

UMLUnified Modeling
Language**Wzorce
projektowe****OCL**Object Constraint
Language**PROJEKTOWANIE
SYSTEMÓW
KOMPUTEROWYCH**
notatki w internecie[KONTAKT](#)[POBIERZ PDF](#)[ŹRÓDŁA](#)[NARZĘDZIA UML](#)[HOME](#)**Teoria****Diagramy UML**

- diagram przypadków użycia
- diagram aktywności
- diagram klas
- diagram obiektów
- diagram sekwencji
- diagram komunikacji
- diagram czasowe
- przeglądowe diagramy interakcji
- diagram komponentów
- diagram pakietów
- diagram maszyny stanowej
- diagram wdrożenia
- diagram struktur złożonych

Język OCL**Kreacyjne wzorce projektowe****Strukturalne wzorce projektowe****Czynnościowe wzorce projektowe**

Diagramy maszyny stanowej - *state machine diagram*

Diagramy maszyny stanowej stanowią część **logicznego widoku** modelu w systemie 4+1 Kruchtena

RODZAJE MASZYN STANOWYCH

Maszyny stanowe można podzielić na:

maszyny stanowe zachowania, protokołowe maszyny stanowe.

Maszyny zachowania przedstawiają przejścia między stanami wielu obiektów w szerszym kontekście zachowania systemu, podsystemu czy też przypadku użycia. Natomiast protokołowe maszyny stanowe koncentrują się na konkretnym obiekcie. Przedstawiają dozwolone przejścia pomiędzy stanami tego obiektu.

W przypadku opracowania dla danej dziedziny przedmiotowej obydwu wymienionych rodzajów maszyn stanowych muszą one być wzajemnie niesprzeczne.

PROKOŁOWA MASZYNA STANOWA

Protokołowa maszyna stanowa reprezentuje wszystkie sekwencje operacji na obiekcie, które powodują uaktywnienie poszczególnych stanów tego obiektu. Operacje, które nie generują przejścia obiektu w inny stan, nie są reprezentowane w protokołowej maszynie stanowej. Ten rodzaj diagramów maszyn stanowej oznacza, że nie wskazuje się wewnętrznych czynności stanów, natomiast zaznacza operacje dotyczące przejść między poszczególnymi stanami.

Opis przejścia w protokołowej maszynie stanowej charakteryzuje się następującą składnią:

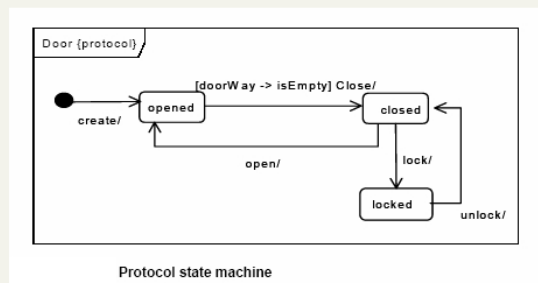
```
"[" < warunek--wstępny > "]" < nazwa-operacji > "/" "[" < warunek--końcowy > "]"
```

Protokołowe maszyny stanowe mogą wystąpić w następujących postaciach:

deklaratywnej, która w sposób ogólny wskazuje ciąg dopuszczalnych stanów i przejść dotyczących konkretnego obiektu - jest to standardowa postać protokołowej maszyny stanowej;

wykonawczej, zawierającej ścisłą i precyzyjną specyfikację przejść oraz związanych z nimi operacji, umożliwiającą bezpośrednie generowanie kodu źródłowego.

Graficznie protokołowe maszyny stanowe zawierają większość kategorii i pojęć maszyn stanowych zachowania. Przedstawia się je w postaci obramowanego diagramu, w nagłówku którego umieszcza się wyróżnik typu diagramu, uzupełniony opcjonalnie o ograniczenie {**protocol**}. Diagramy ilustrujące protokołowe maszyny stanowe mogą stanowić wsparcie w definiowaniu interfejsów i portów na diagramach komponentów.

[Zobacz przykład](#)

MASZYNY STANOWE ZACHOWANIA (ANG. BEHAVIORAL STATE MACHINES)

Przedstawiają przejścia między stanami obiektów w szerszym niż w protokołowych maszynach stanowych kontekście zachowania systemu, podsystemu, przypadku użycia i innych obiektów. Opcjonalny opis przejścia w maszynie stanowej zachowania charakteryzuje się składnią zawierającą pojęcia zdarzenia, warunku i czynności, jak następuje:

```
[< sygnatura-zdarzenia >] ["[" < warunek > "]" ] "/" < wyrażenie-czynności >
```

DIAGRAM STANU

Diagram stanu reprezentuje zachowanie obiektu o skończonej liczbie stanów i zdefiniowanych przejściach między nimi. Diagram stanów jest to **graf**, który reprezentuje **maszynę stanów**. Przejście między stanami jest wyzwolone asynchronicznym zdarzeniem zewnętrznym.

Podstawowymi elementami diagramu są **stany obiektu połączone strzałkami przejść**. Obiekt, reagując na nadchodzące zdarzenia, jeżeli spełnione są określone warunki, zmienia swój stan i położenie na diagramie stanu.

STAN

Stan jest etapem cyklu życia obiektu. Obiekt przebywający w danym stanie spełnia określony warunek.

```
event-signature [guard condition] / action_expression
```

gdzie event-signature to: event-name (parameter-list)

Stan złożony (composite) ma swoją dekompozycję graficzną.

Stan może być użyty do modelowania czynności ciągłej (ongoing activity).

Opcjonalnie symbol stanu można podzielić na poziome zaznaczone przedziały, które opisują: **nazwę, akcje wewnętrzne i przejścia wewnętrzne postaci**:

```

action-label '/' action-expression
np. entry / set echo invisible help / display help

```



Pojedynczy stan reprezentuje moment w zachowaniu obiektu, w którym pewien warunek jest prawdziwy. Stany są reprezentowane przez prostokąty z zaokrąglonymi narożnikami. Każdy stan ma swoją nazwę.

Ze stanem mogą być związane pewne **akcje**, wykonywane w określonym momencie:

- **entry** - jest akcją wykonywaną w momencie gdy obiekt przyjmuje dany stan; akcja ta jest wykonywana jeden raz i niepodzielnie
- **do** - jest akcją wykonywaną nieprzerwanie w czasie, gdy obiekt przebywa w tym stanie
- **exit** - oznacza (analogicznie do entry) moment opuszczenia stanu; podobnie, akcja taka jest wykonywana tylko raz.
- **event** - reprezentuje akcję wykonywaną w momencie nadejścia zdarzenia określonego typu.
- **include** - wywołanie zagnieżdżonej maszyny stanu,

Wykonanie każdej z tych akcji może również generować zdarzenie.

Ile obiekt może mieć stanów?

Bardzo dużo. Jeżeli np. może być 1 000 000 nazwisk, 1 000 imion i 100 000 firm, to liczba stanów wynosi 100 000 000 000 000. Nawet dla małego obiektu liczba stanów może być duża.

Ile stanów ma cała baza danych?

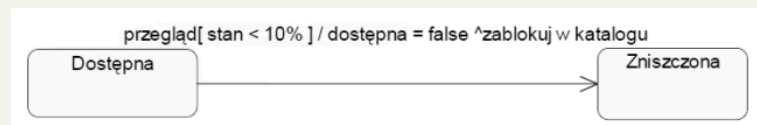
Bardzo, bardzo dużo: iloczyn liczby wszystkich możliwych stanów dla każdej maszyny stanu przez liczbę wszystkich obiektów wszystkich klas.

Stan, w podstawowym znaczeniu, dotyczy pewnego fragmentu historii życia obiektu i można go charakteryzować na trzy uzupełniające się sposoby:

- jako zbiór wartości obiektu (atrybutów i powiązań) w pewnym aspekcie podobnych (rozważane jest tu podobieństwo jakościowe),
- jako okres czasu, w którym obiekt oczekuje na zdarzenie,
- jako okres czasu, w którym obiekt przetwarza.

PRZEJŚCIA

Przejście powoduje zmianę stanu i wykonanie pewnych akcji.



Stany są powiązane ze sobą przejściami. Przejścia definiują warunki, jakie muszą zaistnieć, aby obiekt zmienił swój stan z źródłowego na docelowy. Formalnie opis przejścia składa się z czterech elementów:

- **wyzwalacza (ang. trigger)** - zdarzenia, które może spowodować przejście i zmianę stanu
- **dozoru (ang. guard condition)** - warunku, jaki musi być spełniony, aby przejście zostało wykonane; warunek ten jest ewaluowany w momencie pojawienia się wyzwalacza
- **akcji (ang. action)** - operacji wykonywanej w momencie przejścia ze stanu do stanu; nawet jeżeli akcja przejścia jest złożona z wielu akcji elementarnych, jest ona wykonywana niepodzielnie
- **zdarzenia (ang. event)** - wysłanego w momencie wykonania przejścia.



STANY ZŁOŻONE

Stany złożone posiadają wewnętrzną maszynę stanów. Wejście do stanu jest jej stanem początkowym, a wyjście - końcowym.



Dotychczas była mowa o stanach prostych. Są one niepodzielne - znalezienie się obiektu w takim stanie ma zawsze taki sam efekt i pomija ewentualne zmieniające się zewnętrzne okoliczności.

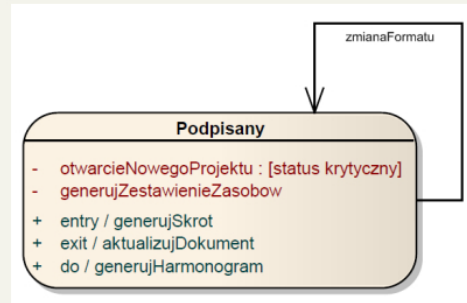
W niektórych sytuacjach wewnątrz stanu można jednakże wyróżnić podstany. Innymi słowy, wewnątrz stanu znajduje się inny diagram stanu. Diagram podstanów jest przetwarzany w sposób zbliżony do zwykłego diagramu stanu. Jednak w ogólnym przypadku stan złożony dopuszcza także istnienie podstanów współbieżnych, co oznacza, że obiekt znajdujący się w jednym stanie jednocześnie znajduje się w kilku podstanach. Wówczas podstany równoległe tworzą niezależne regiony wewnątrz stanu zewnętrznego, w których przejścia następują niezależnie od siebie.

Wejście do stanu powoduje także wejście wszystkich podstanów początkowych we wszystkich regionach. Następnie przejścia są realizowane równoległe i niezależnie we wszystkich regionach, aż do podstanów końcowych. Przejście do stanu końcowego we wszystkich regionach powoduje uruchomienie zdarzenia zakończenia stanu i skojarzonych z nim wyzwalaczy.

Przykład dotyczy stanu Otwarty, reprezentującego otwarty stan Rezerwacji książek. Rezerwacja może objąć do 4 książek jednocześnie. Stan Rezerwacji pozostaje otwarty w trakcie dodawania kolejnych książek, jednak wyróżniono w nim podstany: Wyszukiwanie informacji o książce, Weryfikację, czy danej książki już wcześniej nie zarezerwowano oraz Aktualizację danych Rezerwacji. Wszystkie podstany prowadzą do opuszczenia stanu przez Rezerwację, co jest związane np. z próbą dodania do niej nowej książki.

SEKCJA PRZEJŚĆ WEWNĘTRZNYCH

Sekcja przejść wewnętrznych (ang. *internal transitions compartment*) wskazuje szczególne przypadki przejść, których wykonanie nie prowadzi do zmiany danego stanu. Przejść wewnętrznych nie należy utożsamiać z przejściami zwrotnymi, których stan źródłowy i docelowy są tym samym stanem, mimo że znaczenie jest bardzo zbliżone. W obu przypadkach wykonanie czynności związanej z przejściem nie prowadzi do zmiany stanu. O ile jednak w przypadku przejścia zwrotnego są wykonywane czynności wejściowe (entry) oraz wyjściowe (exit) danego stanu, w przypadku przejść wewnętrznych są one pomijane. Przejścia wewnętrzne nie mają odrębnego oznaczenia graficznego, lecz są specyfikowane w obszarze stanu.



PSEUDOSTAN W MODELOWANIU DIAGRAMU MASZyny STANOWEJ

NOTACJA



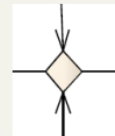
Stan początkowy (initial state) to zainicjowanie maszyny stanowej. Reprezentuje obiekt w momencie jego utworzenia. Każdy diagram stanu może zawierać tylko jeden taki stan. Do stanu początkowego nie dochodzą żadne przejścia.



Stan końcowy reprezentuje usunięcie obiektu z systemu. Stan ten jest opcjonalny (nie wszystkie obiekty są usuwane), w systemie także może występować wiele różnych stanów końcowych. Ze stanów końcowych nie można przejść do innych stanów.



Przejście (transition) to relacja pomiędzy dwoma stanami wskazująca, że obiekt znajdujący się w pierwszym stanie wykona pewne akcje i przejdzie do drugiego stanu.



Decyzja, przedstawiająca wybór pomiędzy dwiema wartościami logicznymi pewnego wyrażenia. Warto zauważyć, że odpowiednio korzystając z warunków przejść można pominąć ten pseudostan, jednak często jego użycie zwiększa czytelność modelu



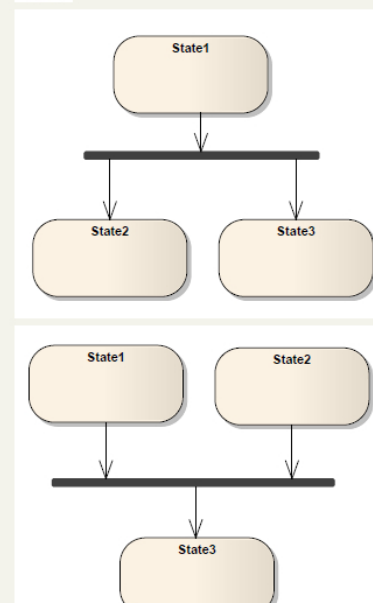
History

Płytkie wznowienie (shallow history). Wskazuje na ostatni aktywny stan obiektu przed wznowieniem. Nie przechowuje informacji o podstanach tego stanu. W przypadku pierwszej aktywacji pseudostanu wznowienia domyślnie aktywowany jest stan bezpośrednio powiązany z nim za pośrednictwem przejścia.



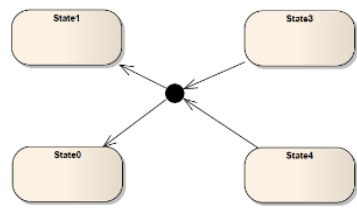
History

Głębokie wznowienie (deep history). Odpowiada ostatniej konfiguracji aktywnej stanu złożonego. Wskazuje na ostatni aktywny stan obiektu przed wznowieniem, przechowując zarazem informacje o podstanach tego stanu.



Rozwidlenie (fork). Pozwala na rozdzielenie wyjściowego przejścia na dwa lub więcej przejść wyjściowych do różnych obszarów współbieżnych, stanów lub maszyn stanowych.

Scalenie (join). Integruje kilka przejść wychodzących z różnych źródłowych obszarów współbieżnych, stanów lub maszyn stanowych.



Węzeł (junction). Pozwala na złączenie wielu przejść wejściowych oraz ich rozdzielanie na szereg przejść wyjściowych. Jest on tzw. statycznym rozgałęzieniem warunkowym, co oznacza, że realizacja przejść wejściowych nie determinuje natychmiastowej realizacji przejść wyjściowych. Szczególnymi przypadkami punktu węzłowego są decyzja oraz złączenie.



Punkt zniszczenia (terminate). Zakończenie maszyny stanowej wskutek zniszczenia adekwatnego obiektu.

- Na diagramach maszyny stanowej mogą wystąpić rozbudowane ścieżki przejść. Diagramy te uwzględniają zatem wiele możliwości szczegółowego opisu przejść, relacji między przejściami, jak również wpływu przejść na powiązane z nimi stany. Przykładowo przejścia mogą być łączone i rozwidlane. Pełne oddanie merytorycznej zawartości tych złożonych przejść dokonywane jest z wykorzystaniem pseudostanów (ang. pseudostates).
- Pseudostan jest abstrakcyjną kategorią modelowania diagramu maszyny stanowej, umożliwiającą organizowanie złożonych ścieżek przejść.
- Opisywana kategoria pojęciowa zdecydowanie wykracza poza podstawowe rozumienie przejścia, a jednocześnie nie spełnia roli stanu w rozumieniu maszyn stanowych.
- Pseudostany, których aktywacja uaktywnia stan poprzednio aktywny (w ramach stanu złożonego)

ZAAWANSOWANE SKŁADNIKI DIAGRAMU

Do kategorii zaawansowanych maszyny stanowej zaliczają się:

- sekcje symbolu graficznego stanu,
- klasyfikacja stanów,
- obszary współbieżne,
- pseudostany,
- rodzaje przejść,
- protokołowe maszyny stanowe,
- maszyny stanowe zachowania,
- zdarzenia.

SEKCJE SYMBOLU GRAFICZNEGO STANU

W dowolnym stanie zaprezentowanym na diagramie maszyny stanowej można wyróżnić następujące sekcje (ang. compartments):

- sekcję nazwy,
- sekcję czynności wewnętrznych,
- sekcję przejść wewnętrznych,
- sekcję dekompozycji - w przypadku stanów złożonych.

ETAPY TWORZENIA DIAGRAMU MASZINY STANÓW

Proces tworzenia diagramów maszyny stanowej przebiega w następujących etapach:

1. określenie rodzaju tworzonej maszyny stanowej i identyfikacja obiektów;
2. zidentyfikowanie poszczególnych stanów maszyny stanowej;
3. określenie hierarchii stanów, podstanów oraz obszarów współbieżnych;
4. powiązanie stanów oraz ich podstanów przejściami;
5. zastosowanie adekwatnych pseudostanów;
6. opracowanie specyfikacji sekcji stanów i przejść zgodnie z przyjętymi składnikami.

PRZYKŁADY DIAGRAMÓW



Zobacz przykład



Zobacz przykład

PRZYKŁAD DIAGRAMÓW MASZINY STANOWEJ - AUKCJA INTERNETOWA



Diagram maszyny stanowej dla klasy Aukcja



Diagram maszyny stanowej dla klasy Faktura