

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

UML 2.0. Wprowadzenie

Russ Miles, Kim Hamilton

Tłumaczenie: Rafał Szpoton

ISBN: 978-83-246-0632-0

Tytuł oryginału: [Learning UML 2.0](#)

Format: B5, stron: 280



Najtrudniejszym etapem każdego procesu tworzenia systemu informatycznego jest wykonanie odpowiedniego projektu. Umiejętność pogodzenia wymagań użytkowników i osób finansujących system z możliwościami oferowanymi przez technologię jest kluczowym elementem sukcesu. Im bardziej złożony system, tym bardziej zawiły staje się projekt. Konieczność ustandaryzowania technik projektowania systemów zaowocowała powstaniem narzędzi, dzięki którym nawet najbardziej skomplikowany projekt można przedstawić w prosty i czytelny sposób. Takim narzędziem jest notacja UML – zestaw ikon tworzących diagramy opisujące system i jego elementy.

Książka „UML 2.0. Wprowadzenie” w praktyczny sposób przedstawia techniki modelowania systemów informatycznych za pomocą języka UML 2.0. Czytając ją, nauczysz się graficznie przedstawiać otoczenie systemu, wymagania stawiane przez użytkowników i metody ich implementacji w systemie. Utworzysz diagramy klas, interakcji, komponentów, wdrożenia i inne, które opisują projekt w jednoznaczny oraz prosty sposób. Dowiesz się także, jak zaplanować proces wdrożenia produktu za pomocą UML.

- Elementy języka UML
- Modelowanie wymagań za pomocą przypadków użycia
- Diagramy czynności i sekwencji
- Modelowanie klas i powiązań pomiędzy nimi
- Diagramy komponentów
- Podział modelu na pakiety
- Modelowanie wdrożenia systemu

Poznaj nowoczesne metody projektowania systemów informatycznych

Wydawnictwo Helion
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl



Spis treści

Przedmowa	7
1. Wstęp	11
2. Modelowanie wymagań: przypadki użycia	29
Wychwytywanie wymagań systemowych	31
Zależności pomiędzy przypadkami użycia	39
Przeglądowe diagramy przypadków użycia	47
Co dalej?	49
3. Modelowanie przepływu czynności w systemie: diagramy aktywności	51
Podstawy diagramów aktywności	52
Czynności a akcje	54
Węzły decyzyjne oraz połączenia	55
Jednoczesne wykonywanie wielu zadań	58
Zdarzenia czasowe	59
Wywoływanie innych czynności	60
Obiekty	61
Nadawanie oraz odbieranie sygnałów	63
Rozpoczynanie czynności	65
Kończenie czynności oraz przepływów	65
Partycje (tory pływackie)	67
Zarządzanie złożonymi diagramami czynności	68
Co dalej?	70
4. Modelowanie struktury logicznej systemu: klasy oraz ich diagramy	71
Czym jest klasa?	71
Podstawy klas w języku UML	74
Widoczność	75
Stan klasy: atrybuty	79

Zachowanie klasy: operacje	83
Statyczne części klas	85
Co dalej?	88
5. Modelowanie struktury logicznej systemu: zaawansowane diagramy klas	89
Związki pomiędzy klasami	89
Ograniczenia	97
Klasy abstrakcyjne	98
Interfejsy	100
Szablony	103
Co dalej?	104
6. Powoływanie klas do istnienia: diagramy obiektów	107
Instancje obiektów	107
Połączenia	109
Wiązanie szablonów klas	111
Co dalej?	112
7. Modelowanie uporządkowanych interakcji: diagramy sekwencji	113
Uczestnicy na diagramie sekwencji	114
Czas	115
Zdarzenia, sygnały oraz komunikaty	116
Belki aktywacji	117
Komunikaty zagnieżdżone	118
Strzałki komunikatów	119
Ożywianie przypadku użycia za pomocą diagramu sekwencji	123
Zarządzanie złożonymi interakcjami za pomocą fragmentów sekwencji	129
Co dalej?	133
8. Połączenia opisujące interakcję: diagramy komunikacji	135
Uczestnicy, połączenia oraz komunikaty	135
Uzupełnianie interakcji za pomocą diagramu komunikacji	139
Diagramy komunikacji a diagramy sekwencji	142
Co dalej?	145
9. Harmonogramowanie interakcji: diagramy czasowe	147
Jak wyglądają diagramy czasowe?	147
Tworzenie diagramu czasowego na podstawie diagramu sekwencji	149
Umieszczanie uczestników na diagramie czasowym	150
Stany	150
Czas	151
Linia stanu uczestnika	153
Zdarzenia i komunikaty	155

Ograniczenia czasowe	156
Rozmieszczanie uczestników na diagramie czasowym	158
Notacja alternatywna	159
Co dalej?	163
10. Uzupełnianie obrazu interakcji: przeglądowe diagramy interakcji	165
Części przeglądowego diagramu interakcji	165
Modelowanie przypadku użycia za pomocą przeglądowego diagramu interakcji	167
Co dalej?	173
11. Modelowanie struktury wewnętrznej klasy: struktury złożone	175
Struktury wewnętrzne	176
Prezentacja sposobu użycia klasy	182
Prezentacja wzorców przy użyciu diagramów współpracy	183
Co dalej?	187
12. Zarządzanie częściami systemu oraz ich współużytkowanie: diagramy komponentów	189
Czym jest komponent?	189
Prosty komponent w języku UML	190
Udostępniane oraz wymagane interfejsy komponentu	191
Prezentacja współdziałania komponentów	193
Klasy realizujące komponent	194
Porty oraz struktura wewnętrzna	196
Widoki czarnej oraz białej skrzynki	199
Co dalej?	200
13. Porządkowanie modelu: pakiety	201
Pakiety	202
Przestrzenie nazw oraz klasy odwołujące się do siebie	204
Widoczność elementów	206
Zależności pomiędzy pakietami	206
Importowanie oraz używanie pakietów	207
Zarządzanie zależnościami pomiędzy pakietami	210
Stosowanie pakietów do porządkowania przypadków użycia	211
Co dalej?	212
14. Modelowanie stanu obiektów: diagramy maszyny stanowej	213
Podstawy	214
Stany	215
Przejścia	216
Stany programu	219
Zaawansowane zachowanie stanu	220

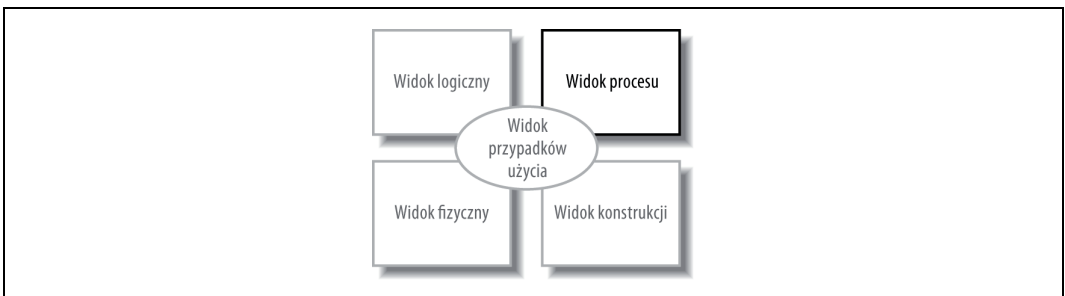
Stany złożone	222
Zaawansowane pseudostany	223
Sygnały	224
Maszyny stanowe protokołu	225
Co dalej?	225
15. Modelowanie wdrożenia systemu: diagramy wdrożenia	227
Wdrażanie prostego systemu	227
Wdrażanie oprogramowania: artefakty	229
Czym jest węzeł?	232
Węzły sprzętowe oraz środowiska uruchomieniowego	232
Komunikacja pomiędzy węzłami	234
Specyfikacje wdrożenia	235
Kiedy stosować diagram wdrożenia?	236
Co dalej?	237
A Język ograniczeń obiektowych	239
B Dostosowywanie UML-a: profile	247
C Historia UML-a	253
Skorowidz	259

Modelowanie przepływu czynności w systemie: diagramy czynności

Przypadki użycia pokazują, *co* powinien robić system. Diagramy czynności umożliwiają określenie tego, *w jaki sposób* system będzie osiągał swoje zamierzone cele. Diagramy czynności przedstawiają akcje zamodelowane na wysokim poziomie oraz połączone razem w łańcuch, reprezentujące procesy zachodzące w systemie. I tak na przykład diagram czynności może zostać użyty do zamodelowania czynności koniecznych do utworzenia konta pamiętnika internetowego.

Diagramy czynności są szczególnie przydatne w modelowaniu *procesów biznesowych*. Proces tego rodzaju jest zestawem skoordynowanych zadań, które trzeba wykonać, aby osiągnąć cel biznesowy, na przykład dostarczenie zamówień do klientów. Niektóre z narzędzi do zarządzania procesami biznesowymi (ang. *business process management*, w skrócie BPM — *przyp. tłum.*) umożliwiają zdefiniowanie procesów biznesowych przy użyciu diagramów czynności lub też podobnej notacji graficznej, a następnie ich wykonanie. Pozwala to na przykład zdefiniować oraz wykonać przy użyciu prostej notacji graficznej zawierającej diagramy czynności proces zatwierdzania płatności, którego jeden z etapów stanowić będzie wywołanie usługi sieciowej zatwierdzającej transakcje wykonane przy użyciu kart kredytowych.

Diagramy czynności są jedynym diagramem UML-a w widoku procesu modelowanego systemu, co wynika z rysunku 3.1.



Rysunek 3.1. Widok procesu przedstawia wysoko poziomowe procesy w systemie, do modelowania których bardzo dobrze nadają się diagramy czynności

Diagramy czynności są jednymi z najbardziej zrozumiałych diagramów UML-a, ponieważ używają symboli podobnych jak powszechnie stosowana notacja przepływu (ang. *flowchart* — *przyp. tłum.*). Dlatego też przydają się one do opisywania procesów dla szerszego audytorium.

W rzeczywistości diagramy czynności, podobnie jak diagramy stanów, pochodzą od diagramów przepływów.

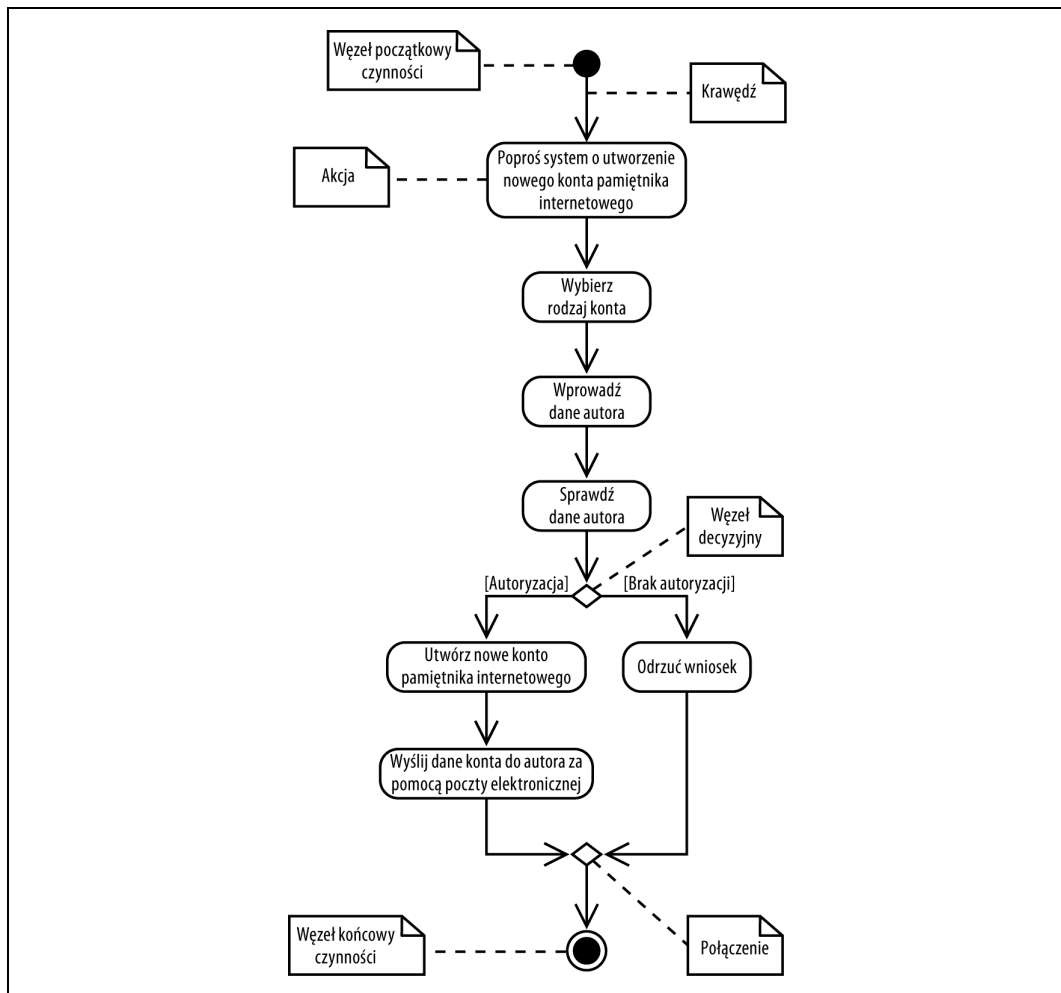
Podstawy diagramów czynności

Przyjrzyjmy się podstawowym elementom diagramów czynności, modelując przy okazji proces napotkany wcześniej w tej książce, to znaczy czynności zdefiniowane w przypadku użycia opisującym tworzenie konta pamiętnika internetowego. Tabela 3.1 zawiera opis przypadku użycia o nazwie *Utwórz nowe konto pamiętnika* (oryginalnie była to tabela 2.2). Proces tworzenia konta pamiętnika internetowego opisują sekcje „Główny przepływ wykonania” oraz „Rozszerzenia”.

Tabela 3.1. Opis przypadku użycia o nazwie *Utwórz nowe konto pamiętnika*

Nazwa przypadku użycia	Utwórz nowe konto pamiętnika	
Powiązane wymagania	Wymaganie A.1.	
Kontekst zadaniowy	Nowy lub już istniejący autor żąda od administratora utworzenia nowego konta pamiętnika internetowego.	
Warunki wstępne	System dostępny jest dla rozpoznanych autorów. W związku z tym autor musi dysponować odpowiednim potwierdzeniem tożsamości.	
Warunek pomyślnego zakończenia	Dla autora tworzone jest nowe konto pamiętnika.	
Warunek niepomyślnego zakończenia	Wniosek o nowe konto pamiętnika jest odrzucany.	
Aktorzy główni	Administrator.	
Aktorzy drugoplanowi	Baza danych z danymi autorów.	
Wyzwalacz	Administrator żąda od systemu CMS utworzenia nowego konta pamiętnika internetowego.	
Główny przepływ wykonania	Krok	Akcja
	1.	Administrator prosi system o utworzenie nowego konta pamiętnika.
	2.	Administrator wybiera rodzaj konta.
	3.	Administrator wprowadza szczegółowe dane autora.
	4.	Szczegółowe dane autora są weryfikowane przy użyciu informacji pobranych z bazy danych autorów.
	5.	Tworzone jest nowe konto pamiętnika.
	6.	Podsumowanie informacji o nowym koncie pamiętnika jest przesyłane do autora pocztą elektroniczną.
Rozszerzenia	Krok	Rozgałęziona akcja
	4.1.	Informacje uzyskane z bazy danych nie pozwalają na potwierdzenie danych autora.
	4.2.	Wniosek o utworzenie nowego konta pamiętnika jest odrzucany.

Na rysunku 3.2 przedstawiony został proces tworzenia konta pamiętnika internetowego zapisany przy użyciu notacji diagramu czynności. Diagram czynności przydaje się w tej sytuacji, ponieważ pozwala w lepszy sposób zobrazować akcje opisane w przypadku użycia (w porównaniu z notacją tablicy zastosowaną w opisie przypadku użycia), a szczególnie te rozgałęzione, które zależą od tego, czy dane autora zostaną zweryfikowane.



Rysunek 3.2. Diagramy czynności modelują dynamiczne zachowanie systemu, koncentrując się na procesach. Podstawowe elementy diagramów czynności przedstawione zostały na podstawie procesu tworzenia konta pamiętnika internetowego

Z rysunku 3.2 wynika, że wykonanie czynności rozpoczyna się w jej węźle początkowym (ang. *initial node* — przyp. tłum.) przedstawionym pod postacią wypełnionego koła. Węzeł początkowy oznacza najzwyczajniej początek czynności. Na drugim końcu diagramu występuje węzeł końcowy czynności (ang. *activity final node* — przyp. tłum.) oznaczający jej koniec i przedstawiony pod postacią dwóch koncentrycznych kół, z których środkowe jest wypełnione.

Pomiędzy węzłem początkowym a końcowym czynności występują akcje (ang. *actions*) obrazowane za pomocą prostokątów o zaokrąglonych narożnikach. Akcje są ważnymi krokami w danej czynności, np. Wybierz rodzaj konta, Wprowadź dane autora itd. Akcją może być wykonane działanie, obliczenie lub dowolny kluczowy krok procesu.

Przepływ czynności przedstawiony jest przy użyciu strzałek nazywanych krawędziami (ang. *edges*) lub ścieżkami (ang. *pathes*). Strzałka na końcu wskazuje kierunek przepływu od jednej akcji do kolejnej. Linia przychodząca do danego węzła nazywana jest krawędzią wchodzącą (ang. *incoming edge*), zaś wychodząca z niego nazywana jest krawędzią wychodzącą (ang. *outgoing edge*).

Krawędzie łączą akcje ze sobą, określając całkowity przepływ czynności. Na początku aktywny staje się węzeł początkowy, następnie ten o nazwie Poproś system o utworzenie nowego konta pamiętnika internetowego i tak dalej.

Pierwszy z węzłów zaprezentowanych w postaci rombu nazywany jest *decyzyjnym* (ang. *decision*) i odpowiada blokowi instrukcji typu if-else w kodzie programu. Należy zauważyć, że od węzła decyzyjnego na rysunku 3.2 odchodzą dwie krawędzie wychodzące, z których każda nazwana jest zgodnie z wynikiem warunku logicznego. W rzeczywistości wykonywana jest tylko jedna krawędź w zależności od tego, czy dane autora zostaną potwierdzone, czy nie. Drugi węzeł w postaci rombu nazywany jest *połączeniem* (ang. *merge*). Połączenie łączy ze sobą krawędzie wychodzące z węzła decyzyjnego, oznaczając w ten sposób koniec zachowania warunkowego.

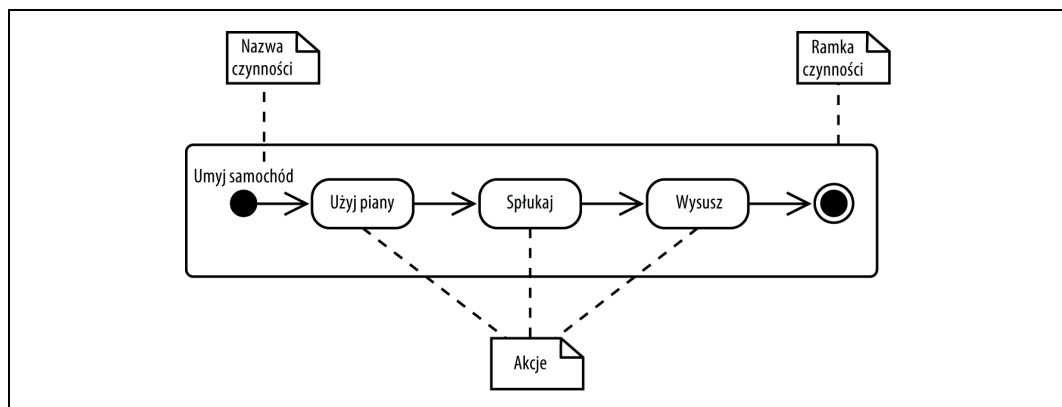
Słowo „przepływ” zostało już poprzednio użyte kilkakrotnie i czytelnik może sobie zadać pytanie, co w takim razie płynie. Odpowiedź zależy od kontekstu. Zazwyczaj jest to przepływ sterowania od jednej akcji do kolejnej. Kiedy jedna akcja kończy działanie, wtedy sterowanie jest przekazywane do drugiej. W kolejnych podrozdziałach przekonamy się jednak, że wraz ze sterowaniem w danej czynności mogą przepływać również obiekty.

Czynności a akcje

Akcje są aktywnymi krokami niezbędnymi do ukończenia procesu. Akcja może być obliczeniem, na przykład Oblicz podatek, lub zadaniem, jak na przykład Sprawdź dane autora.

Słowo „czynność” jest bardzo często błędnie używane zamiast wyrazu „akcja” w celu opisanego kroku na diagramie czynności. Nie są one jednak tożsame. *Czynność* jest modelowanym procesem, jak chociażby mycie samochodu. *Akcja* jest krokiem w danej czynności, jak na przykład Użycie piany, Spłukanie, Wysuszenie.

Akcje występujące w tej prostej czynności mycia samochodu przedstawione zostały na rysunku 3.3.

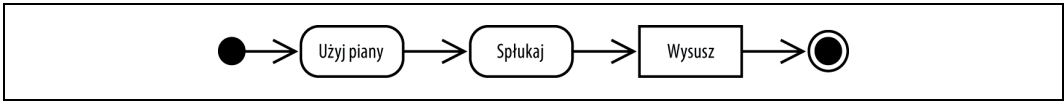


Rysunek 3.3. Trzy akcje: Użycie piany, Spłukanie oraz Wysuszenie tworzą na diagramie czynność o nazwie Umyj samochód

Na rysunku 3.3 cała czynność umieszczona jest wewnątrz zaokrąglonego prostokąta nazywanego *ramką czynności* (ang. *activity frame* — *przyp. tłum.*). Ramka czynności wykorzystywana jest do umieszczania w niej akcji czynności i przydaje się w sytuacji, gdy na jednym diagramie

chcemy przestawić więcej niż jedną czynność. Nazwa czynności powinna zostać umieszczona w lewym górnym rogu ramki.

Ramka czynności jest opcjonalna i często może być pomijana na diagramie czynności, tak jak ma to miejsce na alternatywnym diagramie Umyj samochód, zaprezentowanym na rysunku 3.4.

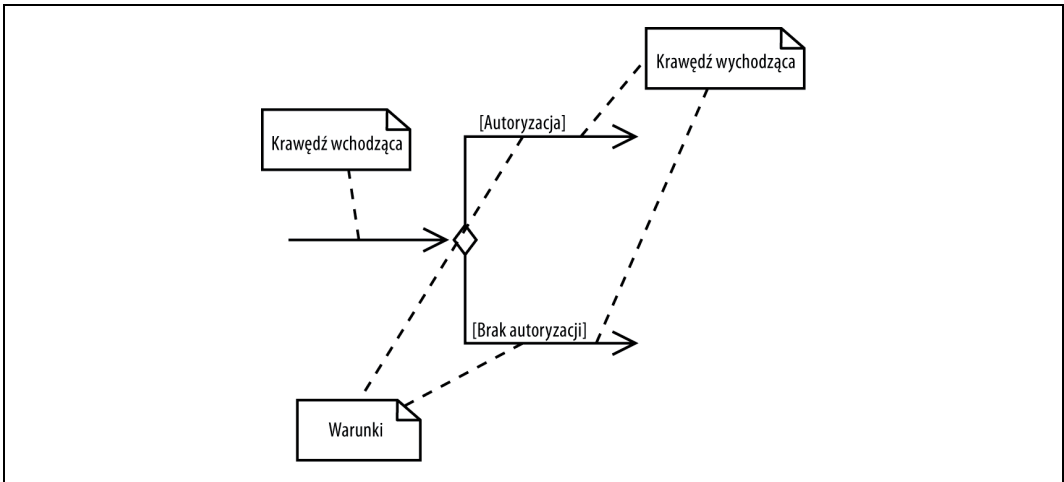


Rysunek 3.4. Ramka czynności może zostać pominięta

Chociaż w ten sposób tracimy nazwę czynności prezentowanej na diagramie, to jednak opuszczenie ramki jest często wygodne w przypadku tworzenia prostych diagramów.

Węzły decyzyjne oraz połączenia

Węzły decyzyjne (ang. *decisions* — *przyp. tłum.*) używane są w przypadku, gdy w zależności od warunku chcemy wykonać inną sekwencję akcji. Węzły tego rodzaju są przedstawiane w postaci rombu z jedną krawędzią wchodzącą oraz wieloma wychodzącymi, tak jak pokazane zostało to na rysunku 3.5.



Rysunek 3.5. Z węzła decyzyjnego sterowanie przekazywane jest tylko wzdłuż jednej krawędzi

Każda rozwidlona krawędź zawiera *warunek* (ang. *guard condition* — *przyp. tłum.*) zapisany w nawiasach kwadratowych. Warunki określają, która krawędź zostanie wybrana w węźle decyzyjnym.

Warunki przyjmują wartości logiczne prawda lub fałsz, na przykład:

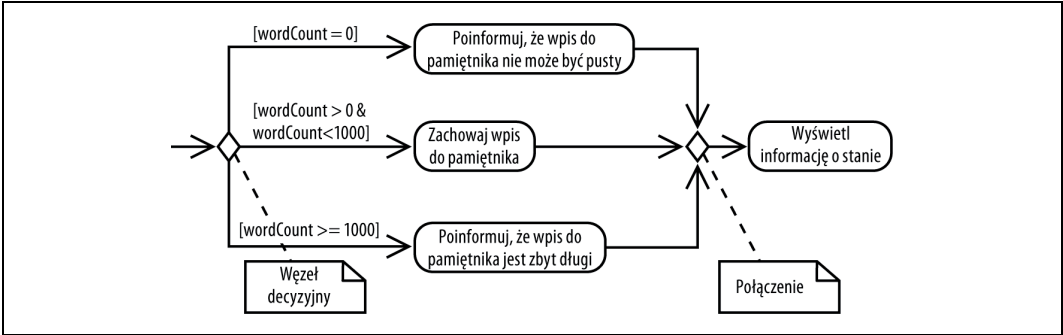
[Autoryzacja]

Jeżeli zmienna `authorized` ma wartość `true`, podążaj zgodnie z krawędzią wychodzącą umieszczoną przy warunku.

[`wordCount >= 100`]

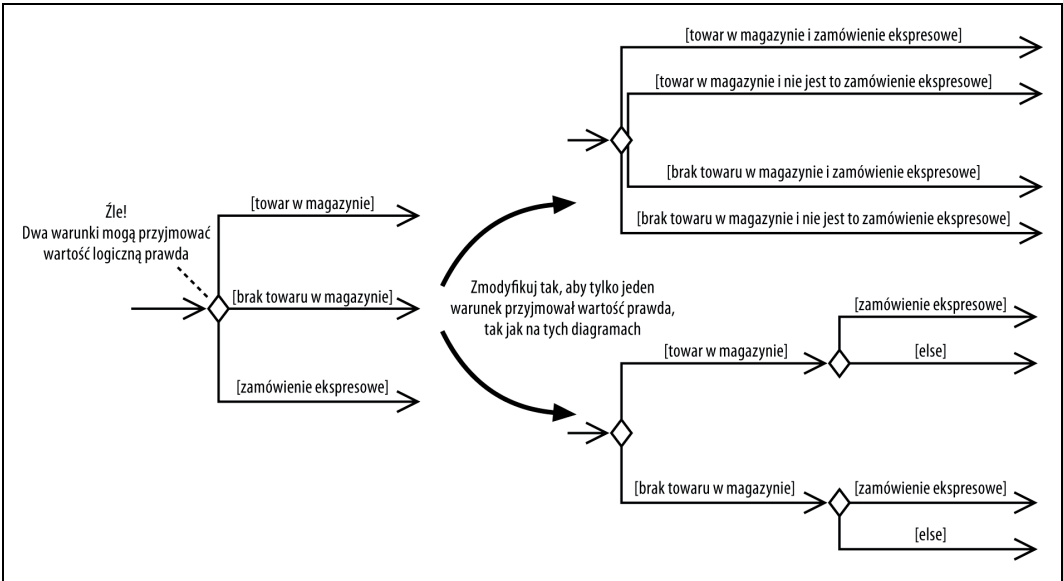
Jeżeli zmienna `wordCount` ma wartość większą lub równą 1000, wtedy podążaj zgodnie z krawędzią wychodzącą umieszczoną przy warunku.

Rozwidlone przepływy łączą się ponownie w węźle połączenia (ang. *merge node* — przyp. tłum.) sygnalizującym koniec warunkowego zachowania rozpoczętego wcześniej w węźle decyzyjnym. Połączenia są również przedstawiane przy użyciu rombu, ale mają wiele krawędzi wchodzących i tylko jedną wychodzącą, co widać na rysunku 3.6.



Rysunek 3.6. Jeżeli liczba słów wynosi przykładowo 1200, wtedy wykonywana jest akcja o nazwie *Poinformuj, że wpis do pamiętnika jest zbyt długi*

Diagramy czynności są najbardziej przejrzyste, jeżeli warunki w węzłach decyzyjnych są kompletne oraz wzajemnie rozłączne. Sytuacja, w której ścieżki nie są wzajemnie rozłączne, przedstawiona została na rysunku 3.7.

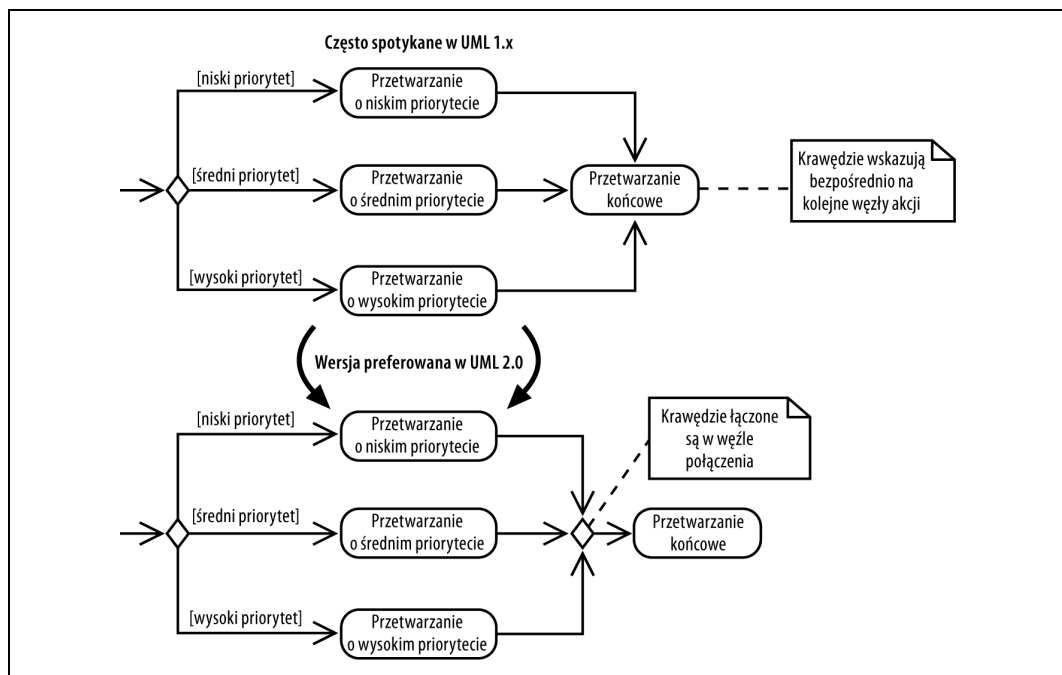


Rysunek 3.7. Uwaga na diagramy z wieloma warunkami przyjmującymi wartość *prawda*

Jeżeli towar jest w magazynie i zamówienie jest ekspresowe, wtedy dwa warunki mają wartość logiczną *prawda*. Która więc krawędź powinna zostać wybrana? Według specyfikacji UML-a w przypadku, gdy wiele różnych warunków przyjmuje wartość *prawda*, wybierana jest tylko jedna krawędź, a jej wybór jest przypadkowy, chyba że określi się uporządkowanie krawędzi. Wystąpieniu tej skomplikowanej sytuacji można zapobiec, tworząc warunki wzajemnie rozłączne.

Inną sytuacją, której należy zapobiec, są warunki niekompletne. Na przykład gdyby na rysunku 3.7 nie było warunku określającego przypadek braku towaru w magazynie, wtedy w razie zaistnienia takiej sytuacji nie można by było wybrać żadnej krawędzi węzła decyzyjnego. Oznacza to, że dana czynność zostanie zamrożona w węźle decyzyjnym. Osoby modelujące system niekiedy pozbywają się warunków, jeżeli dana sytuacja nie występuje (lub też jeżeli chcą się nad nią zastanowić później). Jednakże aby zminimalizować ryzyko nieporozumienia, powinno się zawsze umieszczać warunki pokrywające wszystkie możliwe sytuacje. Jeżeli jest to możliwe w danej czynności, wtedy przydatne bywa nadanie jednej ze ścieżek etykiety *else*¹ (jak pokazano na rysunku 3.7), co pozwoli upewnić się, że wszystkie sytuacje są obsługiwane.

Jeżeli czytelnik ma doświadczenie w pracy z językiem UML w wersji 1.x, może uważać, że przedstawianie węzłów połączeń nie jest konieczne. W językach z rodziny UML 1.x bardzo często można było ujrzyć wiele krawędzi rozpoczynających się w węźle decyzyjnym i biegnących bezpośrednio do akcji, tak jak ma to miejsce na rysunku 3.8. Oznaczało to, że przepływy były łączone niejawnie.



Rysunek 3.8. W UML 2.0 lepiej jest stosować jak najbardziej przejrzysty zapis i jawnie zapisywać węzły połączenia

Począwszy od wersji UML 2.0, w przypadku gdy wiele krawędzi prowadzi bezpośrednio do akcji, kontynuacja wszystkich wchodzących przepływów jest wstrzymywana. Można uniknąć nieporozumień, jawnie przedstawiając węzły połączeń.

¹ Krawędź oznaczona etykietą *else*, tak jak w przypadku konstrukcji bloku warunkowego if-else występującego w wielu językach programowania, wybierana jest w przypadku, gdy żaden z warunków nie przyjmuje wartości logicznej prawda. Tak więc krawędź ta obsługuje domyślnie wszystkie sytuacje nieprzewidziane na diagramie — *przyp. tłum.*

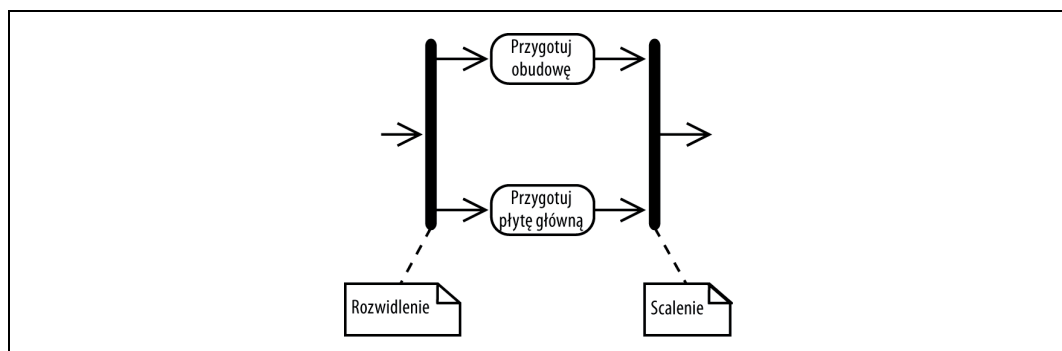
Jednoczesne wykonywanie wielu zadań

Rozważmy przepływ czynności związanych z montażem komputerów i obejmujących następujące kroki:

1. Przygotuj obudowę.
2. Przygotuj płytę główną.
3. Zainstaluj płytę główną.
4. Zainstaluj napędy.
5. Zainstaluj kartę graficzną, dźwiękową oraz modem.

Jak dotąd opisaliśmy już wystarczająco dużo notacji diagramu czynności, aby możliwe było zamodelowanie tego sekwencyjnego przepływu zdarzeń. Załóżmy jednak, że cały przepływ mógłby zostać przyspieszony poprzez jednoczesne przygotowanie obudowy oraz płyty głównej, ponieważ obie te akcje nie zależą od siebie wzajemnie. Kroki, które zachodzą w tym samym czasie, są nazywane *współbieżnymi* (ang. *concurrent*) lub *równoległymi* (ang. *parallel*).

Równoległe akcje prezentowane są na diagramach przy użyciu *rozwidleń* (ang. *forks*) oraz *scaleni* (ang. *joins*), w sposób zaprezentowany na fragmencie diagramu czynności przedstawionym na rysunku 3.9.



Rysunek 3.9. Obie krawędzie wychodzące przetwarzane są w węzle rozwidlenia, w przeciwieństwie do węzłów decyzyjnych, które używają jedynie jednej krawędzi wychodzącej

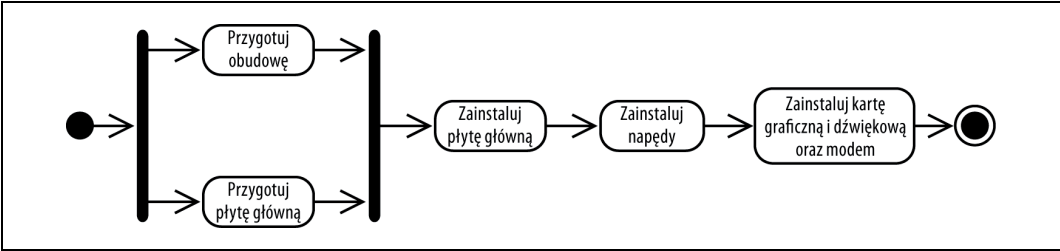
Po rozwidleniu na rysunku 3.9 następuje rozdzielanie na dwa lub więcej równoczesnych przepływów i akcje z nich wykonywane są równocześnie. Na rysunku 3.9 akcje Przygotuj obudowę oraz Przygotuj płytę główną rozpoczynane są jednocześnie.

Scalenie oznacza, że wszystkie wchodzące akcje muszą zakończyć się, zanim będzie mógł być przetwarzany dalej przepływ za scaleniem. Rozwidlenia oraz scalenia *wyglądają* identycznie (oba są rysowane przy użyciu grubych linii), jednakże można je rozróżnić, ponieważ te pierwsze mają wiele przepływów wychodzących, podczas gdy drugie — wchodzących.



W szczegółowym modelu projektu rozwidlenia mogą zostać użyte w celu reprezentacji wielu procesów lub wielu wątków programu.

Rysunek 3.10 przedstawia dokończony diagram dla przepływu czynności zachodzących podczas montażu komputerów.



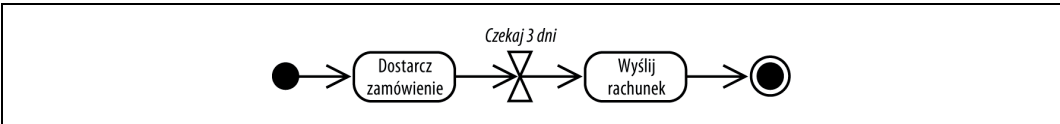
Rysunek 3.10. Przepływ czynności zachodzących podczas montażu komputerów demonstruje, w jaki sposób rozwidlenia oraz scalenia działają w kompletnym diagramie

Jeśli akcje zachodzą równolegle, nie musi to koniecznie oznaczać, że skończą się równocześnie. W rzeczywistości jedno z zadań najprawdopodobniej zakończy się przed innym. Jednakże scalenie nie pozwoli na kontynuację przepływu zdefiniowanego za nim do czasu, aż wszystkie wchodzące przepływy nie zostaną zakończone. Na przykład na rysunku 3.10 akcja występująca po scaleniu o nazwie Zainstaluj płytę główną zostanie wykonana jedynie po zakończeniu obu wcześniejszych akcji — Przygotuj obudowę oraz Przygotuj płytę główną.

Zdarzenia czasowe

Niekiedy również *czas* jest czynnikiem w danej czynności. Można chcieć zamodelować okres oczekiwania, na przykład na chociażby trzy dni, które muszą upłynąć po wysłaniu towaru, zanim będzie mógł zostać wysłany rachunek. Może również zaistnieć konieczność zamodelowania procesów, które uruchamiane są w regularnych odstępach czasu, jak chociażby cotygodniowe tworzenie kopii roboczej systemu.

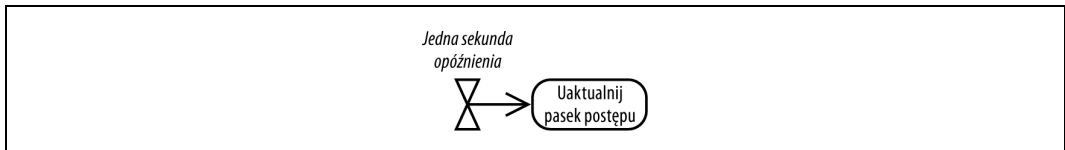
Zdarzenia czasowe (ang. *time events* — *przyp. tłum.*) przedstawiane są przy użyciu symbolu klepsydry. Rysunek 3.11 prezentuje, w jaki sposób zdarzenie czasowe może zostać użyte do zamodelowania okresu oczekiwania. Tekst umieszczony obok klepsydry (Czekaj 3 dni) określa ilość czasu, jaki musi upłynąć. Krawędź wchodząca do zdarzenia czasowego oznacza, że jest ono aktywowane tylko raz. Z rysunku 3.11 wynika więc, że rachunek jest wysyłany jedynie raz, a nie co trzy dni.



Rysunek 3.11. Zdarzenie czasowe z wchodzącą krawędzią reprezentuje oczekiwanie

Zdarzenie czasowe bez wchodzących przepływów jest *cykliczne* (ang. *recurring* — *przyp. tłum.*), co oznacza, że jest aktywowane w odstępach czasu podanych obok symbolu klepsydry. Z rysunku 3.12 wynika, że opisywany na nim pasek postępu jest uaktualniany co sekundę.

Należy zauważyć, że na rysunku 3.12 brakuje węzła początkowego. Zdarzenie czasowe jest alternatywnym sposobem rozpoczęcia czynności. Notacja ta powinna być używana do modelowania czynności wykonywanej cyklicznie.

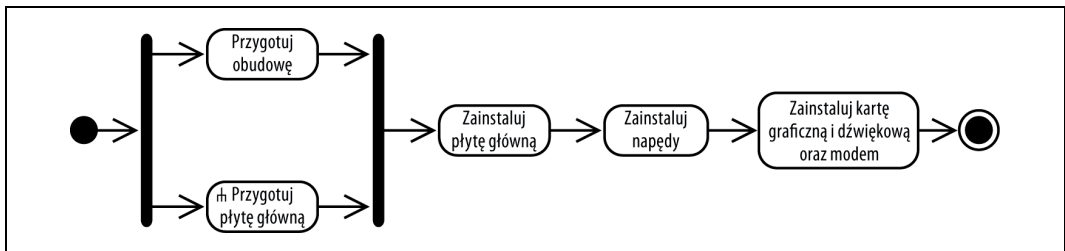


Rysunek 3.12. Zdarzenie czasowe bez wchodzących przepływów modelu reprezentuje zdarzenie cykliczne

Wywoływanie innych czynności

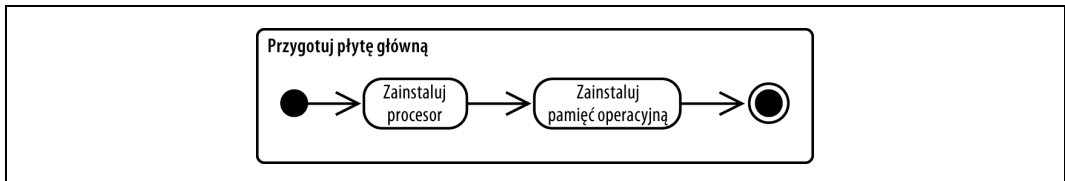
W miarę dodawania szczegółów do diagramu czynności może stać się on zbyt duży lub też ta sama sekwencja akcji może występować na nim więcej niż raz. W takiej sytuacji można zwiększyć czytelność diagramu głównego poprzez udostępnienie szczegółów akcji na oddzielnym rysunku. Umożliwi to mniejsze zaśmiecenie diagramu wysokiego poziomu.

Na rysunku 3.13 przedstawiony został przepływ czynności, które zachodzą podczas montażu komputerów, przeniesionych z rysunku 3.10. Jednakże tym razem obok akcji o nazwie *Przygotuj płytę główną* umieszczony jest odwrócony do góry nogami symbol widel sygnalizujący, że jest to *węzeł wywołania czynności* (ang. *call activity node*). Wywołuje on czynność powiązaną z jej nazwą. Przypomina to wywoływanie procedury w programie.



Rysunek 3.13. Aby szczegóły akcji *Przygotuj płytę główną* nie zaśmieszczały diagramu ogólnego, zostały przedstawione na kolejnym diagramie czynności

Węzeł o nazwie *Przygotuj płytę główną* umieszczony na rysunku 3.13 wywołuje czynność *Przygotuj płytę główną* przedstawioną na rysunku 3.14. Węzeł wywołania czynności może zostać skojarzony z wywoływana przez niego czynnością dzięki nadaniu im tej samej nazwy. Wywołanie czynności zasadniczo rozбивa daną akcję na bardziej szczegółową bez konieczności przedstawiania wszystkich szczegółów na jednym diagramie.



Rysunek 3.14. Diagram czynności o nazwie *Przygotuj płytę główną* rozwija proces przygotowywania płyty głównej

Diagram czynności o nazwie *Przygotuj płytę główną* dysponuje swoim własnym węzłem początkowym oraz końcowym czynności. Ten ostatni węzeł oznacza koniec czynności o nazwie *Przygotuj płytę główną*, lecz nie oznacza zakończenia wywołującej go czynności. W momencie zakończenia czynności o nazwie *Przygotuj płytę główną* sterowanie zwracane jest do

czynności wywołującej, która kontynuuje swoje działanie. To kolejny powód, dla którego wywołanie czynności przypomina wywołanie procedur w programie.



Pominięcie ramki jest możliwe do zaakceptowania dla czynności ogólnych. Powinna ona jednak być zawsze używana dla czynności wywoływanych. Nazwa czynności umieszczona w ramce pomoże skojarzyć czynności wywołwane z wywołującą.

Obiekty

Niekiedy ważnym aspektem modelowanego procesu są obiekty. Przypuśćmy, że firma czytelnika zdecydowała o sprzedaży systemu CMS jako produktu komercyjnego, a on chciałby zdefiniować proces akceptacji napływających zamówień. Każdy krok w procesie akceptacji zamówień będzie potrzebował informacji o zamówieniu, takich jak dotyczące płatności oraz kosztów transakcji. Informacje tego rodzaju mogą zostać zamodelowane na diagramie czynności przy użyciu obiektu *Order*, który zawierać będzie informacje o zamówieniu wymagane w kolejnych krokach. Diagramy czynności oferują szereg sposobów modelowania obiektów w procesach.

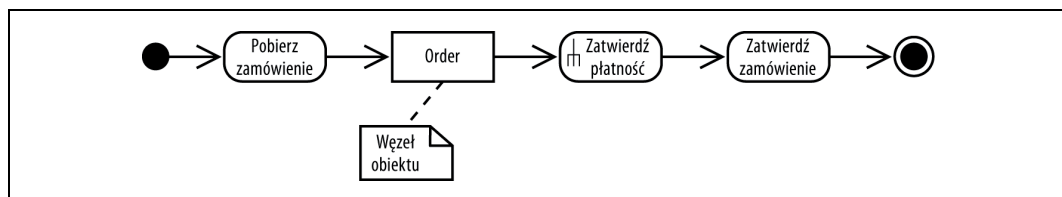


Obiekty nie muszą być programowe. Na przykład w przypadku czynności nieautomatyzowanego montażu komputerów węzeł obiektu może być używany do reprezentowania zamówienia pracy, które rozpoczyna cały proces.

Obrazowanie obiektów przekazywanych pomiędzy akcjami

W celu ukazania danych przepływających przez czynność można użyć na diagramach czynności tak zwanych *węzłów obiektów* (ang. *object nodes*). Węzeł obiektu reprezentuje obiekt, który jest dostępny w określonym miejscu czynności. Może on zostać użyty w celu zaprezentowania faktu, że dany obiekt jest używany, tworzony lub modyfikowany przez dowolną z otaczających go akcji.

Węzeł obiektu reprezentowany jest jako prostokąt, co zostało pokazane na diagramie przedstawiającym proces zatwierdzania zamówienia na rysunku 3.15. Węzeł obiektu *Order* sygnalizuje fakt jego przepływu od akcji *Pobierz zamówienie* do *Zatwierdź płatność*.

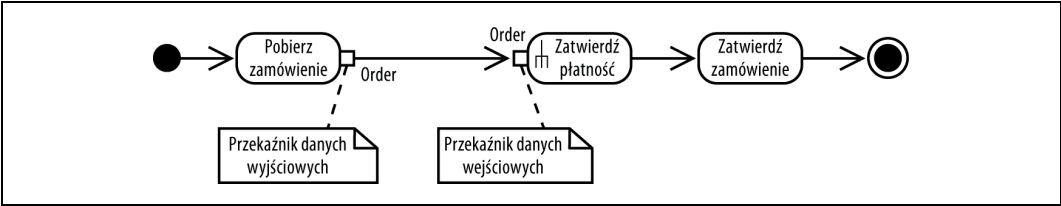


Rysunek 3.15. Węzeł obiektu o nazwie *Order* podkreśla, że jest on istotną informacją w tej czynności, a także pokazuje, które akcje go wykorzystują

Bardziej precyzyjny opis modelowania akcji *Pobierz zamówienie* w postaci węzła odbioru sygnału został umieszczony w podrozdziale zatytułowanym „Nadawanie oraz odbieranie sygnałów”.

Prezentacja danych wejściowych oraz wyjściowych akcji

Rysunek 3.16 przedstawia poprzednią czynność z innej perspektywy, wykorzystującej *przekazniki danych* (ang. *pins*). Przekazniki danych sygnalizują, że obiekt stanowi daną wejściową lub wyjściową akcji.



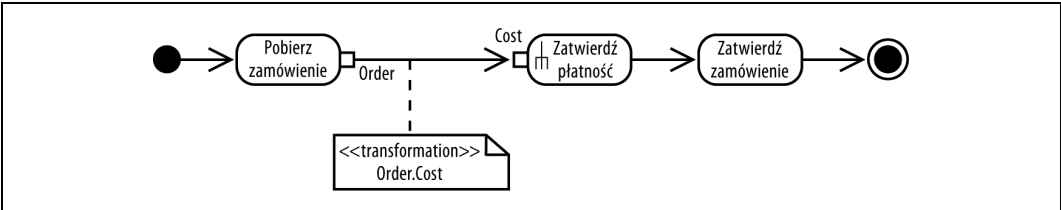
Rysunek 3.16. Przekazniki danych w tym procesie zatwierdzania zamówienia pozwalają na dokładniejszą specyfikację parametrów wejściowych oraz wyjściowych

Przekaznik danych wejściowych (ang. *input pin*) oznacza, że określony obiekt to dane wejściowe akcji. Przekaznik danych wyjściowych (ang. *output pin*) oznacza, że określony obiekt to dane wyjściowe z akcji. Na rysunku 3.16 obiekt *Order* oznacza dane wejściowe dla akcji *Zatwierdź płatność*. Jest on również daną wyjściową akcji *Pobierz zamówienie*.

Na rysunku 3.15 oraz 3.16 przedstawione są analogiczne sytuacje, jednak użycie przekazników danych pomaga uwypuklić fakt, że obiekt stanowi wymagane dane wejściowe oraz wyjściowe. Użycie węzła obiektu oznacza zwyczajnie, że obiekt jest dostępny w konkretnym punkcie czynności. Niemniej węzły obiektu posiadają swoją własną mocną stronę — pomagają w uwypukleniu przepływu danych w czynności.

Jeżeli akcja *Zatwierdź płatność* wymaga jedynie części obiektu *Order*, a nie całego, wtedy można użyć *przekształcenia* (ang. *transformation*) w celu pokazania, które części są wymagane. Przekształcenia pozwalają na pokazanie, w jaki sposób dane wyjściowe z jednej akcji stanowią dane wejściowe dla innej.

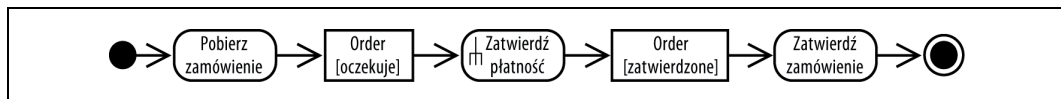
Z rysunku 3.17 wynika, że akcja o nazwie *Zatwierdź płatność* wymaga jako danej wejściowej obiektu reprezentującego koszt o nazwie *Cost*, który uzyskiwany jest z obiektu *Order*, co sygnalizuje przekształcenie określone w notatce.



Rysunek 3.17. Przekształcenie pokazuje, skąd pochodzą parametry wejściowe

Prezentacja zmiany stanu obiektów w czynności

Istnieje również możliwość ukazania zmiany stanu obiektu w trakcie jego przepływu przez czynność. Z rysunku 3.18 wynika, że obiekt o nazwie *Order* jest w stanie nazwanym *oczekuje* przed akcją *Zatwierdź płatność* oraz zmienia swój stan na *zatwierdzone* zaraz po niej. Stan obiektu przedstawiany jest w nawiasach kwadratowych.

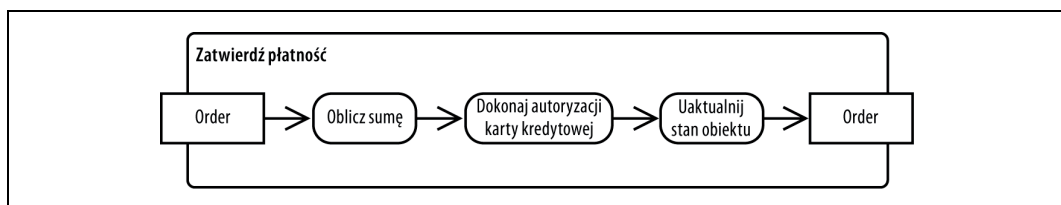


Rysunek 3.18. Ten diagram koncentruje się na opisie zmiany stanu obiektu *Order* w trakcie procesu zatwierdzania zamówienia

Prezentacja danych wejściowych oraz wyjściowych czynności

Oprócz stanowienia danych wejściowych oraz wyjściowych akcji węzły obiektów mogą być również danymi wejściowymi oraz wyjściowymi czynności. Dane wejściowe oraz wyjściowe czynności przedstawiane są w postaci węzłów obiektów leżących na granicy jej ramki, tak jak to widać na rysunku 3.19. Tego rodzaju notacja przydatna jest do podkreślenia faktu, że cała czynność wymaga danych wejściowych oraz udostępnia wyjściowe.

Rysunek 3.19 przedstawia obiekt *Order* w charakterze danych wejściowych oraz wyjściowych czynności o nazwie *Zatwierdź płatność*. W sytuacji gdy przedstawiane są parametry wejściowe oraz wyjściowe, węzeł początkowy oraz końcowy czynności jest usuwany z jej diagramu.



Rysunek 3.19. Węzły obiektów mogą być stosowane w celu podkreślenia danych wejściowych oraz wyjściowych z czynności

Nadawanie oraz odbieranie sygnałów

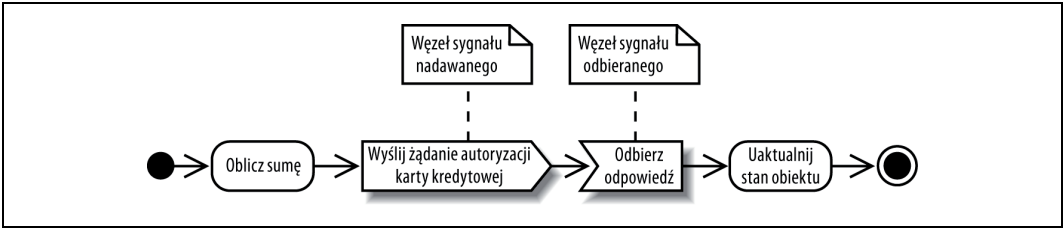
Czynności mogą wymagać interakcji z zewnętrznymi osobami, systemami lub procesami. Na przykład w momencie dokonywania płatności przy użyciu karty kredytowej konieczne jest sprawdzenie danych karty przy użyciu usługi jej autoryzacji udostępnianej przez jej wystawcę.

Na diagramach czynności *sygnały* (ang. *signals*) reprezentują interakcje z zewnętrznymi uczestnikami. Sygnały są komunikatami, które mogą być nadawane oraz odbierane, tak jak ma to miejsce w następujących sytuacjach:

- Program wysyła żądanie do wystawcy karty kredytowej w celu zaakceptowania transakcji przy jej użyciu, a następnie odbiera od niego odpowiedź (sygnał został wysłany oraz odebrany z punktu widzenia czynności określającej akceptację karty płatniczej).
- Otrzymanie zgłoszenia zamówienia powoduje rozpoczęcie procesu jego obsługi (z punktu widzenia czynności obsługi zamówienia sygnał został odebrany).
- Kliknięcie przycisku powoduje wykonanie skojarzonego z nim kodu (z punktu widzenia czynności obsługi zdarzenia naciśnięcia przycisku sygnał został odebrany).
- System informuje klienta, że jego zamówienie jest opóźnione (z punktu widzenia czynności obsługi zamówienia sygnał został wysłany).

Węzeł sygnału odbieranego (ang. *receive signal node*) może spowodować uruchomienie akcji przedstawionej na diagramie czynności. Odbiorca sygnału wie, w jaki sposób ma na niego zareagować, i oczekuje nadejścia sygnału w pewnym momencie, choć nie wie, kiedy to dokładnie nastąpi. Węzeł sygnału nadawanego (ang. *send signal node*) wysyła go do zewnętrznych uczestników. W chwili gdy zewnętrzny (w stosunku do systemu — *przyp. tłum.*) człowiek lub system otrzyma komunikat, w odpowiedzi na niego wykona prawdopodobnie pewną czynność, która nie jest jednak modelowana na diagramie.

Rysunek 3.20 zawiera uszczegółowione kroki z rysunku 3.19 w celu pokazania, że akcja autoryzacji karty kredytowej wymaga komunikacji z zewnętrznym programem. Węzeł sygnału wychodzącego oznacza, że sygnał jest wysyłany do zewnętrznego uczestnika. W tym przykładzie sygnał jest żądaniem zatwierdzenia karty kredytowej. Sygnały nadawane są w sposób asynchroniczny, co oznacza, że czynność nie musi oczekiwać na odpowiedź, lecz po wysłaniu sygnału kontynuuje natychmiast działanie, począwszy od kolejnej akcji.



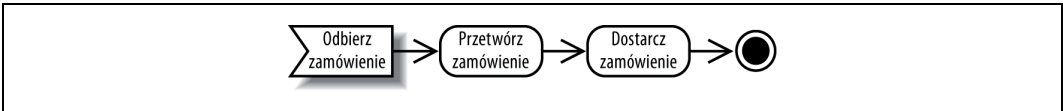
Rysunek 3.20. Węzły sygnału nadawanego oraz odbieranego wskazują na interakcję z zewnętrznymi uczestnikami

Węzeł sygnału odbieranego wskazuje, że sygnał odbierany jest z zewnętrznego procesu. W tym przypadku system oczekuje na odpowiedź od wystawcy karty kredytowej. Po napotkaniu węzła sygnału odbieranego akcja oczekuje do czasu odbioru sygnału, a czynność jest kontynuowana jedynie w przypadku jego otrzymania.



Należy zauważyć, że łączenie sygnałów nadawanych oraz odbieranych powoduje w rezultacie zachowanie podobne do wywołania synchronicznego lub oczekującego na odpowiedź. Sygnały nadawane oraz odbierane są bardzo często łączone na diagramach czynności, ponieważ bardzo często zachodzi potrzeba uzyskania odpowiedzi na te pierwsze.

Jeżeli napotkamy węzeł sygnału odbieranego bez przepływu wchodzącego, oznacza to, że w przypadku gdy czynność zawierająca węzeł jest aktywna, on zawsze oczekuje na sygnał. W przypadku przedstawionym na rysunku 3.21 czynność jest wykonywana zawsze po otrzymaniu zamówienia.



Rysunek 3.21. Czynność rozpoczynająca się od węzła sygnału odbieranego, zastępującego występujący zazwyczaj węzeł początkowy

Taki węzeł różni się od węzła sygnału odbieranego z wchodzącą krawędzią, jak chociażby tego o nazwie Odbierz odpowiedź z rysunku 3.20. Taki węzeł sygnału odbieranego z wchodzącą krawędzią zaczyna oczekiwanie dopiero po zakończeniu poprzedniej akcji.

Rozpoczynanie czynności

Najprostszym oraz najczęstszym sposobem rozpoczynania czynności jest użycie pojedynczego węzła początkowego. Większość diagramów, które dotąd widzieliśmy, używa tej notacji. Istnieją jednak również inne sposoby rozpoczynania czynności mające specjalne znaczenie:

- Czynność rozpoczyna się od odebrania danych wejściowych, co zostało omówione wcześniej, w podrozdziale zatytułowanym „Prezentacja danych wejściowych oraz wyjściowych czynności”.
- Czynność rozpoczyna się w odpowiedzi na zdarzenia czasowe, co zostało omówione wcześniej, w podrozdziale zatytułowanym „Zdarzenia czasowe”.
- Czynność rozpoczyna się w wyniku wzbudzenia przez sygnał.

Aby zasygnalizować, że czynność rozpoczyna się w wyniku wzbudzenia przez sygnał, należy zamiast węzła początkowego użyć *węzła sygnału odbieranego*. Wewnątrz tego węzła należy określić rodzaj sygnału, który będzie rozpoczynać daną czynność. Na rysunku 3.21 przedstawiona została czynność rozpoczynana po otrzymaniu zamówienia.

Kończenie czynności oraz przepływów

Do tej pory węzły końcowe czynności nie były zbyt interesujące. W rzeczywistości nie były one używane jako nic innego niż zwykłe znaczniki końca. W realnych sytuacjach można napotkać znacznie bardziej złożone zakończenia procesów. Przepływy na przykład mogą być przerywane, ale mogą się też zakończyć bez przerywania całej czynności.

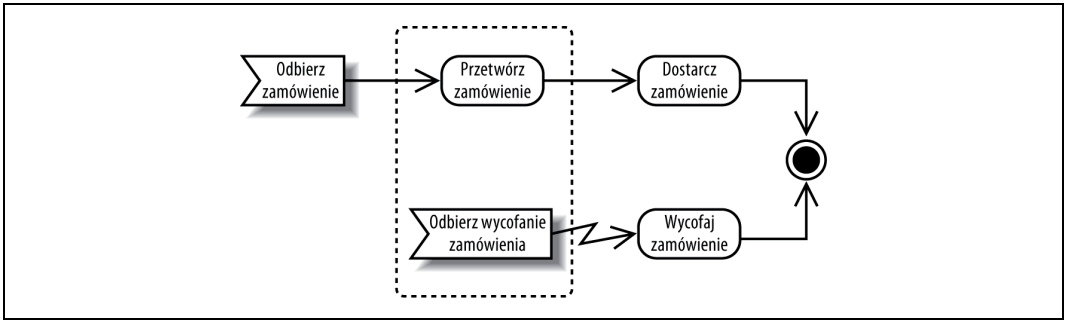
Przerywanie czynności

Na rysunku 3.21 przedstawiony został typowy diagram czynności z pojedynczym zakończeniem. Należy zauważyć, że istnieje tylko jedna ścieżka prowadząca do końcowego węzła czynności. Każda akcja na tym diagramie ma szansę zakończenia.

Niekiedy istnieje konieczność zamodelowania procesu, który może być przerywany przez zdarzenie. Taka sytuacja mogłaby się zdarzyć, gdybyśmy na przykład mieli długo wykonujący się proces, który mógłby zostać przerywany przez użytkownika. Natomiast w czynności obsługi zamówienia systemu CMS mogłaby zachodzić konieczność zaksięgowania odwołanego zamówienia. Tego rodzaju przerywania mogą być przedstawiane przy użyciu *obszarów przerwania* (ang. *interruption regions*).

Obszar przerwania oznaczany jest przy użyciu zaokrąglonego prostokąta narysowanego za pomocą linii przerywanej, otaczającego akcje, które mogą zostać przerywane, oraz zdarzenie mogące powodować przerwanie. Ze zdarzenia przerywającego wychodzi linia przypominająca błyskawicę. Rysunek 3.22 rozszerza rysunek 3.21 poprzez dodanie obsługi możliwości wycofania zamówienia.

Jeżeli w sytuacji przedstawionej na rysunku 3.22 prośba wycofania zamówienia zostanie otrzymana w chwili, gdy akcja o nazwie *Przetwórz zamówienie* jest aktywna, wtedy akcja ta zostanie przerywana, a aktywna stanie się *Wycofaj zamówienie*. Obszary przerwania odnoszą się jedynie do zawartych w nich akcji. Jeżeli wycofanie zamówienia zostanie otrzymane w chwili, gdy aktywna jest akcja *Dostarcz zamówienie*, wtedy nie zostanie ona przerywana, ponieważ nie znajduje się w obszarze przerwania.



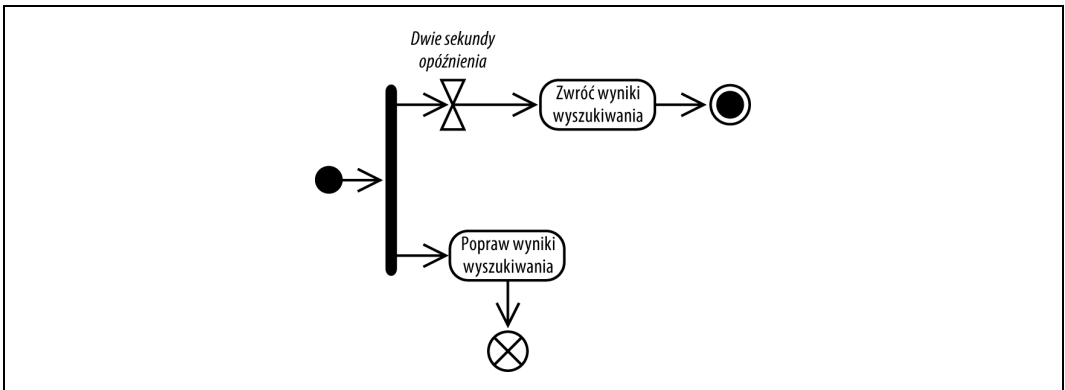
Rysunek 3.22. Obszar przerwań sygnalizuje możliwość przerywania procesu



Niekiedy można napotkać diagramy z wieloma końcowymi węzłami czynności, w odróżnieniu od takiego z wieloma przepływami wchodzącymi do pojedynczego końcowego węzła czynności. Taki diagram jest dopuszczalny i może pomóc w uporządkowaniu wielu linii na diagramie z wieloma rozgałęzieniami. Niemniej jednak diagramy czynności są zazwyczaj łatwiejsze do zrozumienia, jeżeli zawierają pojedynczy końcowy węzeł czynności.

Kończenie przepływu

Nową funkcją wersji UML 2.0 jest możliwość pokazania końca przepływu bez konieczności kończenia całej czynności. Węzeł końcowy przepływu (ang. *flow final node*) kończy jedynie własną ścieżkę, a nie całą czynność. Jest on oznaczany przy użyciu symbolu koła z wpisanym znakiem X, jak to zaprezentowano na rysunku 3.23.



Rysunek 3.23. Węzeł końcowy przepływu kończy jedynie własną ścieżkę, a nie całą czynność

Rysunek 3.23 przedstawia mechanizm wyszukiwania dla systemu CMS, z dwu sekundowym oknem do tworzenia najlepszych możliwych wyników wyszukiwania. Po wystąpieniu dwusekundowego opóźnienia wyniki wyszukiwania są zwracane i cała czynność, włączając akcję Popraw wyniki wyszukiwania, kończy się. Jednakże jeżeli akcja Popraw wyniki wyszukiwania zakończy się przed upływem dwusekundowego czasu oczekiwania, wtedy nie spowoduje to zakończenia całej czynności, ponieważ przepływ kończy się węzłem końcowym przepływu.



Używając węzła końcowego przepływu po rozgałęzieniu, należy zachować ostrożność. Po napotkaniu węzła końcowego zakończą się wszystkie inne akcje w danej czynności (również te przed węzłem końcowym). Jeżeli chcemy, aby wszystkie rozwidłone akcje zakończyły całkowicie swoje działanie, wtedy należy dodać scalenie.

Partycje (tory pływackie)

W czynności mogą brać udział różni uczestnicy, jak chociażby różne grupy lub role w organizacji lub systemie. Poniższe scenariusze wymagają do dokończenia czynności udziału wielu uczestników (nazwy uczestników napisane są czcionką pochyloną):

Czynność przetwarzania zamówienia

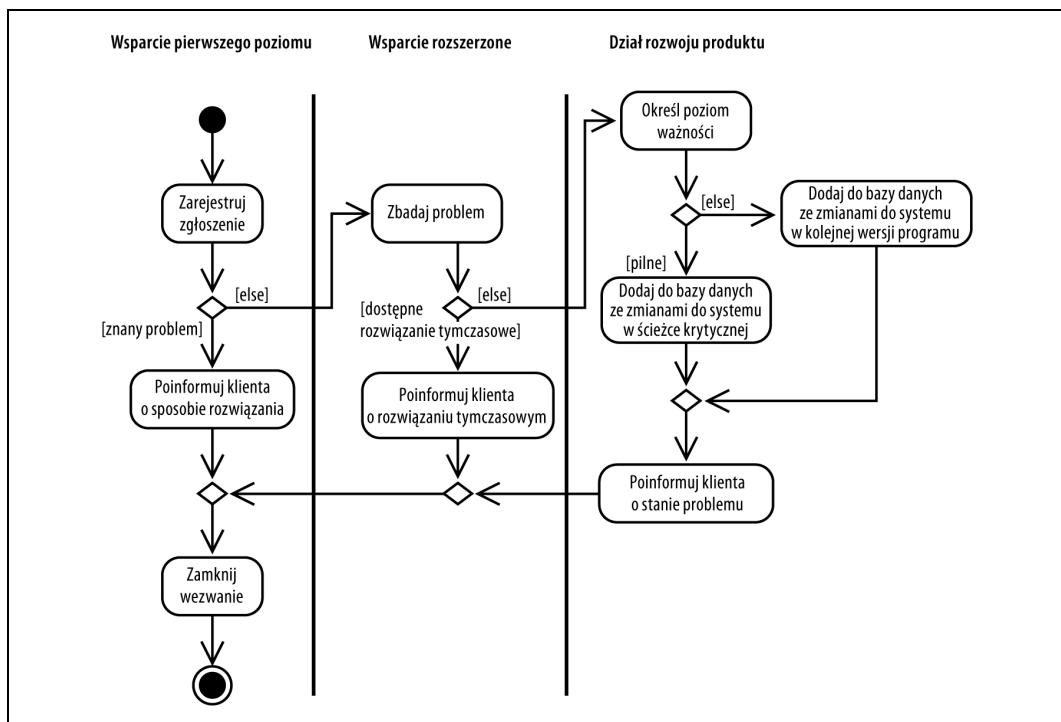
Wymaga od *działu logistyki* dostarczenia produktu oraz od *działu księgowości* wystawienia rachunku dla klienta.

Proces obsługi technicznej

Wymaga różnych poziomów obsługi, do których należą *wsparcie pierwszego poziomu*, *wsparcie rozszerzone* oraz *dział rozwoju produktu*.

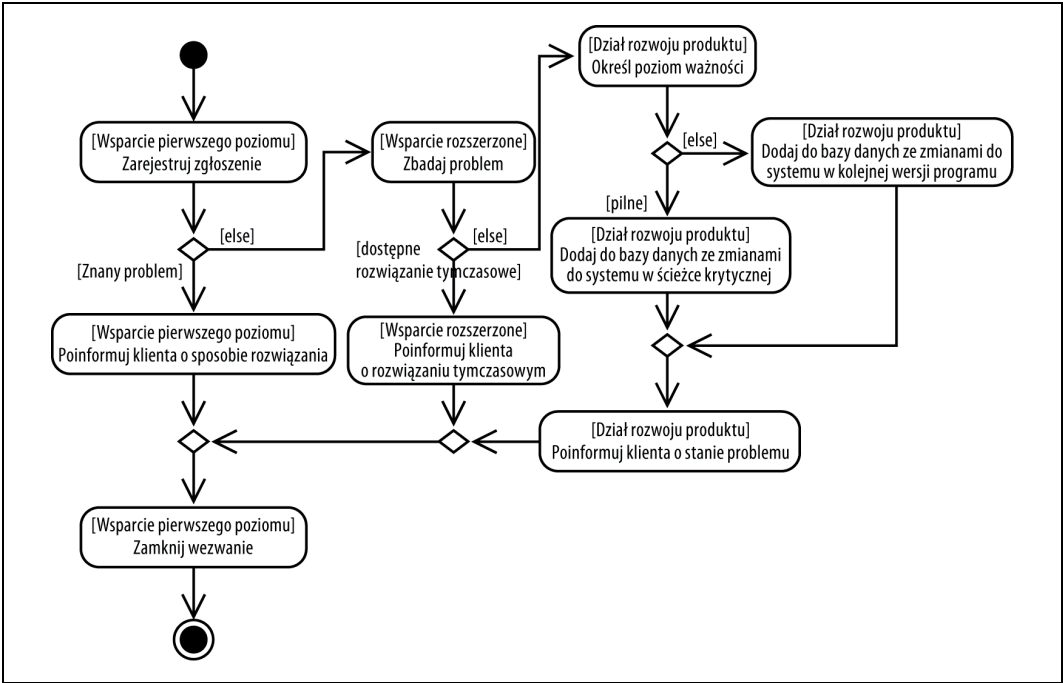
Aby pokazać, za które akcje są odpowiedzialni dani uczestnicy, można użyć *partycji* (ang. *partitions*). Partycje dzielą diagram na wiersze oraz kolumny (w zależności od usytuowania diagramu czynności) i zawierają akcje, które są wykonywane przez odpowiedzialne za nie grupy. Kolumny oraz wiersze są niekiedy określane mianem *torów pływackich* (ang. *swimlanes*).

Na rysunku 3.24 przedstawiony jest proces obsługi technicznej uwzględniający trzy rodzaje uczestników: wsparcie pierwszego poziomu, wsparcie rozszerzone oraz dział rozwoju produktu.



Rysunek 3.24. Partycje pomagają w organizacji diagramu czynności, określając strony odpowiedzialne

Odpowiedzialność może być również zasygnalizowana przy użyciu *adnotacji* (ang. *annotations*). Warto zauważyć, że w takim przypadku nie istnieją już tory pływackie. W zamian nazwa odpowiedzialnej strony jest umieszczona w nawiasach w samym węźle, jak pokazano to na rysunku 3.25. Tego rodzaju notacja zazwyczaj sprawia, że diagram jest bardziej zwężły, jednakże uczestnicy są przedstawieni mniej przejrzyście niż w przypadku torów pływackich.



Rysunek 3.25. Używanie adnotacji zamiast torów pływackich jest sposobem przedstawienia odpowiedzialności bezpośrednio w akcji

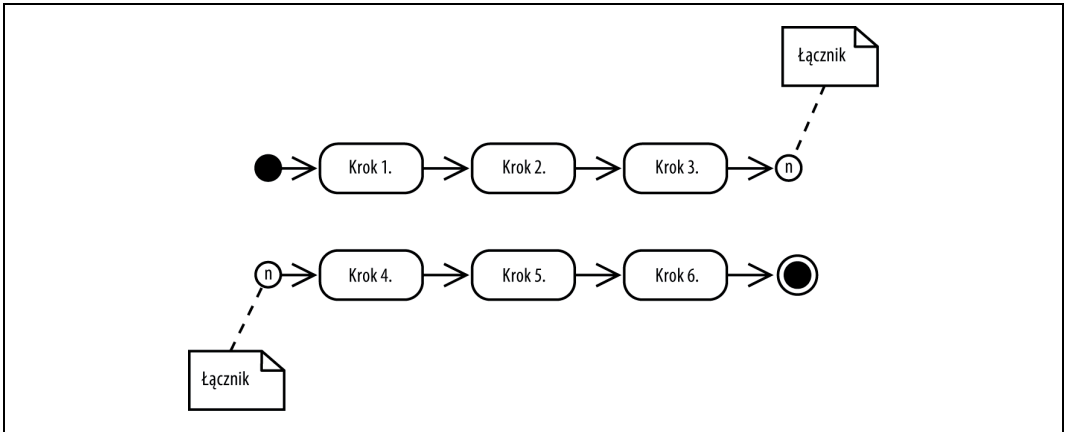
Zarządzanie złożonymi diagramami czynności

Diagramy czynności posiadają wiele dodatkowych symboli służących do modelowania szerokiego zakresu procesów. Przedstawione poniżej podrozdziały zawierają niektóre z wygodnych skrótów upraszczających diagramy czynności.

Łączniki

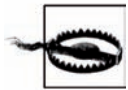
Jeżeli diagram czynności zawiera dużo akcji, efektem tego może być umieszczenie na nim wielu długich, przecinających się linii, co spowoduje, że będzie go trudniej odczytać. W takiej sytuacji pomagają właśnie *łączniki* (ang. *connectors* — *przyp. tłum.*).

Łączniki pomagają uprościć diagramy, łącząc krawędzie z symbolami, a nie z konkretnymi liniami. Łącznik oznaczany jest za pomocą kółka z wpisaną nazwą. Nazwy łączników przyjmują najczęściej postać jednoliterową. W przypadku rysunku 3.26 nazwą łącznika jest litera n.



Rysunek 3.26. Łączniki mogą zwiększyć przejrzystość dużych diagramów czynności

Łączniki występują w parach — jeden z nich ma krawędź wchodzącą, drugi wychodzącą. Drugi łącznik rozpoczyna się w miejscu zakończenia pierwszego. Dlatego też przepływ na rysunku 3.26 jest identyczny z tym, w którym Krok 3. miałby krawędź prowadzącą bezpośrednio do Kroku 4.

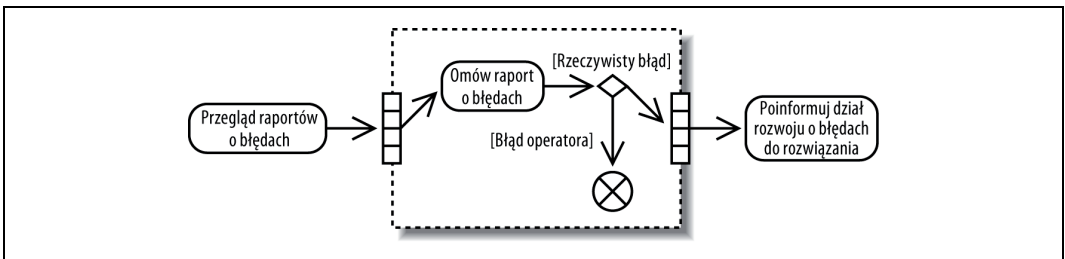


Z łącznikami należy uważać. Jeżeli na jednym diagramie zostanie użytych zbyt wiele różnych łączników, jego czytelnik może stracić zbyt wiele czasu, próbując je połączyć.

Obszary rozszerzenia

Obszary rozszerzenia (ang. *expansion regions* — *przyp. tłum.*) sygnalizują, że akcje umieszczone w tym obszarze są wykonywane dla każdego obiektu ze zbioru wejściowego. Na przykład obszar rozszerzenia mógłby zostać użyty w celu zamodelowania funkcji oprogramowania przyjmującej jako dane wejściowe listę plików, a następnie przeszukującej każdy z nich na obecność szukanego wyrazu.

Obszary rozszerzenia przedstawiane są jako duże zaokrąglone prostokąty narysowane przerywaną linią z czterema kwadracikami dołączonymi z każdej strony. Cztery kwadraty reprezentują zbiory wejściowe oraz wyjściowe (ale nie wymuszają ograniczenia rozmiaru zbioru do czterech elementów). Z rysunku 3.27 wynika, że raport o błędach jest dyskutowany dla każdego raportu ze zbioru wejściowego. Jeżeli jest to rzeczywisty błąd, wtedy czynność jest wykonywana dalej. W przeciwnym przypadku błąd jest odrzucany, a przepływ dla tych danych wejściowych jest kończony.



Rysunek 3.27. Akcje w obszarze rozszerzenia są wykonywane dla każdego z elementów zbioru

Co dalej?

Diagramy komunikacji oraz sekwencji są kolejnymi diagramami UML-a, które mogą służyć do modelowania dynamicznego zachowania systemu. Diagramy te koncentrują się na ukazywaniu sekwencji zdarzeń oraz szczegółów interakcji, jak chociażby tego, które obiekty są użyte w interakcji, jakie metody są wywoływane. Diagramy sekwencji są opisane w rozdziale 7., natomiast diagramy komunikacji w rozdziale 8.

Jeżeli czytelnik nie zapoznał się jeszcze z rozdziałem 2., opisującym przypadki użycia, warto polecić jego przeczytanie, ponieważ diagramy czynności udostępniają właśnie wspólną metodę na przedstawienie graficznej reprezentacji przepływu przypadków sterowania.