

Laboratorium 6

Diagram stanów

Cel laboratorium

W oparciu o zdefiniowane elementy z poprzednich laboratorium należy zaproponować diagramy stanów dla projektowanego systemu.

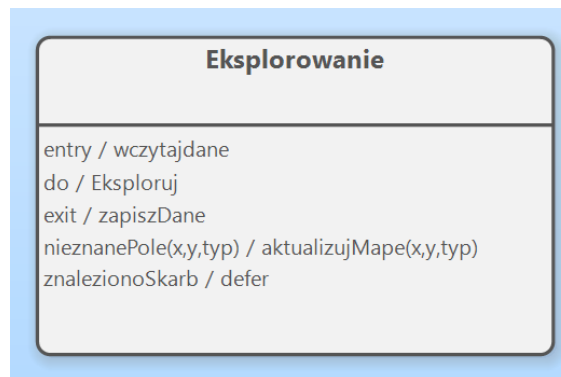
Diagramy stanów opisują wszystkie możliwe stany, do których może przejść dany obiekt, a także to, jak zmienia się stan obiektu pod wpływem zdarzeń do niego docierających. W większości technik obiektowych diagramy stanu są rysowane dla pojedynczych klas, aby pokazać cały cykl życia pojedynczego obiektu. Diagramy stanów są używane do:

- Modelowania zachowania obiektów w systemie,
- Analizy cykli życia złożonych obiektów,
- Projektowania systemów sterujących, np. maszyn stanów,
- Opisywania interfejsów użytkownika, w których stany zmieniają się w zależności od akcji.

Podstawowe elementy diagramu stanów

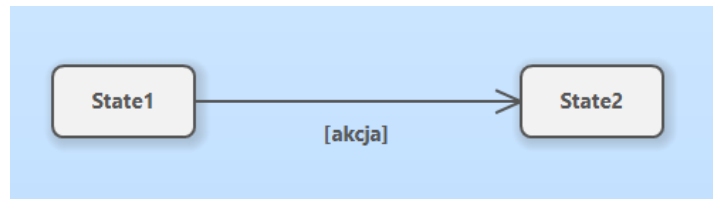
Stan (State) – oznacza sytuację, w której obiekt się znajduje w danym momencie. Stan najprościej mówiąc, to jakakolwiek okoliczność lub sytuacja, w jakiej znajduje się obiekt w czasie swojego życia, gdy spełnia warunek, wykonuje czynność lub czeka na zdarzenie. Stan prezentowany jest podobnie do czynności z diagramu aktywności – uwaga na pomyłki! W diagramie stanów nie przedstawia się czynności czy akcji wykonywanych przez obiekt, tylko same stany, które on przyjmuje w reakcji na wykonane czynności. Wyróżniamy dwa “stany” Stanu. Może on być:

- aktywny –gdy przechodzimy do niego przy użyciu przejścia;
- nieaktywny –gdy wychodzimy z niego



Rysunek 1 Stan

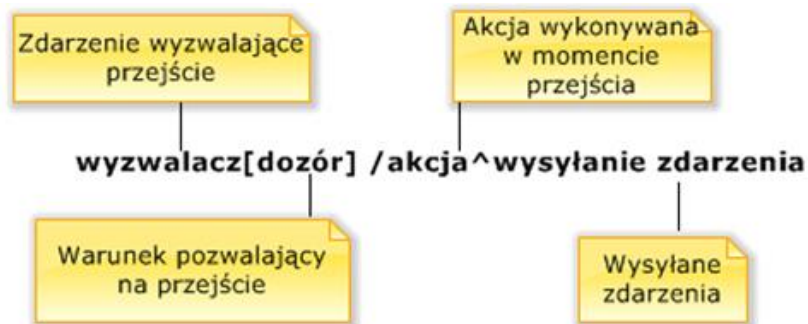
Przejście (Transition) – strzałka wskazująca zmianę stanu obiektu wywołaną zdarzeniem lub warunkiem. Przejście to związek między dwoma stanami. Wskazuje, że obiekt znajdujący się w pierwszym stanie wykona akcje i przejdzie do drugiego stanu ilekroć znajdzie określone zdarzenie i po spełnieniu określonych warunków. Przejście jest niepodzielne, tzn. nie można go przerwać.



Rysunek 2 Przejście

Akcja (Action) – operacja lub działanie wykonywane podczas zmiany stanu.

Zdarzenie (Event) – czynnik, który wywołuje przejście ze stanu do stanu.



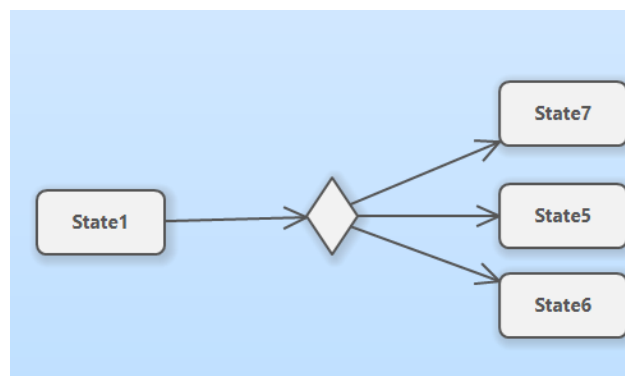
Rysunek 3 Definicja akcji i zdarzenia

Stan początkowy – stan rozpoczynający funkcjonowanie maszyny stanowej. To pseudostan– używany tylko do zapewnienia połączeń między stanami (w tym wypadku –dla rozpoczęcia diagramu), nie jest akcją. Nie może zawierać zdarzenia uruchamiającego.

Stan końcowy – prezentuje zakończenie maszyny stanowej. Ze stanu końcowego nie ma przejścia do stanu następnego.

Wewnętrzna aktywność stanu – działania, które mogą być wykonywane w danym stanie

Węzeł wyboru (ang. choice) - umożliwia dokonanie wyboru spośród kilku możliwości. Należy pilnować, by tylko jedna i zawsze jedna ścieżka była dostępna po spełnieniu warunku (warunek else).

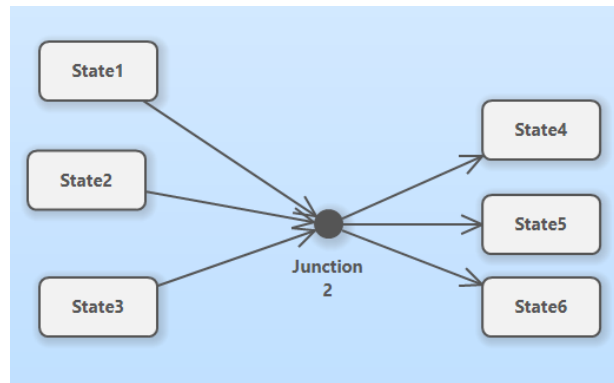


Rysunek 4 Węzeł wyboru

Węzeł skrzyżowania (ang. junction)

Stosowany do łączenia jednej lub wielu tranzycji wejściowych i ich rozdzielanie na jedną lub szereg tranzycji wyjściowych. Realizuje statyczne rozgałęzienie warunkowe. Różnica między junction a

węzłem wyboru: Należy pamiętać, że junction realizuje statyczne rozgałęzienia warunkowe, w przeciwieństwie do węzła wyboru, które realizuje dynamiczne rozgałęzienie warunkowe.



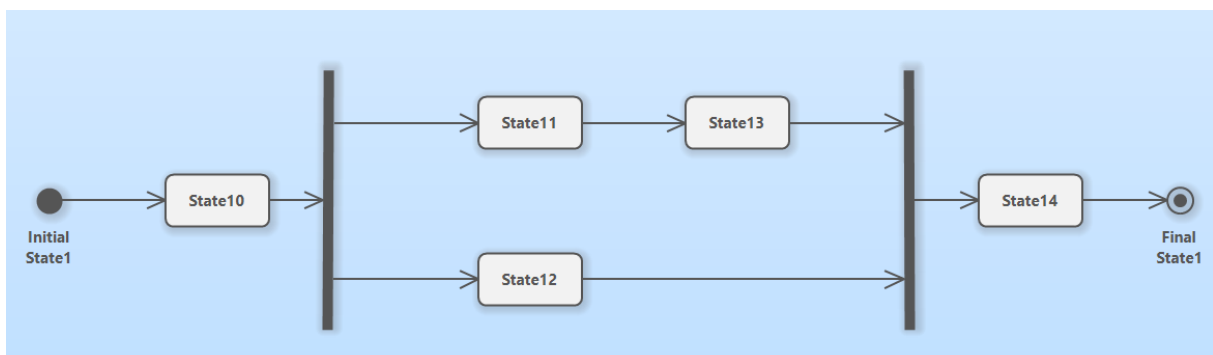
Rysunek 5 Węzeł skrzyżowania

Węzeł zniszczenia (ang. terminate) - znak X prezentowany na końcu tranzycji oznacza, że linia życia danej maszyny stanu została zakończona. Maszyna stanowa nie zmienia stanu ani nie realizuje żadnej akcji. Utworzenie terminate jest równoznaczne z wywołaniem akcji niszczenia obiektu.

Płytkie wznowienie (ang. shallowhistory) - służy do zapamiętania ostatniego aktywnego stanu maszyny stanowej, w którym została przerwana. Płytkie wznowienie charakteryzuje się tym, że nie przechowuje informacji o podstanach ostatniego aktywnego stanu maszyny stanowej. Prezentowany przez H w kółeczku.

Głębokie wznowienie (ang. deephistory) - głębokie wznowienie od płytkiego różni się tym, że głębokie wznowienie zapamiętuje ostatnią konfigurację aktywnego stanu złożonego. Czyli przerwanie nie spowoduje utraty wszystkiego, co w danym momencie było realizowane, tylko wskaże ostatni aktywny stan obiektu przed wznowieniem, wraz z informacją o jego podstanach. Prezentowany jako H z gwiazdką w kółeczku.

Rozwidlenie i scalenie (ang. fork node, join node) - rozwidlenie(ang. fork node) umożliwia reprezentację współbieżności stanów. Wskazuje początek przepływów równoległych. Stosowany do podzielenia ścieżki na minimum dwie równoległe ścieżki, których przepływy realizowane są w tym samym czasie. Scalenie(ang. join node) umożliwia scalenie przepływów. Scalenie umożliwia zsynchronizowanie ścieżek.



Rysunek 6 Przykład użycia rozwidlenia i scalenia

Obszary współbieżne / stany złożone (ang. compositestate)

Inżynieria Oprogramowania – Laboratorium

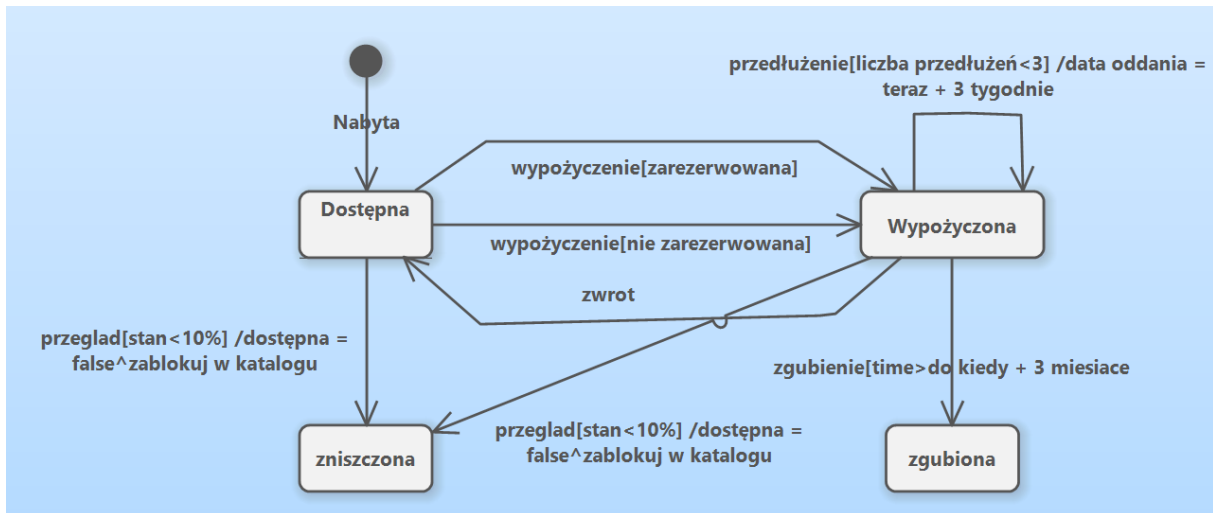
Stany w stanach złożonych aktywowane współbieżnie. Obszary współbieżne to wydzielane części stanu w których zamieszcza się równolegle aktywowane stany. Wszystkie obszary współbieżne muszą być wykonane aby stan mógł być zakończony.

Jak zbudować diagram stanów?

1. Określ stany obiektu – Wypisz wszystkie możliwe stany, w jakich może znajdować się obiekt.
2. Zidentyfikuj zdarzenia – Określ zdarzenia wywołujące przejścia między stanami.
3. Narysuj diagram:
 - Umieść stan początkowy na początku diagramu.
 - Dodaj wszystkie stany jako zaokrąglone prostokąty.
 - Połącz je strzałkami reprezentującymi przejścia i opisz zdarzenia, które je wywołują.
 - Umieść stan końcowy, jeśli istnieje

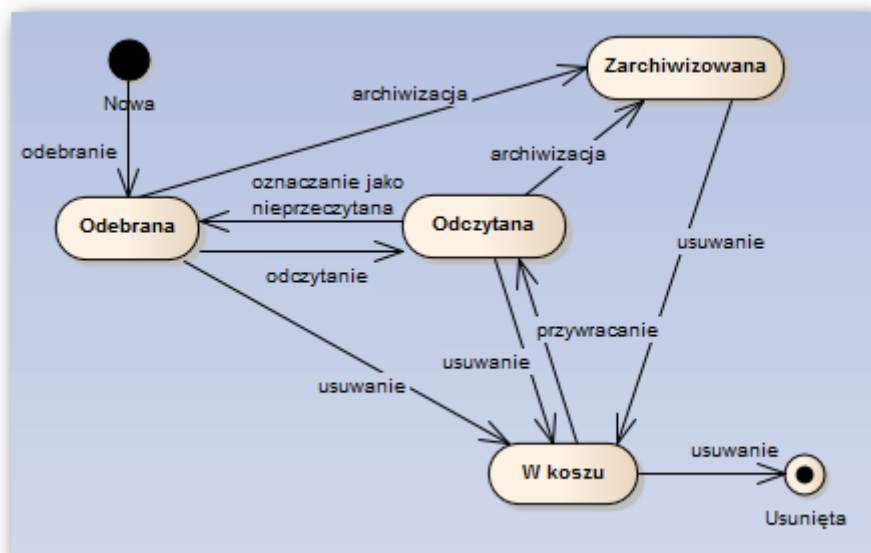
Przykład 1.

Diagram ten obejmuje przykładowy cykl życia obiektu Książka w bibliotece. Książka zostaje Nabyta, a następnie (po rejestracji) staje się Dostępna. Jej Wypożyczenie (niezależnie od tego, czy była rezerwowana, czy nie), przeprowadza ją do stanu Wypożyczona. Książka może pozostać w tym stanie np. wskutek przedłużenia (warunkiem jest, że liczba przedłużeń nie przekroczyła 3) – wówczas data oddania jest przesuwana o 3 tygodnie. Zarówno ze stanu Dostępności, jak i Wypożyczenia książka może przejść do stanu Zniszczenia, o ile wskutek zdarzenia Przegląd jej stan zostanie oceniony na mniej niż 10%. Jeżeli tak się stanie, atrybut Książki dostępna otrzymuje wartość false oraz wysyłane jest zdarzenie zablokowania dostępności Książki w katalogu. Wypożyczona książka, której nie oddano w terminie 3 miesięcy od terminu zwrotu, jest uważana za Zagubioną



Rysunek 7 Diagram stanów

Przykład 2.



Rysunek 8 Diagram stanów

Przykład 3.

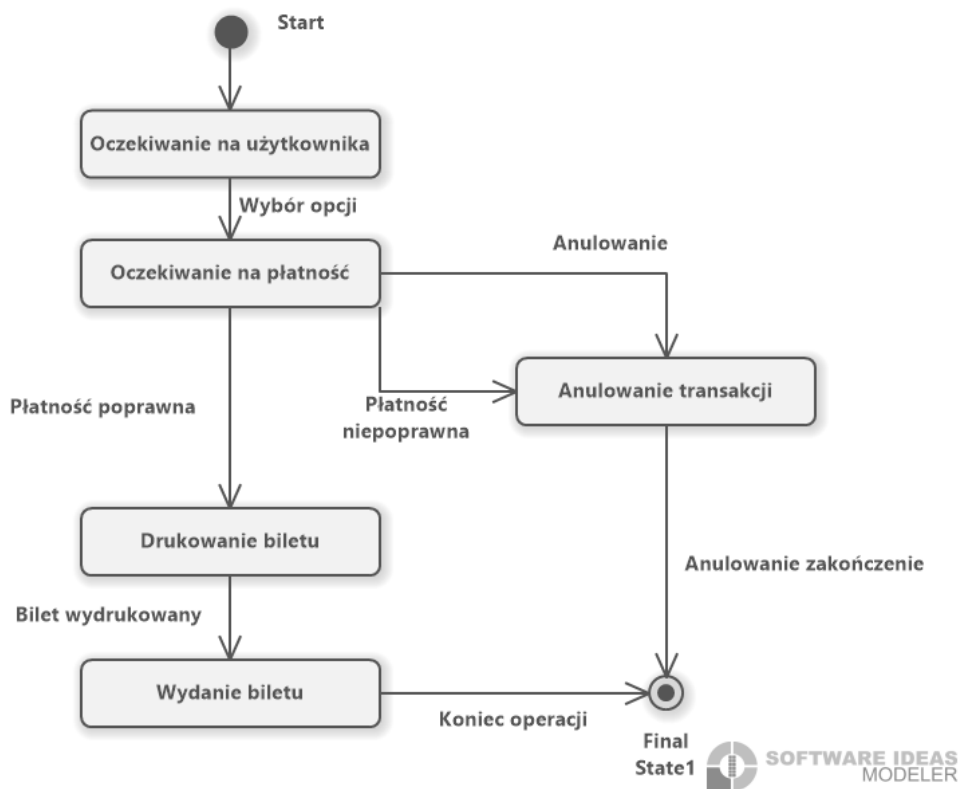
Założmy, że mamy automat biletowy jako przykład systemu, który działa na podstawie stanów. Automat biletowy przechodzi przez różne stany w zależności od działań użytkownika i warunków systemowych.

Scenariusz: Automat biletowy

Opis systemu

Automat biletowy:

1. Zaczyna w stanie oczekiwania na użytkownika.
2. Po wybraniu opcji użytkownika przechodzi do oczekiwania na płatność.
3. Po poprawnej płatności przechodzi do stanu drukowania biletu.
4. Jeśli płatność jest niepoprawna, wraca do stanu oczekiwania na płatność lub anuluje transakcję.
5. Po wydrukowaniu biletu kończy operację i wraca do stanu początkowego.



Rysunek 9 Diagram stanów

Opis poszczególnych stanów

1. Stan początkowy:
 - System jest gotowy do działania, ale czeka na interakcję ze strony użytkownika.
 - Reprezentowany jako czarne koło.
2. Oczekiwanie na użytkownika:
 - Stan, w którym automat czeka na wybór opcji przez użytkownika (np. wybór biletu, lokalizacji).

Inżynieria Oprogramowania – Laboratorium

- Przejście:
 - Zdarzenie: "Użytkownik wybiera bilet".
 - Przechodzi do Oczekiwania na płatność.
- 3. Oczekiwanie na płatność:
 - System czeka na wprowadzenie płatności przez użytkownika.
 - Możliwe przejścia:
 - Płatność poprawna: przejście do Drukowania biletu.
 - Płatność niepoprawna lub anulowanie: przejście do Anulowania transakcji.
- 4. Drukowanie biletu:
 - Automat przetwarza płatność i drukuje bilet.
 - Przejście:
 - Zdarzenie: "Bilet wydrukowany" → przejście do Wydania biletu.
- 5. Wydanie biletu:
 - Stan, w którym automat wydaje bilet użytkownikowi.
 - Przejście: "Bilet wydany" → przejście do Stanu końcowego.
- 6. Anulowanie transakcji:
 - Przejście w przypadku anulowania przez użytkownika lub błędu płatności.
 - Przejście: "Anulowanie zakończone" → przejście do Stanu końcowego.
- 7. Stan końcowy:
 - Koniec operacji, system wraca do gotowości. Reprezentowany jako czarne koło otoczone okręgiem.

Zdarzenia i akcje

Stan początkowy	Zdarzenie	Stan docelowy
Oczekiwanie na użytkownika	Użytkownik wybiera opcję	Oczekiwanie na płatność
Oczekiwanie na płatność	Płatność poprawna	Drukowanie biletu
Oczekiwanie na płatność	Płatność niepoprawna / anulowanie	Anulowanie transakcji
Drukowanie biletu	Bilet wydrukowany	Wydanie biletu
Wydanie biletu	Bilet wydany	Stan końcowy
Anulowanie transakcji	Anulowanie zakończone	Stan końcowy

Przedstawiony diagram pozwala:

- Przeanalizować, jakie stany może przyjmować system w różnych sytuacjach.

Inżynieria Oprogramowania – Laboratorium

- Zrozumieć interakcje użytkownika z automatem.
- Zidentyfikować potencjalne błędy (np. błędna płatność, anulowanie).
- Przygotować system do implementacji (przekształcić w maszyny stanów w kodzie).

Poniżej umieszczono zapis diagramu stanów w PlantUML, który przedstawia cykl życia automatu biletowego oraz otrzymaną reprezentację graficzną kodu.

```
@startuml
[*] --> Oczekiwanie_na_uzytkownika : Start

Oczekiwanie_na_uzytkownika --> Oczekiwanie_na_platnosc : Wybór opcji
Oczekiwanie_na_platnosc --> Drukowanie_biletu : Płatność poprawna
Oczekiwanie_na_platnosc --> Anulowanie_transakcji : Płatność niepoprawna
Oczekiwanie_na_platnosc --> Anulowanie_transakcji : Anulowanie

Drukowanie_biletu --> Wydanie_biletu : Bilet wydrukowany
Wydanie_biletu --> [*] : Koniec operacji
Anulowanie_transakcji --> [*] : Anulowanie zakończone

note right of Oczekiwanie_na_uzytkownika : Czeki na wybór użytkownika
note right of Oczekiwanie_na_platnosc : Czeki na płatność
note right of Drukowanie_biletu : Drukowanie biletu
note right of Wydanie_biletu : Wydanie biletu użytkownikowi
note right of Anulowanie_transakcji : Transakcja anulowana
@enduml
```



Rysunek 10 Diagram stanów