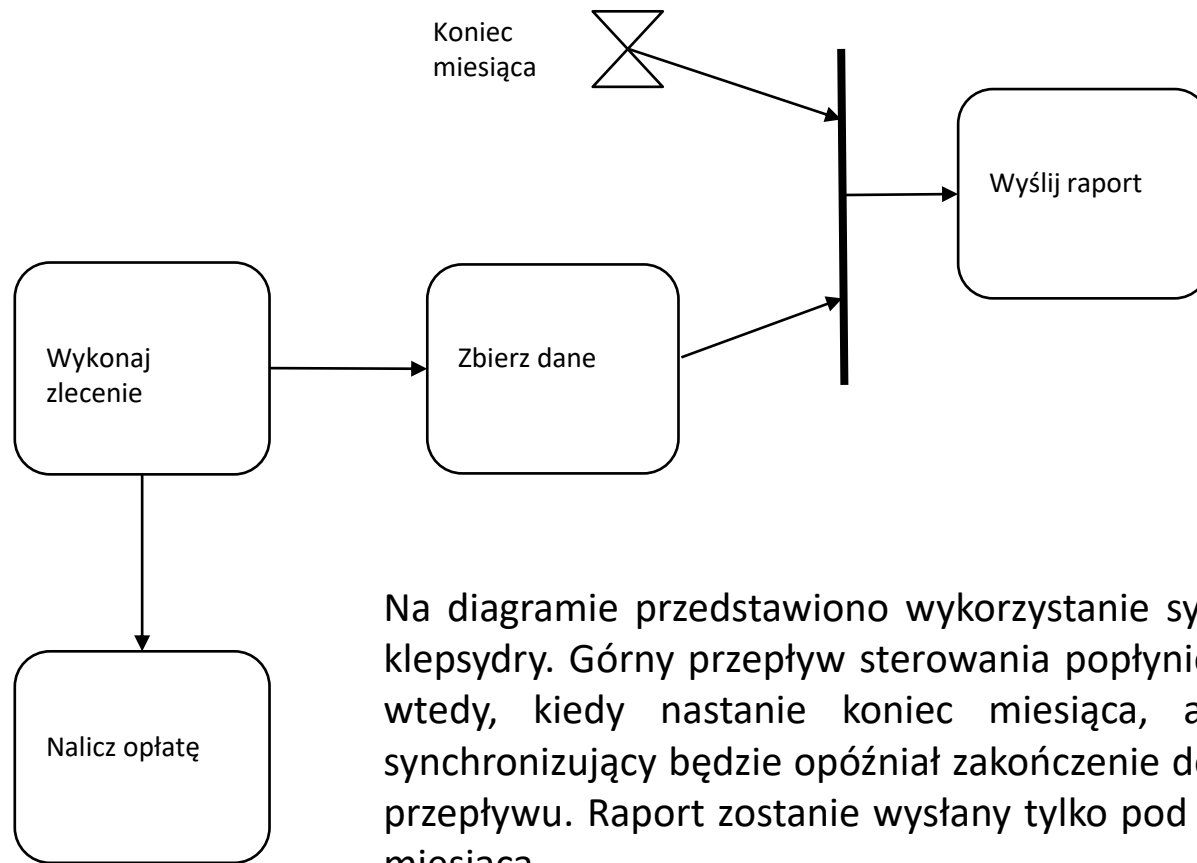


Aktywność również może posiadać wiele przepływów sterowania i równoległe wykonywanych czynność, a zakończenie jednej, wcale nie oznacza zakończenia pozostałych.

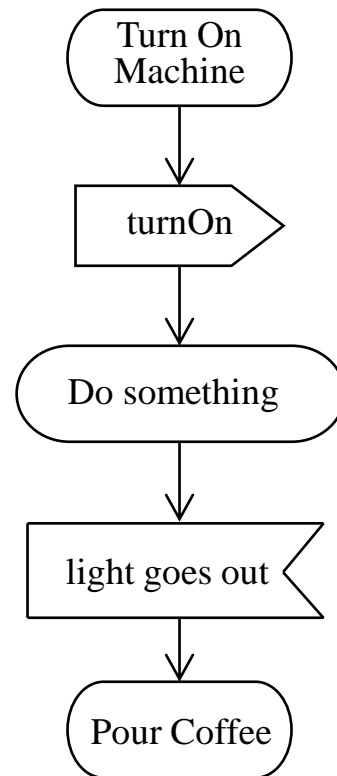
Symbol czasu

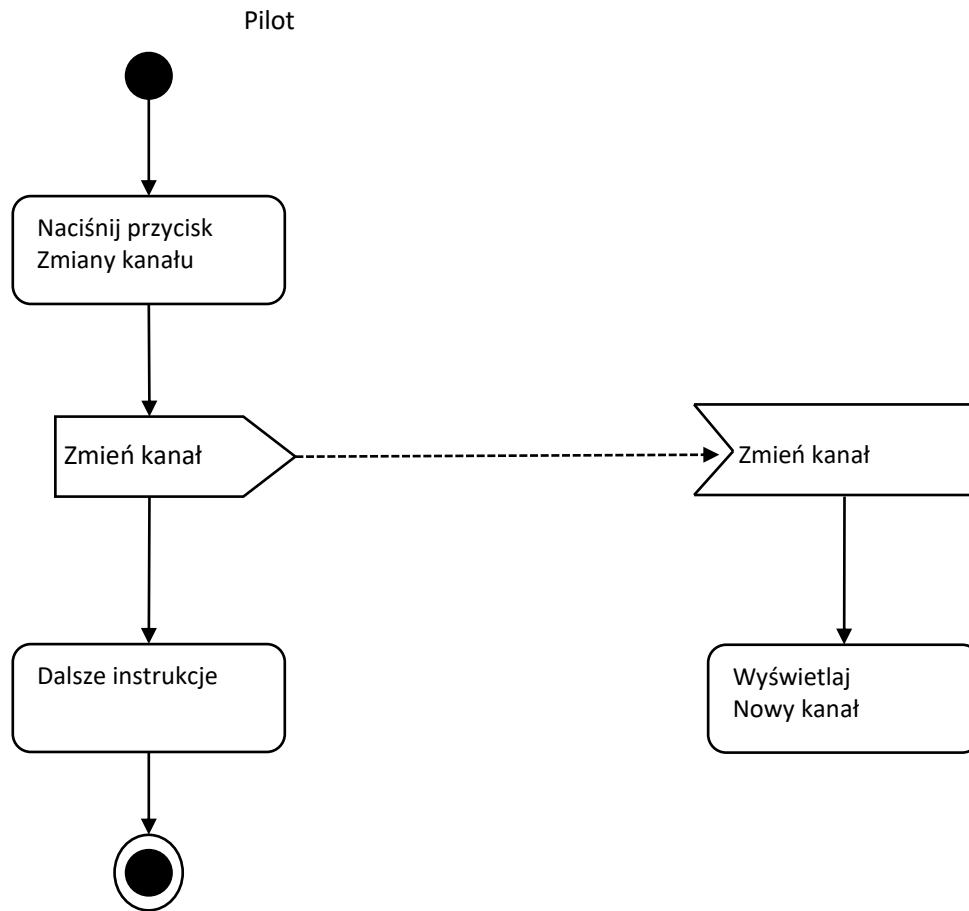


Symbol czasu używany jest do modelowania cyklicznych akcji. W przypadku powyższym czynność zostanie wykonana w przypadku spełnienia warunku cyklicznego.

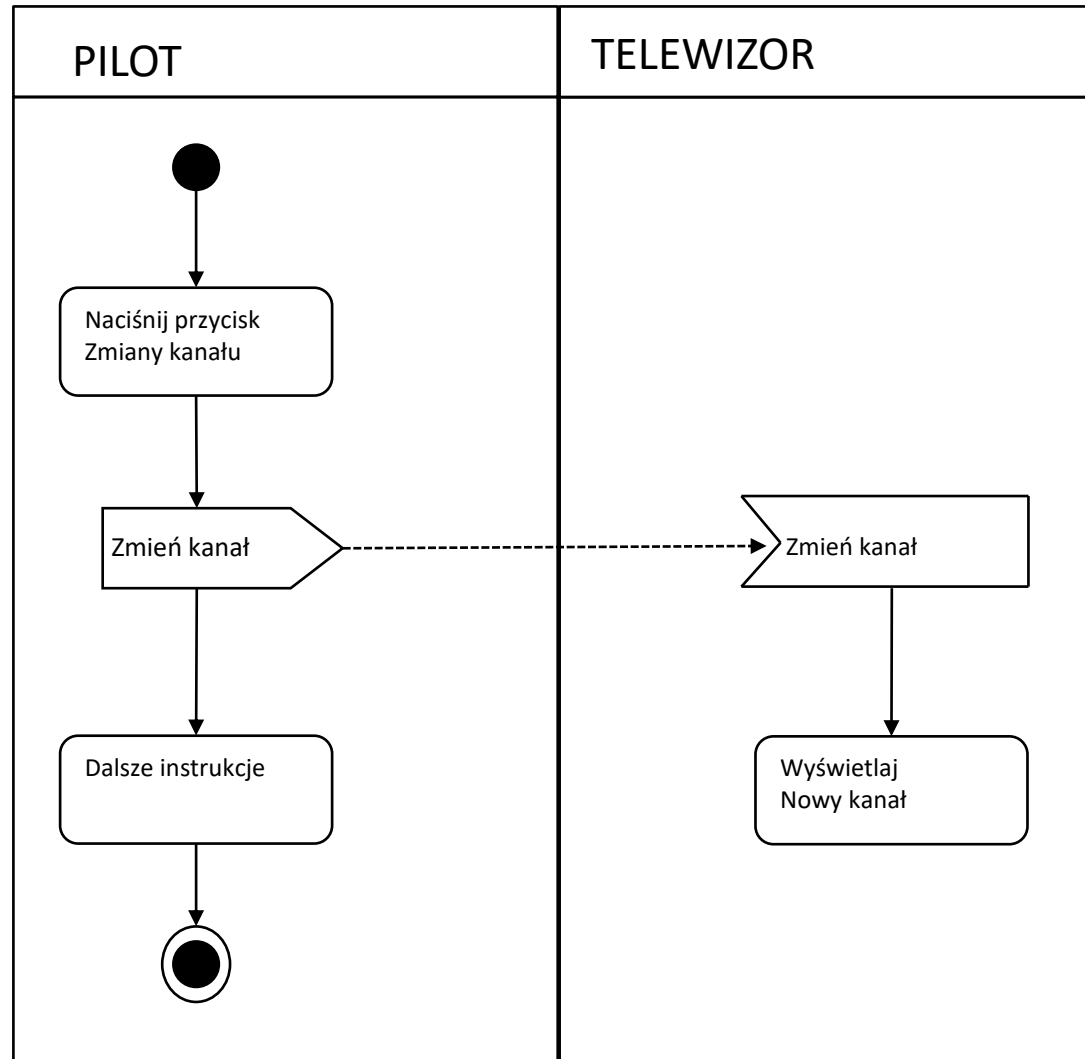


Na diagramie przedstawiono wykorzystanie symbolu klepsydry. Górny przepływ sterowania popłynie tylko wtedy, kiedy nastanie koniec miesiąca, a blok synchronizujący będzie opóźnił zakończenie dolnego przepływu. Raport zostanie wysłany tylko pod koniec miesiąca

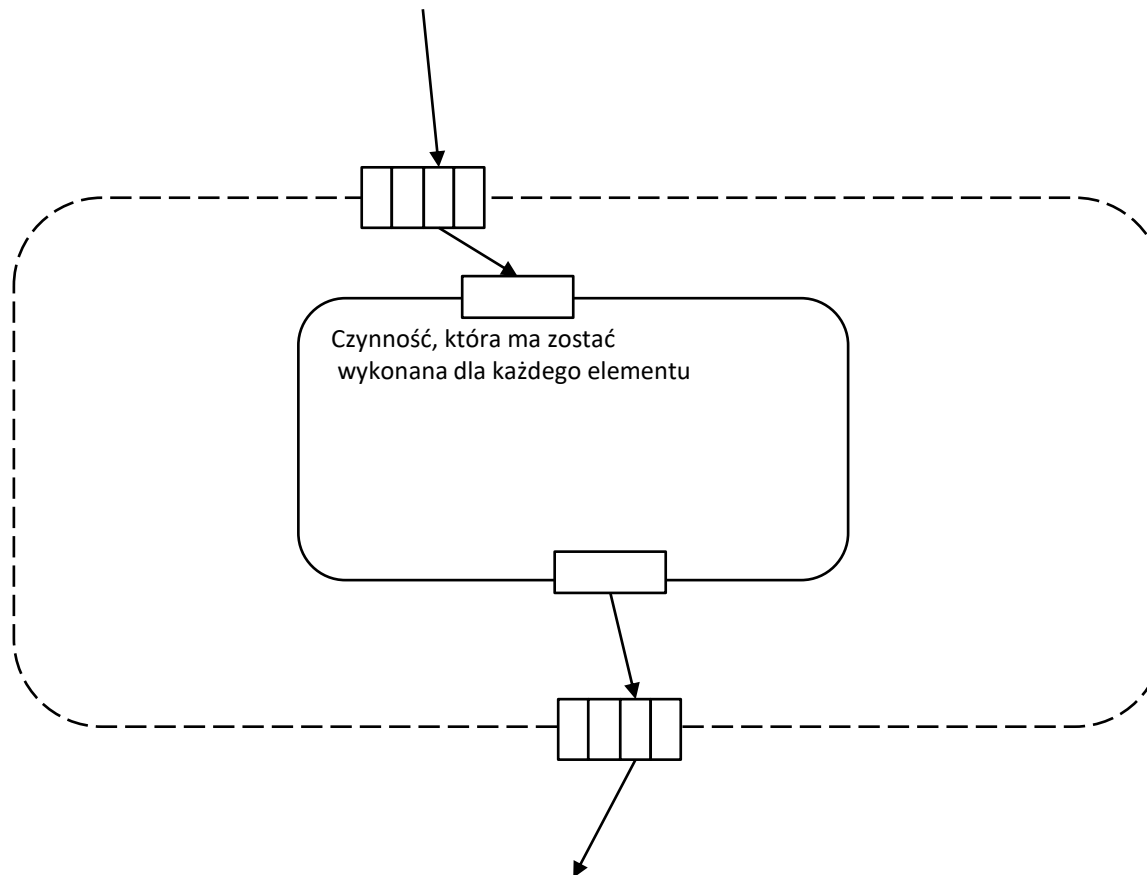




Sygnaly – wykorzystanie torów

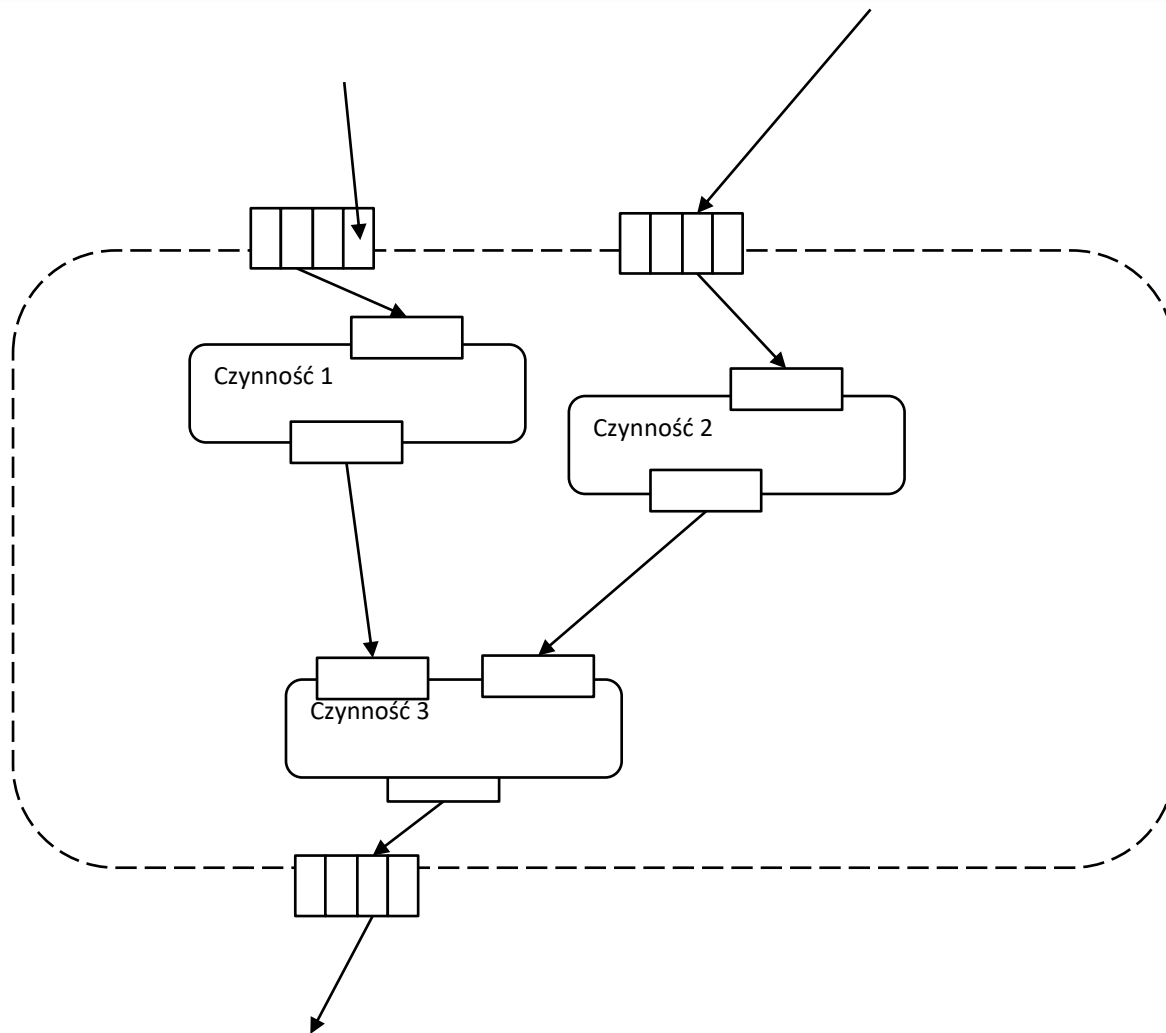


Sekwencja



Sekwencja wykonywana jest wielokrotnie, dla każdego z jej wejść. Produkjuje określoną ilość wyjść.

Sekwencja z wielokrotnym wejściem



Sekwencja może mieć wiele wejść, które wykonywane są równoległe. Wykonywanie sekwencji nie rozpocznie się, dopóki na obu wejściach nie ma odpowiednich danych. Wewnętrzny obszar wykonywany jest raz dla każdej z par wartości, z której produkowany jest jeden rezultat. Wszystkie rezultaty przekazywane są na wyjście.

Modelowanie dynamicznych aspektów systemu, w szczególności:

- 1. Modelowanie przepływu czynności - zwrócenie uwagi na czynności i ich analiza z punktu widzenia aktorów współpracujących z systemem.**
- 2. Modelowanie operacji – diagramy pełnią rolę schematów blokowych, w których przedstawia się szczegóły obliczeń. Otoczenie diagramu obejmuje parametry operacji i jej obiekty lokalne.**

Modelowanie przepływów czynności

1. Ustal najważniejsze czynności - nie da się zazwyczaj przedstawić wszystkich.
2. Wybierz elementy przedsiębiorstwa, które winny realizować bardziej złożony przepływ. Mogą to być elementy rzeczywiste lub abstrakcyjne. Utwórz tor dla każdego takiego obiektu.
3. Zidentyfikuj warunki wstępne stanu początkowego i warunki końcowe stanu końcowego dla modelowanego przepływu. W ten sposób łatwiej będzie określić granice przepływu.
4. Wychodząc ze stanu początkowego przepływu zapisuj kolejne czynności i akcje. Obrazuj je odpowiednio w postaci stanów czynności i stanów akcji.
5. Dla akcji złożonych lub zbiorów akcji zdefiniuj stany czynności i dla każdego z nich skojarz oddzielny diagram czynności, który przedstawia zebrane w nim akcje.
6. Zobrazuj przejścia między stanami czynności i stanami akcji. Zacznij od przepływów sekwencyjnych, następnie określ rozgałęzienie, potem rozwidlenia i scalenia.
7. Jeśli w przepływie występują obiekty, to umieść je na diagramie czynności. Uwzględnij zmieniające się atrybuty i stany obiektów.

Modelowanie operacji

1. Rozważ abstrakcje uczestniczące w modelowanej operacji: parametry, atrybuty klasy otaczającej i innych sąsiednich klas.
2. Zidentyfikuj warunki wstępne stanu początkowego i warunki końcowe stanu końcowego modelowanej operacji. Znajdź niezmienniki klasy otaczającej, które muszą być zachowane podczas realizacji operacji.
3. Począwszy od stanu początkowego zapisuj kolejne czynności i akcje oraz przedstawiaj je odpowiednio na diagramie.
4. W miarę potrzeby korzystaj z rozgałęzień do modelowania ścieżek warunkowych i iteracji.
5. Jeśli operacja należy do klasy aktywnej (i tylko wtedy), skorzystaj z rozwidleń i scaleń, modelując tym samym równoległy przepływ sterowania.