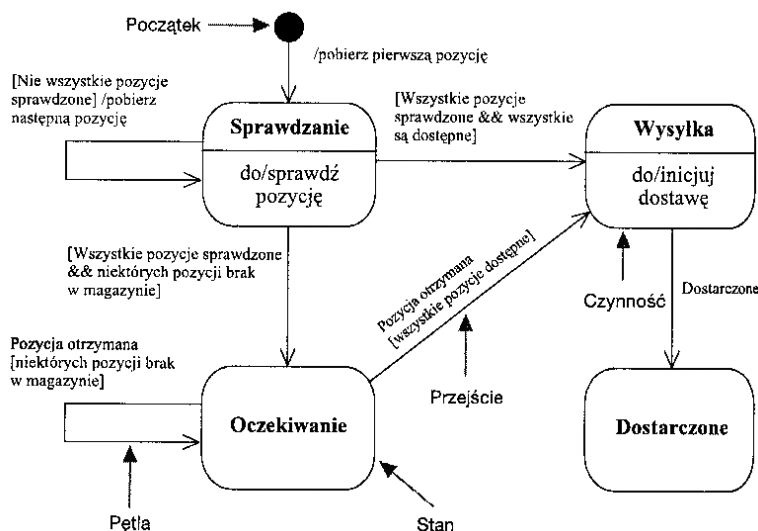


6 Diagramy stanów.

6.1 Wprowadzenie.

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. Rysunek 1 przedstawia diagram stanów UML-a dla zamówienia w systemie przetwarzania zamówień. Diagram pokazuje różne stany zamówienia.



Rysunek 1: Przykładowy diagram stanów.

Każdy diagram ma **początek** i **pierwsze przejście** - w tym wypadku do stanu *Sprawdzanie*. Przejście to jest oznaczone */pobierz pierwszą pozycję*. Syntaktyka etykiety przejścia składa się z trzech części, z których każda jest opcjonalna: *zdarzenie [dozór] / akcja*. W tym wypadku jest jedynie akcja *pobierz pierwszą pozycję*. Po wykonaniu tej akcji przechodzi się do stanu *Sprawdzanie*.

Z tym stanem jest związane wykonanie pewnej czynności, wskazanej za pomocą etykiety o syntaktyce *do/czynność*. W tym wypadku ta czynność nazywa się *sprawdź pozycję*. Termin **akcja** został tu użyty dla przejścia, zaś termin **czynność** dla stanu. Mimo że oba oznaczają procesy zwykle implementowane przez jakieś metody klasy *Zamówienia*, to są traktowane różnie. Akcje są związane z przejściami i traktuje się je jako procesy szybkie i nieprzerwywalne (atomowe). Czynności są związane ze stanami i mogą trwać dłużej. Czynność może zostać przerwana przez jakieś zdarzenie. Należy zwrócić uwagę, że znaczenie słowa szybko zależy od implementowanego systemu. W terminach systemu czasu rzeczywistego szybko może oznaczać "w ciągu kilku instrukcji maszynowych", natomiast dla zwykłego systemu informatycznego szybko może oznaczać "w ciągu kilku sekund".

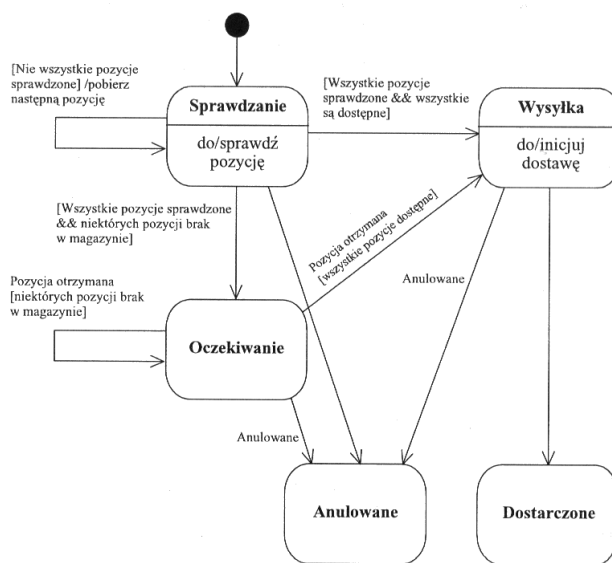
Jeśli z przejściem nie jest stowarzyszone zdarzenie, oznacza to, że dochodzi do niego natychmiast po zakończeniu czynności związanych ze stanem. W omawianym wypadku oznacza to, że do przejścia do następnego stanu dochodzi wówczas, gdy tylko zakończy się sprawdzanie pozycji. Ze stanu *Sprawdzanie* są możliwe trzy przejścia. Wszystkie mają tylko dozory w etykietach. **Dozór** jest to warunek logiczny zwracający jedną z dwu wartości - prawda lub fałsz. Do dozorowanego przejścia dochodzi tylko wtedy, gdy dozór zwraca wartość prawda.

Z danego stanu można wybrać **tylko jedno** przejście, tak więc dozory dla zdarzeń powinny wzajemnie się wykluczać. Na rysunku 1 są trzy warunki.

1. Jeśli nie zostały sprawdzone wszystkie pozycje, to należy pobrać następną pozycję i wrócić do stanu *Sprawdzanie*, aby ją sprawdzić.
2. Jeśli zostały sprawdzone wszystkie pozycje i wszystkie są dostępne w magazynie, to należy przejść do stanu *Wysyłka*.
3. Jeśli zostały sprawdzone wszystkie pozycje, ale nie wszystkie są dostępne w magazynie, należy przejść do stanu *Oczekiwanie*.

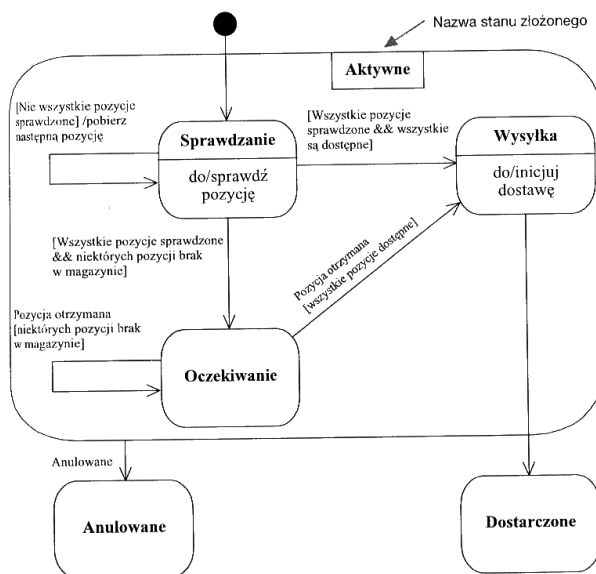
Warto przyjrzeć się najpierw stanowi *Oczekiwanie*. Nie wykonuje się w nim żadnych czynności, tak więc zamówienie czeka na zdarzenie. Oba przejścia z tego stanu są oznaczone zdarzeniem *Pozycja otrzymana*. Oznacza to, że zamówienie czeka na to zdarzenie. W chwili nadejścia tego zdarzenia są wyliczane wartości dozorów na przejściach i zamówienie przechodzi albo do stanu *Wysyłka*, albo z powrotem do *Oczekiwanie*. W stanie *Wysyłka* jest czynność, która inicjuje dostawę. Jest też jedno niedozorowane przejście wyzwalane przez zdarzenie *Dostarczone*. Oznacza to, że do przejścia dochodzi zawsze po tym zdarzeniu. Jednak do przejścia nie dochodzi po zakończeniu czynności; przeciwnie - po zakończeniu czynności inicjującej dostawę zamówienie pozostaje w stanie *Wysyłka*, aż do nadejścia zdarzenia *Dostarczone*.

Ostatnią rzeczą, którą należy się zająć, to przejście oznaczone etykietą *anulowane*. Aby mieć możliwość anulowania zamówienia w dowolnej chwili przed nadejściem dostawy, należy narysować oddzielne przejścia z każdego ze stanów *Sprawdzanie*, *Oczekiwanie*, *Wysyłka*. Innym sposobem jest utworzenie **stanu złożonego** składającego się ze wszystkich trzech stanów, a następnie narysować pojedyncze przejście z tego stanu. Podstawa po prostu dziedziczą przejścia stanu złożonego, obejmującego je. Rysunki 2 i 3 pokazują, w jaki sposób te dwa podejścia odzwierciedlają to samo zachowanie systemu.



Rysunek 2: Diagram stanów bez stanów złożonych.

Nawet przy tylko trzech powtórzonych przejściach rysunek 2 wygląda na dość zagracony. Rysunek 3 przedstawia jaśniejszy obraz i w wypadku późniejszych zmian pomaga nie zgubić zdarzeń anulowania.



Rysunek 3: Diagram stanów ze stanem złożonym.

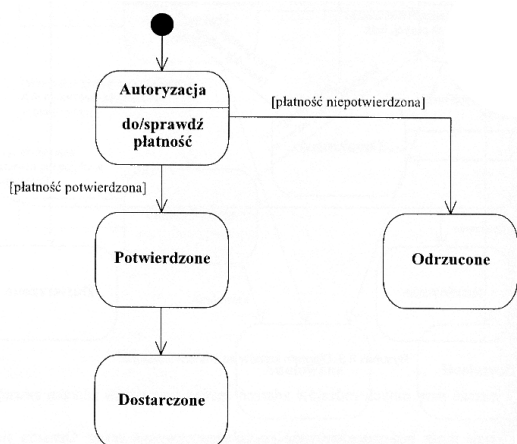
Oprócz czynności można jednak umieścić w stanie jeszcze parę innych rzeczy. Jeśli stan reaguje na pewne zdarzenie działaniem nie powodującym przejścia, to można to pokazać, umieszczając w symbolu stanu opis w postaci *nazwaZdarzenia/nazwaAkcji*.

Jest jeszcze parę innych rodzajów zdarzeń oprócz nazwanych zdarzeń zewnętrznych. Zdarzenie może być wygenerowane po upływie pewnego czasu. Oznacza się to słowem kluczowym **after** (z ang. po). Na przykład można napisać *after (20 minutes)*, [z ang. po (20 minutach)]. Zdarzenie może być wygenerowane po spełnieniu pewnego warunku. Oznacza się to słowem kluczowym **when** (z ang. gdy). Na przykład można napisać *when (temperature > 100 degrees)*, [z ang. gdy (temperatura > 100 stopni)].

Oprócz tego są dwa zdarzenia specjalne - **entry** (z ang. wejście) i **exit** (z ang. wyjście). Każda akcja stowarzyszona ze zdarzeniem entry jest wykonywana przy każdym wejściu do stanu. Akcja stowarzyszona ze zdarzeniem exit jest wykonywana przy każdym wyjściu ze stanu. Jeśli jest to przejście z akcją powracające do tego samego stanu (pętla), to wówczas najpierw będzie wykonywana akcja stowarzyszona z wyjściem ze stanu (tzn. ze zdarzeniem exit), następnie akcja związana z przejściem, a na końcu akcja związana z wejściem do stanu (tzn. ze zdarzeniem entry). Jeśli ze stanem jest stowarzyszona dodatkowo jeszcze jakaś czynność, to będzie ona wykonana po akcji związanej z wejściem do stanu.

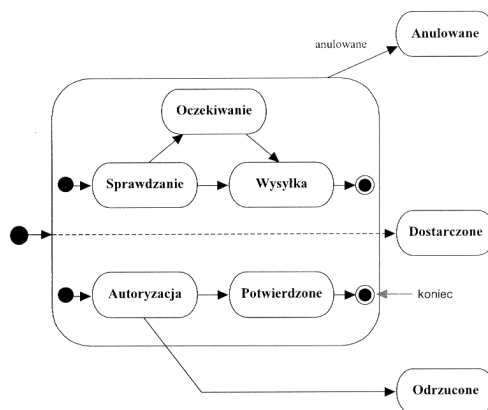
6.2 Diagramy stanów współbieżnych.

Oprócz stanów związanych z dostępnością pozycji zamówienia istnieją również stany związane z autoryzacją płatności. Przyglądając się tym stanom, można zobaczyć diagram podobny do tego z rysunku 4.



Rysunek 4: Potwierdzenie płatności - diagram stanów.

Należy zacząć od autoryzacji płatności. Czynność sprawdź płatność kończy się sygnałem potwierdzenia dokonania autoryzacji. Jeśli płatność powiodła się, zamówienie czeka w stanie Potwierdzone do chwili nadejścia zdarzenia dostawa. W przeciwnym wypadku zamówienie wchodzi w stan Odrzucone. Zachowanie Zamówienia jest kombinacją zachowań przedstawionych na rysunkach 1 i 4. Stowarzyszone z nim stany i stan Anulowane, omówiony wcześniej, mogą być połączone w diagramie stanów współbieżnych (zobacz rysunek 5).



Rysunek 5: Diagram stanów współbieżnych.

Współbieżne składniki stanu na diagramie są to miejsca, w których dane zamówienie może przebywać w dwu różnych stanach - każdym z innego składnika. W chwili opuszczania przez zamówienie stanów współbieżnych trafia ono do pojedynczego stanu. Widać, że zamówienie rozpoczyna swoją drogę jednocześnie od stanów Sprawdzenie i Autoryzacja. Jeśli czynność płatność czeka w stanie Autoryzacja zakończy się sukcesem, to zamówienie znajdzie się w stanach Sprawdzenie i Potwierdzone. Jeśli pojawi się zdarzenie anulowane, to zamówienie trafi tylko do stanu Anulowane.

Diagramy stanów współbieżnych przydają się w wypadkach, gdy dany obiekt ma pewne zbiory niezależnych zachowań. Jednak nie należy mieć zbyt wielu zbiorów współbieżnych zachowań dla pojedynczego obiektu. Jeśli dla jednego obiektu jest kilka skomplikowanych diagramów stanów współbieżnych, to należałoby się zastanowić nad rozbięciem tego obiektu na kilka prostszych.

6.3 Kiedy stosować diagramy stanów.

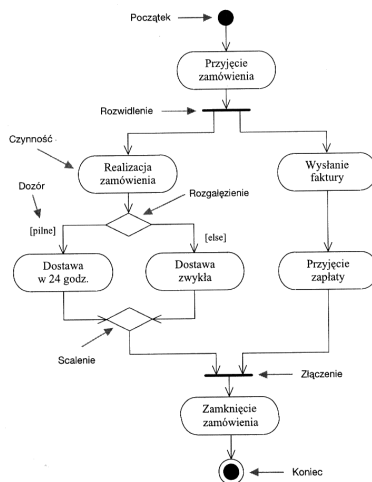
Diagramy stanów przydają się do opisywania zachowania obiektu obejmującego kilka przypadków użycia systemu. Diagramy stanów nie nadają się do opisu zachowań obejmujących współdziałanie wielu obiektów. W takich wypadkach lepiej połączyć technikę diagramów stanów z innymi technikami.

Na przykład diagramy interakcji są dobre do opisywania zachowania kilku obiektów w pojedynczym przypadku użycia systemu, a diagramy czynności przydają się w przedstawianiu ogólnego ciągu czynności dla kilku obiektów i przypadków użycia. Nie wszystkim diagramy stanów wydają się czymś naturalnym. Warto przyjrzeć się temu, jak inni je stosują. Może się zdarzyć, że zespół uzna, iż diagramy stanów nie są przydatne w pracy. Nie stanowi to wielkiego problemu - jak zwykle trzeba pamiętać o stosowaniu kilku różnych technik, które akceptują wszyscy. Stosując diagramy stanów, nie należy rysować ich dla każdej klasy systemu. Mimo że jest to czasami stosowane, to prawie zawsze jest stratą czasu. Diagramy stanów stosuje się tylko dla tych klas, które mają jakieś interesujące zachowanie, i tam, gdzie na rysowanie diagramu stanów pomoże zrozumieć, co się dzieje. Wiele osób uważa, że zachowanie obiektów interfejsu użytkownika i obiektów sterujących warto przedstawić na diagramach stanów.

7 Diagramy czynności.

7.1 Wprowadzenie.

Diagramy czynności to jeden z najbardziej nieoczekiwanych elementów UML-a. Diagramy te są szczególnie przydatne w modelowaniu przepływu zadań (ang. workflow) i w opisie procesów z dużą liczbą równoległych czynności. Na rysunku 6 podstawowym symbolem jest stan czynności lub po prostu czynność. Czynność jest to stan działania lub działalności: wykonywanie pewnego rzeczywistego procesu, na przykład pisanie listu, lub wykonywanie procedury programistycznej, na przykład wykonywanie metody klasy. Diagram czynności opisuje, jak są uszeregowane działania, oraz daje możliwość opisu czynności warunkowych lub współbieżnych. Diagram czynności jest wersją diagramu stanów, w którym większość stanów to stany czynności. Dlatego też większość terminologii pochodzi z diagramów stanów.

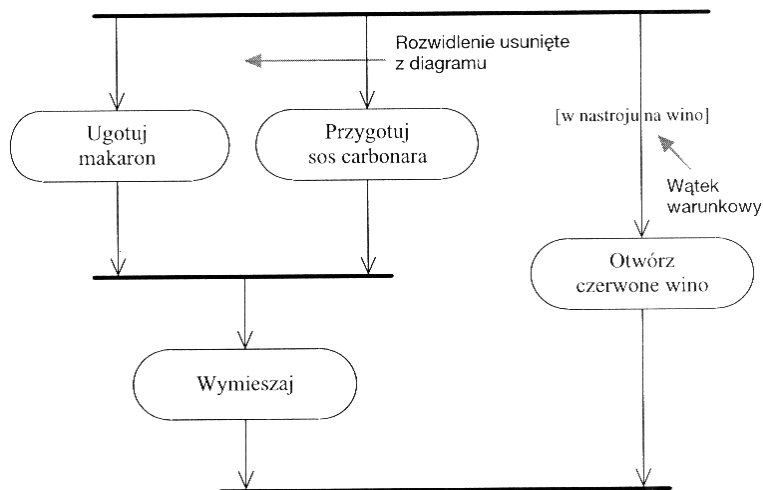


Rysunek 6: Diagram czynności

Proces warunkowy jest określony za pomocą rozgałęzienia i scalenia. Rozgałęzienie (decyzja) ma jedno wejście i kilka dozorowanych wyjść. Tylko jedno wyjście może być wybrane, tak więc dozory powinny się wzajemnie wykluczać. Użycie predefiniowanego dozoru [else] (z ang. w przeciwnym wypadku) jako dozoru oznacza, że jeśli wszystkie inne dozory są fałszywe, to należy wybrać wyjście odpowiadające dozorowi [else].

Na rysunku 6 po realizacji zamówienia jest *rozgałęzienie*. Jeśli zamówienie jest pilne, to dostawa jest realizowana w ciągu 24 godzin; w przeciwnym wypadku jest dostawą zwykłą. *Scalenie* ma wiele wejść i jedno wyjście. Scalenie oznacza koniec procesów warunkowych rozpoczętych rozgałęzieniem. Nie trzeba rysować rombów, aby oznaczyć rozgałęzienia i scalenia. Stan czynności, jak każdy inny stan, może mieć wiele dozorowanych wejść i wyjść. Romby można stosować, gdy rozgałęzienia i scalenia na diagramach mają być wyraźnie widoczne.

Procesy współbieżne są określane za pomocą *rozwidleń* i *złączeń*. Rozwidlenie ma jedno wejście i kilka wyjść. Kiedy jest uruchamiane wejście, to są wybierane równoległe wszystkie wyjścia. Zatem na rysunku 6, po przyjęciu zamówienia, jednocześnie jest realizowane zamówienie i wysyłana faktura. Diagram mówi, że te czynności mogą zachodzić współbieżnie. W istocie oznacza to, że ich kolejność jest nieważna. Można najpierw wykonać zamówienie, wysłać fakturę, dokonać dostawy i otrzymać zapłatę; albo wysłać fakturę, otrzymać zapłatę, a następnie wykonać zamówienie i dokonać dostawy. Można też wykonywać te czynności, przeplatając jedną z drugą - wziąć pierwszą pozycję zamówienia z magazynu, wypisać fakturę, wziąć drugą pozycję, włożyć fakturę do koperty itd. Można też wykonywać niektóre te czynności jednocześnie - pisać fakturę i sięgać do magazynu. Według diagramu każdy z tych sposobów jest poprawny.



Rysunek 7: Rozwidlenia, złączenia i wątki warunkowe.

Diagram czynności umożliwia wybór kolejności rzeczy do zrobienia. Innymi słowy określa podstawowe reguły porządkujące wykonywanie czynności. To jest właśnie różnica między diagramem czynności a klasycznym schematem przepływu sterowania - schematy są zwykle ograniczone do procesów sekwencyjnych, podczas gdy diagramy czynności potrafią ująć dodatkowo procesy współbieżne.

Jest to ważne przy modelowaniu procesów biznesowych. Przedsiębiorstwa często niepotrzebnie mają sekwencyjne procesy. Technika zachęcająca do stosowania współbieżności jest bardzo cenna, bo zwraca uwagę na możliwości wykonywania pewnych czynności równoległe. Może to poprawić wydajność i szybkość procesów biznesowych. Diagramy czynności przydają się także w programowaniu współbieżnym, ponieważ umożliwiają graficzne pokazanie wątków i ich potrzeb synchronizacyjnych. Zawsze przy procesach współbieżnych jest potrzebna synchronizacja. Nie można zamknąć zamówienia, dopóki nie doszło do dostawy i nie zapłacono za nią. Można zilustrować to złączeniem zaraz przed czynnością Zamknięcie zamówienia. W wypadku złączenia do wyjścia ze stanu dochodzi tylko wtedy, gdy na wejściu zakończono wszystkie działania. Rozwidlenia i złączenia muszą być sparowane. W najprostszym wypadku oznacza to, że za każdym razem, gdy jest rozwidlenie, to jest potrzebne złączenie łączące wszystkie wątki zapoczątkowane tym rozwidleniem. (Reguła ta wynika stąd, że diagram czynności jest rodzajem diagramu stanów.) Reguła ta ma jednak kilka rozszerzeń.

- Wątek pochodzący z rozwidlenia sam może się rozwidlić, a nowe wątki mogą się złączyć, zanim zostanie osiągnięty docelowy punkt złączenia.
- Jeśli wątek pochodzący z rozwidlenia prowadzi bezpośrednio do innego rozwidlenia, to drugie rozwidlenie

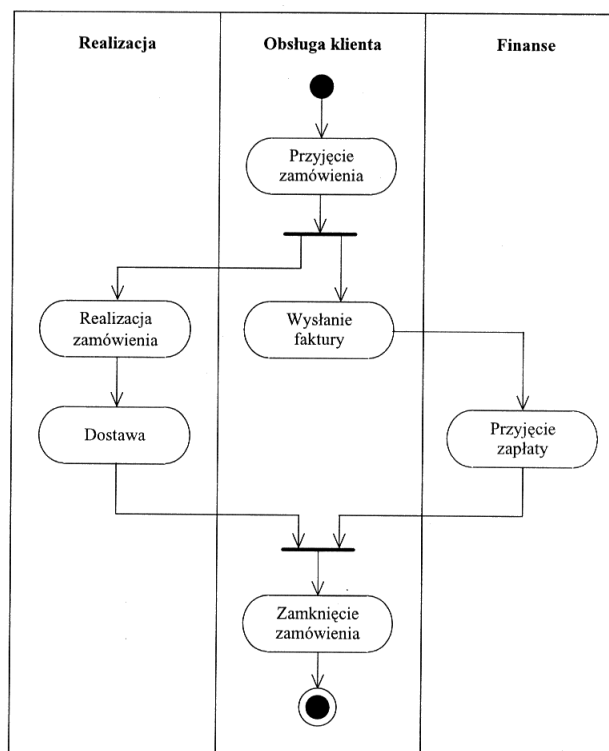
może być usunięte, a wątki z niego wyprowadzone mogą wychodzić z pierwotnego rozwidlenia. W ten sposób na rysunku 7 usunięto rozwidlenie między czynnościami przygotowywania jedzenia a pierwszym rozwidleniem. Analogicznie, jeśli z jednego złączenia dochodzi się bezpośrednio do innego, to można osunąć pierwsze złączenie i doprowadzić wątki do drugiego. Dzięki temu skrótowi notacyjnemu rysunek jest bardziej przejrzysty; semantycznie nic się nie zmienia.

- Bardziej złożona konstrukcja, nazywana stanem synchronizacyjnym (ang. synch state), umożliwia synchronizację w sytuacjach, w których reguła parowania rozwidleń i złączeń zwykle to uniemożliwia.

Jest jeden wyjątek od reguły mówiącej, że wszystkie stany wejściowe muszą zostać zakończone, zanim dojdzie do złączenia. Można dodać warunek do wątku wychodzącego z rozwidlenia. W efekcie otrzyma się wątek warunkowy. W trakcie wykonywania, jeśli warunek w wątku warunkowym jest fałszywy, zakłada się, że z punktu widzenia złączenia wątek ten jest zakończony. Na rysunku 7, nawet jeśli nie mam ochoty na wino, to w dalszym ciągu będę mógł zjeść spaghetti carbonara.

7.2 Tory.

Diagramy czynności mówią, co się dzieje, ale nie mówią, kto co robi. W wypadku programowania oznacza to, że diagram nie zawiera informacji, która klasa jest odpowiedzialna za daną czynność. W wypadku modelowania dziedziny oznacza to, że diagram nie zawiera informacji, kto jest odpowiedzialny za dane czynności. Jednym ze sposobów obejścia tego jest oznaczenie każdej czynności etykietą zawierającą klasę lub osobę odpowiedzialną za czynność. Jest to skuteczne, ale nie daje tej samej jasności, co diagramy interakcji w wypadku komunikacji między obiektami. Radą na to są **tory**.



Rysunek 8: Tory.

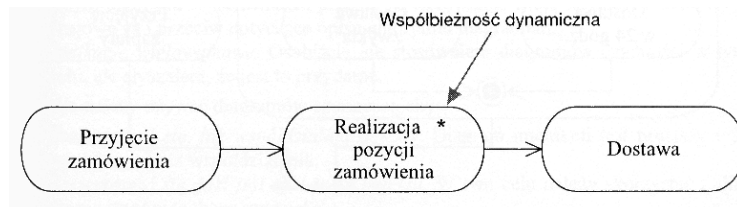
Aby skorzystać z torów, należy rozmieścić elementy diagramu w pionowych strefach przedzielonych liniami - torami. Każdy tor diagramu reprezentuje odpowiedzialność konkretnej klasy lub, tak jak to pokazano na rysunku 8, konkretnego wydziału. Zaletą torów jest to, że łączą one reprezentację logiki procesu z diagramów czynności z reprezentacją odpowiedzialności z diagramów interakcji. Mogą jednak być trudne do narysowania ze względu na złożoność niektórych diagramów. Czasami można stosować nie całkiem proste tory - lepsze to niż nic. Czasami po prostu nie udaje się umieścić wielu informacji na jednym diagramie.

Niektórzy, rysując diagramy czynności, starają się od razu przypisać czynności obiektom. Inni wolą najpierw narysować diagram czynności, aby mieć pojęcie o całym procesie, a dopiero później przypisują czynności obiektom. Niektórzy wolą przypisać zachowanie obiektom natychmiast. Należy samemu zdecydować, co jest

wygodniejsze. Najważniejszą rzeczą jest, aby na samym końcu mieć przypisane czynności klasom. Do tego lepiej jest używać diagramów interakcji.

7.3 Współbieżność dynamiczna.

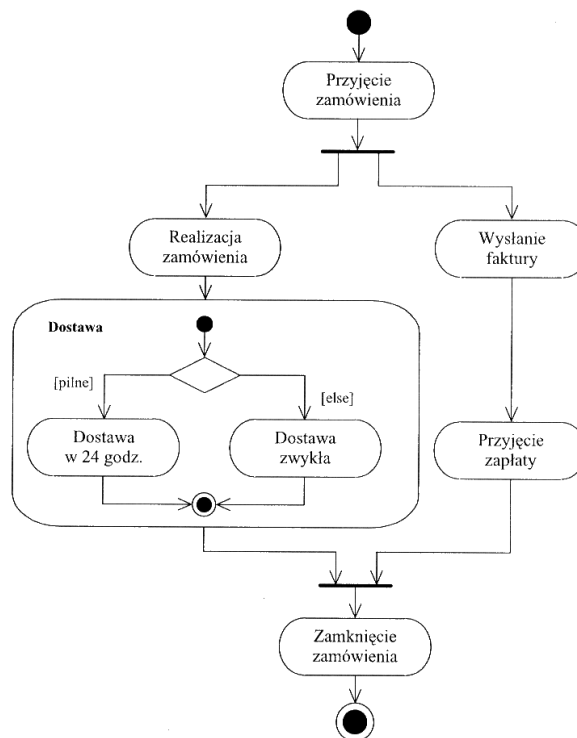
Współbieżność dynamiczna umożliwia zilustrowanie iteracji bez konieczności tworzenia pętli. Na rysunku 9 czynność *Realizacja pozycji zamówienia* jest wykonywana po jednym razie dla każdej *pozycji zamówienia*. Znacznik iteracji (*) oznacza, że czynność ta jest wykonywana wielokrotnie. Przejście do *Dostawy* jest uruchamiane tylko wtedy, gdy zostaną wykonane wszystkie pozycje zamówienia. Jeśli trzeba wykonać kilka czynności naraz, to można to oznaczyć, tworząc z *Realizacji pozycji zamówienia* czynność złożoną.



Rysunek 9: Współbieżność dynamiczna

7.4 Dekompozycja czynności.

Czynność może być rozbita na podczynności. Jest to zbliżone do pojęcia stanów złożonych i podstanów na diagramach stanów. Można pokazać stan złożony osobno na diagramie wyższego poziomu albo na diagramie czynności wraz z elementami wewnętrznymi, tak jak to zrobiłem na rysunku 10. W omawianym wypadku czynnościami związanymi z dostawą przypisałem początek i koniec. Można także narysować przejścia bezpośrednio do i z diagramu poziomu niższego. Zaletą jawnych początków i końców jest to, że czynności związane z dostawą mogą być używane w innych kontekstach, a diagram wyższego poziomu jest niezależny od zawartości diagramu poziomu niższego.



Rysunek 10: Dekompozycja czynności.

7.5 Kiedy stosować diagramy czynności.

Jak większość technik modelowania procesów diagramy czynności mają swoje słabe i mocne strony, dlatego najlepiej ich używać wraz z innymi technikami. Mocną stroną diagramów czynności jest to, że umożliwiają stosowanie procesów współbieżnych i zachęcają do tego. Czynniki to z nich idealne narzędzie modelowania przepływu zadań i programowania wielowątkowego. Ich bardzo słabą stroną jest to, że nie uwidaczniają wyraźnie związków między obiektami a czynnościami. Można określić powiązanie między czynnością a obiektem, etykietując czynność nazwą obiektu lub korzystając z torów dzielących diagram czynności w zależności od odpowiedzialności. Nie daje to jednak takiego wyniku, jaki można uzyskać na diagramach interakcji. Z tego powodu niektórzy uważają, że stosowanie diagramów czynności nie jest wystarczająco obiektowe, a co za tym idzie, jest złe. Uważam jednak, że ta technika jest bardzo przydatna i że nie należy odrzucać przydatnych narzędzi.

Diagramy czynności warto stosować w następujących sytuacjach:

- Analiza przypadku użycia. Na tym etapie projektant nie jest zainteresowany przypisaniem akcji obiektom - po prostu chce zrozumieć, jakie akcje muszą być wykonane i jakie są zależności procesowe. Metody przypisuje obiektom później i pokazuje te przypisania za pomocą diagramów interakcji.
- Zrozumienie przepływu zadań. Zanim nawet projektant zajmie się przypadkami użycia, stosuje diagramy czynności, aby zrozumieć procesy biznesowe. Później łatwo już narysować, pracując wraz z ekspertem w danej dziedzinie, jaka jest sekwencja czynności i jak ją można zmienić.
- Opisanie skomplikowanego algorytmu sekwencyjnego. W tym wypadku diagram czynności nie jest niczym więcej niż zgodnym z notacją UML-a diagramem opisującym algorytm - schematem przepływu sterowania. Mają tu zastosowanie zwyczajowe za i przeciw dotyczące opisu algorytmu diagramem.
- Aplikacje wielowątkowe. Osobiście nie stosowałem diagramów czynności w tym celu, ale słyszałem, że jest to przydatne.

Nie należy używać diagramów czynności, aby:

- Zorientować się, jak współdziałają obiekty. Diagram interakcji jest prostszy i da jaśniejszy obraz współdziałania.
- Zorientować się, jaki jest cykl życia obiektu. W tym celu należy skorzystać z diagramu stanów.
- Zilustrować złożone warunki. Należy zastosować tablicę decyzyjną.

Diagramy czynności uściślono i doprecyzowano w wyniku prac nad wersją 1.3 UML-a. To jednocześnie dobra i zła wiadomość. Problem polega na tym, że jeżeli te diagramy są stosowane w modelowaniu koncepcyjnym, to wiele z tej nowej precyzji po prostu przeszkadza. W takich wypadkach nie warto być stuprocentowo dokładnym, ponieważ wystarczy wiedzieć, co się dzieje. Niezależnie od nastawienia i wysiłków jest mało prawdopodobne, aby to, co jest robione w modelowaniu koncepcyjnym, było od razu poprawne, o ile oczywiście nie można wykonać i przetestować samych diagramów. Z drugiej strony w związku z tym, że istnieje standard dla diagramów stanów i diagramów czynności, twórcy narzędzi mają lepsze podstawy do tworzenia narzędzi do wykonywania tych diagramów. Narzędzia takie umożliwią wykonywanie i testowanie diagramów.