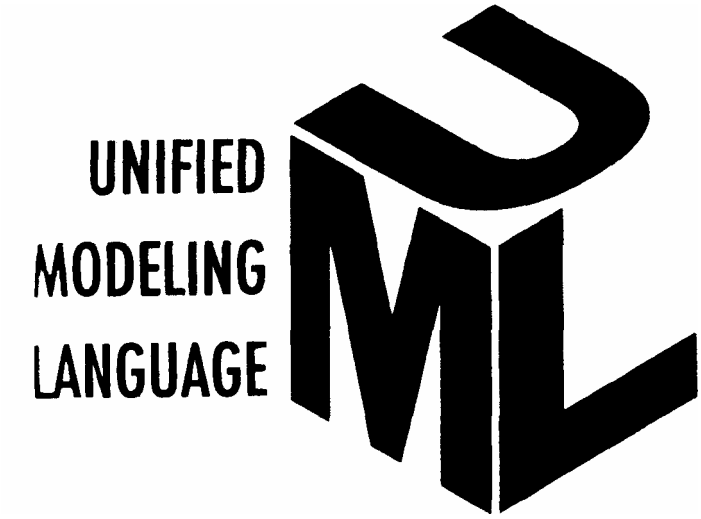


- Przypadki Użycia

- Diagramy Przypadków Użycia

Wersja elektroniczna: mgr inż. Kamil Kuliberda;
na podstawie „UML przewodnik użytkownika”,
Grady Booch, Jamek Rumbaugh, Ivar Jacobson.

Przypadki Użycia

Poruszone tematy:

- Przypadki użycia, aktorzy, związki zawierania i rozszerzania
- Modelowanie zachowania bytu
- Realizacja przypadków użycia za pomocą kooperacji

Żaden system nie istnieje w próżni. Każde interesujące oprogramowanie współdziała z żywymi lub zautomatyzowanymi aktorami, którzy, korzystając z niego, osiągają pewne cele. Ci aktorzy spodziewają się, że system będzie funkcjonował w przewidywany sposób. Przypadek użycia specyfikuje zachowanie systemu lub jego części. Jest to opis zbioru ciągów akcji (i ich wariantów) wykonywanych przez system w celu dostarczenia określonemu aktorowi godnego uwagi wyniku.
Przypadki użycia służą do utrwalania oczekiwanego zachowania budowanego systemu bez konieczności określania sposobu implementacji tego zachowania. Umożliwiają wypracowanie porozumienia między programistami a użytkownikami i ekspertami w dziedzinie problemu. Co więcej, ułatwiają weryfikację architektury i samego systemu w miarę tworzenia go. Podczas implementacji systemu przypadek użycia jest urzeczywistniany przez kooperację, której uczestnicy wspólnie dążą do jego realizacji.
Dobrze zbudowane przypadki użycia reprezentują jedynie zasadnicze aspekty zachowania systemu i nie są ani zbyt ogólne, ani zbyt szczegółowe.

Wprowadzenie

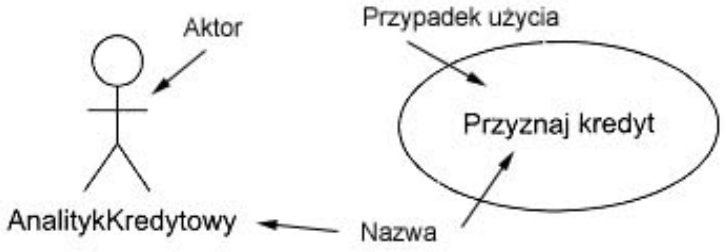
Dobrze zaprojektowany dom jest czymś więcej niż tylko zbieraniną ścian, które podtrzymują dach chroniący przed złą pogodą. Dyskutując z architektem o projekcie domu, z pewnością poświęcisz dużo uwagi temu, jak będziesz
z tego domu korzystać. Jeśli lubisz przyjęcia, rozważysz możliwości poruszania się i prowadzenia konwersacji w pokoju gościnnym; będziesz chciał uniknąć zakamarków, w których mogłyby gromadzić się niewielkie grupki gości. Myśląc o przygotowaniu posiłków, będziesz dążył do takiego zaprojektowania kuchni, żeby było w niej dużo miejsca na wszelkie urządzenia i na przechowywanie żywności. Nawet wyznaczenie drogi z garażu do kuchni, wzdłuż której będzie się przenosić zakupy, będzie miało wpływ na ostateczny układ połączeń między pokojami. Jeśli masz liczną rodzinę, zwrócisz uwagę na sprawy związane z użytkowaniem łazienki. Zaplanowanie odpowiedniej liczby łazienek i właściwego ich rozmieszczenia już we wczesnej fazie projektowania pozwoli Ci uniknąć późniejszych porannych zatorów, kiedy to wszyscy wstają i szykują się do pracy lub do szkoły. Szczególnie ważne będzie to dla nastolatków, którzy bardzo źle znoszą tego typu problemy przy szkolnych niepowodzeniach.
Rozważenie wszystkich możliwości korzystania z domu to przykład analizy opartej na przypadkach użycia, które są podstawowym czynnikiem kształtującym architekturę. Wiele rodzin może zdefiniować podobny zbiór przypadków użycia - w domu się jada, sypia, wychowuje dzieci i przechowuje pamiątki. Każda rodzina może także określić kilka specyficznych przypadków użycia lub wariantów wymienionych wcześniej podstawowych przypadków użycia. Potrzeby wielkiej

rodziny są inne niż potrzeby samotnej, dorosłej osoby, która niedawno ukończyła studia. Właśnie te warianty mają największy wpływ na końcową postać domu.
Bardzo ważne w procesie tworzenia przypadków użycia jest to, że nie definiuje się jednocześnie sposobu ich implementacji. Określając na przykład, jak powinien działać bankomat, możesz za pomocą przypadków użycia ustalić tryb jego współpracy z użytkownikami. Nie musisz przy tym wcale znać się na wnętrzu bankomatu. Przypadki użycia określają wymagane zachowanie, ale nie narzucają sposobu jego implementacji. Ich wielka zaleta polega na tym, że dzięki nim użytkownicy i eksperci w dziedzinie problemu mogą rozmawiać z projektantami i programistami bez konieczności analizowania nieistotnych szczegółów. Tymi szczegółami i tak trzeba się będzie zająć, ale przypadki użycia umożliwiają skoncentrowanie się na sprawach o większym znaczeniu i takich, które mogą zagrażać powodzeniu danego przedsięwzięcia.
W UML takie zagadnienia modeluje się w postaci przypadków użycia, których specyfikacje są niezależne od realizacji. Przypadek użycia to opis zbioru ciągów akcji (i ich wariantów) wykonywanych przez system w celu dostarczenia określonemu aktorowi godnego uwagi wyniku. W definicji tej jest kilka ważnych fragmentów.

Przypadek użycia określa zbiór ciągów akcji, z których każdy reprezentuje interakcję elementów z otoczenia systemu (jego aktorów) z samym systemem (i jego głównymi abstrakcjami). Te działania są w istocie funkcjami systemowymi, których na etapie identyfikacji wymagań i analizy używasz do obrazowania, specyfikowania, tworzenia i dokumentowania spodziewanego zachowania systemu. Przypadek użycia reprezentuje wymaganie funkcjonalne dla systemu jako całości. W systemie bankowym jednym z najważniejszych przypadków użycia będzie na przykład przyznawanie kredytów.
Przypadek użycia obejmuje interakcję aktorów i systemu. Aktor reprezentuje spójny zbiór ról odgrywanych przez użytkowników przypadku użycia w czasie jego realizacji. Może to być człowiek lub zautomatyzowany system. W modelu systemu bankowego przyznawanie kredytu będzie obejmować interakcję klienta i analityka kredytowego.
Przypadek użycia może mieć warianty. We wszystkich ciekawych systemach napotkasz przypadki użycia, które są uszczegółowionymi wersjami innych przypadków użycia, przypadki użycia, które są częściami innych przypadków użycia, oraz przypadki użycia, które rozszerzają znaczenie innych podstawowych przypadków użycia. Za pomocą tych trzech rodzajów związków możesz niejako wyłączyć przed nawias wspólne, zdatne do ponownego wykorzystania fragmenty zachowań opisanych przez pewien zbiór przypadków użycia. Modelując na przykład system bankowy, zidentyfikujesz mnóstwo wariantów podstawowego przypadku użycia, jakim jest przyznawanie kredytu. Kredyt hipoteczny o wielkiej wartości ma zupełnie inne właściwości niż niewielki kredyt handlowy. W obu wypadkach jednak te przypadki użycia obejmują pewne wspólne czynności, takie

jak ocena zdolności kredytowej klienta, którą należy przeprowadzić niezależnie od rodzaju kredytu.
Przypadek użycia ma za zadanie dostarczyć pewien godny uwagi wynik. Z punktu widzenia określonego aktora przypadek użycia opisuje działanie mające dla niego jakąś wartość, na przykład obliczenie wyniku, utworzenie nowego obiektu lub zmianę stanu danego obiektu. W modelu systemu bankowego przyznanie kredytu prowadzi do zatwierdzenia kredytu, czyli do udostępnienia klientowi pewnej kwoty pieniędzy.
Przypadki użycia służą do analizy całego systemu. Można je jednak zastosować także do analizy części systemu (np. podsystemów), a nawet pojedynczych klas i interfejsów. Przypadki użycia nie tylko definiują oczekiwane zachowanie tych bytów, ale są także podstawą do opracowania dla nich przypadków testowych w miarę rozwijania ich. Przypadki użycia dotyczące podsystemów są bardzo dobrym źródłem testów regresywnych, a przypadki

użycia dotyczące systemu - testów integracyjnych i systemowych. UML udostępnia symbole graficzne przypadku użycia i aktora (rys. 16.1). Notacja ta umożliwia zobrazowanie przypadku użycia niezależnie od jego realizacji i w towarzystwie innych przypadków użycia.



Rys. 16.1. Aktor i przypadek użycia

Pojęcia i ich definicje

Przypadek użycia to opis zbioru ciągów akcji (i ich wariantów) wykonywanych przez system w celu dostarczenia określonemu aktorowi godnego uwagi wyniku. Na diagramie przypadek użycia jest przedstawiany w postaci elipsy.

Nazwy

Każdy przypadek użycia musi mieć nazwę, która wyróżnia go spośród innych przypadków użycia. Nazwa jest napisem. Jeśli jest to sama nazwa, to mówimy o nazwie prostej. Jeśli nazwa jest poprzedzona nazwą pakietu, w którym dany

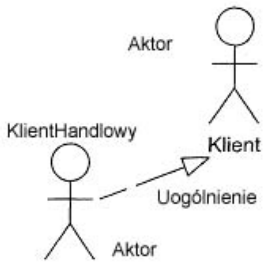
przypadek użycia zdefiniowano, to mówimy o nazwie ścieżkowej. Symbol przypadku użycia zawiera zwykle jedynie jego nazwę (rys. 16.2).

Rys. 16.2. Nazwy proste i ścieżkowe

Uwaga: Nazwa przypadku użycia może być tekstem zawierającym dowolną liczbę liter, cyfr i znaków przestankowych (z wyjątkiem dwukropka oddzielającego nazwę przypadku użycia od nazwy jego pakietu), zapisanym w wielu wierszach. W praktyce nazwę podaje się w formie krótkiego wyrażenia z czasownikiem w trybie rozkazującym na początku, określającego pewną czynność pochodzącą ze słownictwa modelowanego systemu.

Przypadki użycia i aktorzy

Aktor reprezentuje spójny zbiór ról odgrywanych przez użytkowników przypadku użycia w czasie interakcji z tym przypadkiem użycia. Zwykle jest to rola dla człowieka, urządzenia lub nawet innego systemu. Pracując w banku, możesz być na przykład AnalitykiemKredytowym. Jeśli jednocześnie korzystasz z usług tego banku, będziesz również odgrywać rolę Klienta. Egzemplarz aktora jest więc jednostką będącą w interakcji z systemem w specyficzny sposób. Choć aktorzy występują w modelach, nie są częścią systemu. Istnieją poza systemem.
Na rysunku 16.3 widać symbol graficzny aktora - schematyczny rysunek postaci ludzkiej złożony z kilku kresek. Stosując uogólnienia, możesz zdefiniować zarówno ogólne rodzaje aktorów (np. Klient), jak i ich uszczegółowienia (np. KlientHandlowy).



Rys. 16.3. Aktorzy

Uwaga: Mechanizmy rozszerzania UML odnoszą się także do aktorów. Możesz na przykład wprowadzić nowy stereotyp aktora i jego specyficzny symbol, aby za pomocą środków graficznych lepiej zobrazować swoje zamierzenia.

Związki między aktorami a przypadkami użycia mogą być jedynie powiązaniami. Takie powiązanie oznacza, że aktor i przypadek użycia porozumiewają się ze sobą, wysyłając i odbierając komunikaty.

Przypadki użycia i ciągi zdarzeń

Przypadek użycia opisuje, *co* system (podsystem, klasa lub operacja) robi, ale nie określa, *jak* to robi. Budując model, musisz pamiętać o zasadzie oddzielenia pojęć, polegającej na odizolowaniu tego, co dotyczy pracy systemu, od tego, co dotyczy jego realizacji.
Przypadek użycia może być wyspecyfikowany przez opisanie ciągu zdarzeń w formie tekstu zrozumiałego nawet dla laika. Zapisując ten ciąg zdarzeń, powinieneś uwzględnić informację o tym, jak i kiedy przypadek użycia zaczyna się i kończy, kiedy dochodzi do jego interakcji z aktorami i jakie obiekty są przekazywane. Należy także podać główny ciąg zdarzeń i inne, alternatywne.
Oto przykładowy opis przypadku użycia Weryfikuj Użytkownika, pochodzący z modelu oprogramowania bankomatu.

Główny ciąg zdarzeń: Przypadek użycia zaczyna się, gdy system pyta *Klienta* o PIN. *Klient* może teraz wprowadzić hasło z klawiatury. *Klient* potwierdza wprowadzone dane za pomocą klawisza Wykonaj. System sprawdza poprawność PIN-u. Jeśli PIN jest poprawny, system informuje o tym. Na tym kończy się przypadek użycia.

Nadzwyczajny ciąg zdarzeń: Klient może w każdej chwili anulować transakcję, naciskając klawisz Stop. Przypadek użycia wraca teraz do punktu początkowego. Na koncie *Klienta* nie zachodzą żadne zmiany.

Nadzwyczajny ciąg zdarzeń: Przed potwierdzeniem PIN-u *Klient* może w każdej chwili wprowadzić go jeszcze raz.

Uwaga: Ciąg zdarzeń przypadku użycia można określić na wiele sposobów. Można go zapisać w postaci nieformalnego tekstu strukturalnego (jak w naszym przykładzie), w postaci formalnego tekstu strukturalnego (z warunkami wstępnymi i końcowymi) lub w postaci pseudokodu.

Przypadki użycia i scenariusze

Zazwyczaj na początku ciąg zdarzeń przypadku użycia opisuje się tekstowo. W miarę coraz lepszego rozumienia wymagań stawianych systemowi przechodzi się do diagramów interakcji, żeby określić te ciągi graficznie. Zwykle jest to jeden diagram przebiegu przedstawiający główny ciąg zdarzeń oraz warianty tego diagramu definiujące ciągi nadzwyczajne.
Warto oddzielić główny ciąg zdarzeń od alternatywnych, ponieważ przypadek użycia obejmuje ich więcej niż jeden. Trudno, bowiem wyrazić wszystkie szczegóły złożonego przypadku użycia w jednym ciągu zdarzeń. W systemie kadrowym znajdziesz na przykład przypadek użycia Zatrudnij pracownika. Ta bardzo ogólna funkcja przedsiębiorstwa może mieć wiele wariantów. Może to być zatrudnienie osoby pracującej w innej firmie (najczęstszy scenariusz), przeniesienie pracownika z jednego działu do drugiego (bardzo częste w wielkich międzynarodowych firmach) lub nawet zatrudnienie obcokrajowca (zwykle podlega to specjalnym regulacjom prawnym). Każdy z tych scenariuszy może być wyrażony za pomocą innego ciągu zdarzeń.

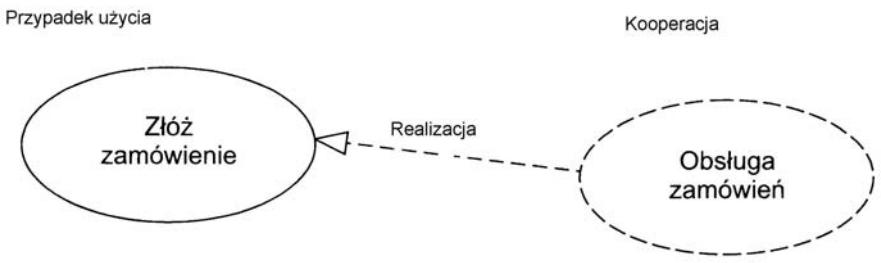
Omawiany tu przypadek użycia (Zatrudnij pracownika) w istocie opisuje zbiór ciągów, z których każdy reprezentuje jeden z możliwych przebiegów przez wszystkie te warianty. Każdy taki ciąg nosi nazwę scenariusza. Scenariusz jest pewnym szczególnym ciągiem akcji, który ilustruje specyfikowane zachowanie. Scenariusze są dla przypadków użycia tym, czym dla klas obiekty, to znaczy są egzemplarzami przypadków użycia.

Uwaga: Liczba scenariuszy jest zawsze znacznie większa niż liczba przypadków użycia. Średnio skomplikowany system ma około kilkudziesięciu przypadków użycia, z których każdy rozwija się do kilkudziesięciu scenariuszy. Każdy przypadek użycia ma scenariusze podstawowe (definiujące ciągi główne) i drugorzędne (definiujące ciągi alternatywne).

Przypadki użycia i kooperacje

Przypadek użycia opisuje oczekiwane zachowanie budowanego systemu (podsystemu, klasy lub operacji), ale nie określa sposobu implementacji tego zachowania. To rozgraniczenie jest bardzo ważne, ponieważ analiza systemu (specyfikacja zachowania) powinna być w jak największym stopniu niezależna od zagadnień implementacyjnych (dotyczących sposobu realizacji tego zachowania). Kiedyś jednak musi dojść do zaimplementowania przypadków użycia. Aby się z tym uporać, należy utworzyć zestawy klas i innych bytów, których współpraca doprowadzi do implementacji danego przypadku użycia. Te zestawy bytów, włącznie z ich strukturą statyczną i dynamiczną, modeluje
się w UML w postaci kooperacji.
Z rysunku 16.4 wynika, że realizację przypadku użycia można jawnie określić za pomocą kooperacji. W większości wypadków jednak poszczególne przypadki użycia są realizowane przez dokładnie jedną kooperację; nie trzeba, więc jawnie modelować tego związku.

Uwaga: Nawet jeśli nie zobrazujesz tego związku jawnie, narzędzia wspomagające zapewne przechowają informację o nim.



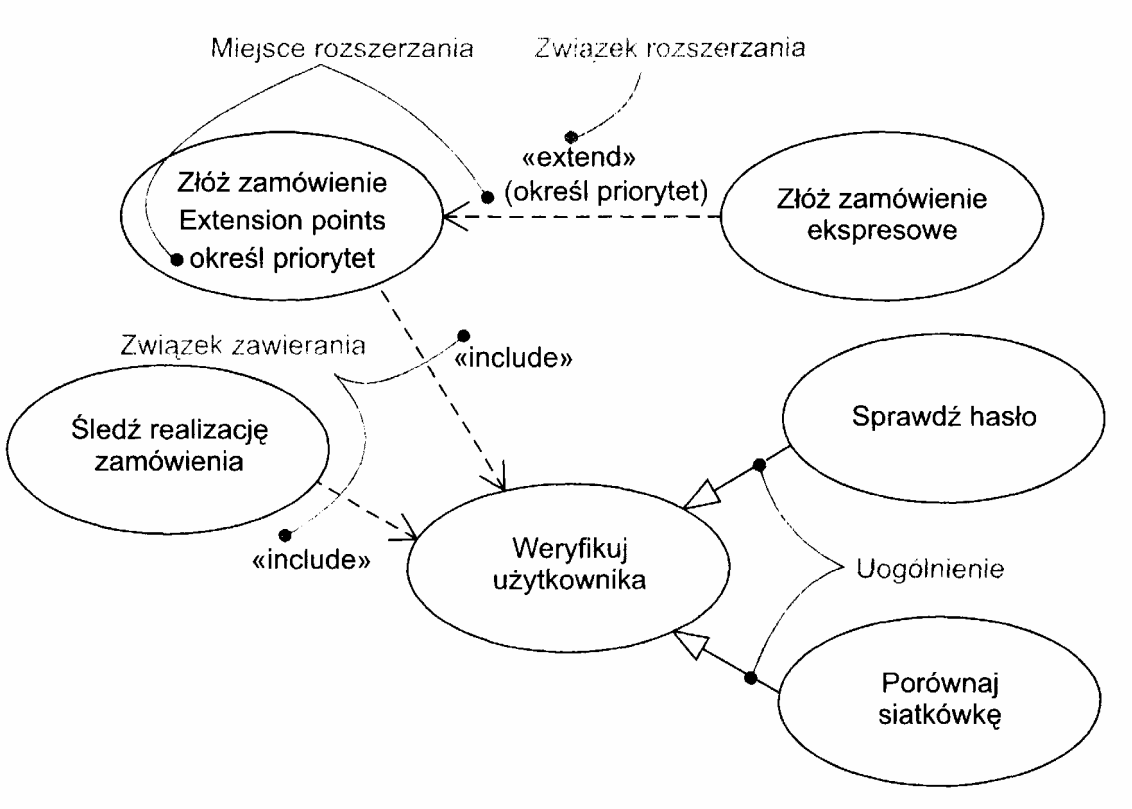
Rys. 16.4. Przypadek użycia i kooperacja

Uwaga: Znalezienie minimalnego zbioru dobrze zaprojektowanych kooperacji, które zadośćuczynią ciągom zdarzeń określonym we wszystkich przypadkach użycia systemu, jest zagadnieniem związanym z architekturą oprogramowania.

Porządkowanie przypadków użycia

Przypadki użycia można grupować w pakiety, tak jak klasy.
Przypadki użycia można uporządkować także przez zdefiniowanie uogólnień między nimi oraz związków zawierania i rozszerzania. Chodzi o wydzielenie

wspólnych fragmentów zachowania (przez usunięcie ich z obejmujących je przypadków użycia) lub wspólnych wariantów (przez wklejanie ich do rozszerzających je przypadków użycia).
Uogólnienie między przypadkami użycia jest jak uogólnienie między klasami. Oznacza, że przypadek użycia-potomek dziedziczy całe zachowanie i znaczenie po przypadku użycia-przodku. Potomek może dodać do odziedziczonego zachowania nowe elementy, a może też to zachowanie zupełnie zmienić. Potomek może zawsze zastąpić swego przodka (zarówno przodek, jak i potomek mogą mieć egzemplarze konkretne). W systemie bankowym występuje na przykład przypadek użycia Weryfikuj użytkownika, którego zadaniem jest sprawdzenie tożsamości klienta. Ten przypadek użycia ma dwóch szczegółowych potomków (Sprawdź hasło i Porównaj siatkówkę). Ich działanie jest podobne do Weryfikuj użytkownika i można je zastosować w zastępstwie Weryfikuj użytkownika. Zachowanie obydwu potomków jest jednak odmienne - pierwszy z nich wczytuje i sprawdza tekstowe hasło, a drugi porównuje siatkówkę użytkownika z zapamiętanym unikatowym wzorcem. Na rysunku 16.5 widać, że uogólnienie między przypadkami użycia jest przedstawiane jako linia ciągła zakończona zamkniętym, niewypełnionym grotem (taka sama jak w wypadku uogólnienia klas).
Związek zawierania między przypadkami użycia polega na tym. że bazowy przypadek użycia jawnie włącza zachowanie innego przypadku użycia w miejscu przez siebie określonym. Włączany przypadek użycia nigdy nie występuje samodzielnie - jego egzemplarze mogą być tylko częścią większego, zawierającego go przypadku użycia. Możesz myśleć o takim związku jak o zasysaniu zachowania włączonego przypadku użycia przez bazowy przypadek użycia.



Rys. 16.5. Uogólnienie, zawieranie i rozszerzanie

Związku zawierania używa się w celu umknięcia wielokrotnego opisywania tego samego ciągu zdarzeń. Wspólne zachowanie jest definiowane w odrębnym przypadku użycia, który jest następnie włączany przez bazowe przypadki użycia. Taki związek jest w istocie przykładem delegowania - pewien zbiór zobowiązań systemu jest specyfikowany w jednym ze składników modelu (zawieranym przypadku użycia), a inne części modelu (pozostałe przypadki użycia) włączają ten zbiór zobowiązań, ilekroć jest im potrzebny.
Związek zawierania obrazuje się w postaci zależności stereotypowanej jako include. Aby określić miejsce w ciągu zdarzeń, w którym bazowy przypadek użycia włącza zachowanie innego przypadku użycia, wystarczy napisać include, a potem nazwę włączanego przypadku użycia. Oto przykład zapisu ciągu zdarzeń przypadku użycia Śledź realizację zamówienia.

Główny ciąg zdarzeń: Pobierz i zweryfikuj numer zamówienia include (Weryfikuj użytkownika). Ustal stan każdej pozycji zamówienia i złóż raport użytkownikowi.

Związek rozszerzania między przypadkami użycia polega na tym, że bazowy przypadek użycia w sposób domniemany włącza zachowanie innego przypadku użycia w miejscu określonym pośrednio przez rozszerzający

przypadek użycia. Bazowy przypadek użycia może wystąpić samodzielnie, ale pod pewnymi warunkami jego zachowanie może być rozszerzone przez zachowanie innego przypadku użycia. Przypadek bazowy może być rozszerzony jedynie w ściśle określonych miejscach, nazywanych - co nie powinno nikogo dziwić - miejscami rozszerzania. Możesz myśleć o takim związku jak o dołączaniu zachowania przypadku rozszerzającego do przypadku bazowego.
Związek rozszerzania służy do modelowania fragmentów przypadku użycia postrzeganych przez użytkownika jako opcjonalne zachowanie systemu. Możesz oddzielić działania opcjonalne od wymaganych. Jednym z zastosowań rozszerzania jest wyodrębnienie podciągu zdarzeń, które zachodzą tylko pod pewnymi warunkami. Wreszcie możesz użyć tego związku do modelowania kilku ciągów zdarzeń, które mogą być dołączone w konkretnym miejscu, zależnie od jawnej interakcji z aktorem.
Związek rozszerzania obrazuje się w postaci zależności stereotypowanej jako extend. Miejsca rozszerzania bazowego przypadku użycia mogą być wymienione w dodatkowej sekcji jego symbolu. Są to etykiety, które mogą się pojawić w opisie ciągu zdarzeń bazowego przypadku użycia. Oto przykład zapisu ciągu zdarzeń przypadku użycia Złóż zamówienie.
Główny ciąg zdarzeń: include (Weryfikuj użytkownika). Pobierz od użytkownika listę pozycji jego zamówienia. (określ priorytet). Przekaż zamówienie do realizacji.
W tym przykładzie określ priorytet to miejsce rozszerzania. Przypadek użycia może mieć więcej niż jedno miejsce rozszerzania, z których każde może występować wielokrotnie. Miejsca rozszerzania są rozpoznawane po nazwie. W normalnych warunkach rozważany w przykładzie bazowy przypadek użycia będzie wykonany bez względu na priorytet zamówienia. Jeśli natomiast pojawi się egzemplarz zamówienia ekspresowego, ciąg zdarzeń bazowego przypadku użycia będzie taki jak opisany powyżej, z tym, że w miejscu rozszerzania (określ priorytet) zostaną dodatkowo wykonane czynności opisane w rozszerzającym przypadku użycia (Złóż zamówienie ekspresowe). Po zakończeniu tych czynności przetwarzanie bazowego przypadku użycia będzie kontynuowane. Jeśli istnieje więcej miejsc rozszerzania, w każdym z nich zostanie dołożony ciąg zdarzeń rozszerzającego przypadku użycia.

Uwaga: Porządkowanie przypadków użycia przez wydzielanie wspólnych czynności (za pomocą związków zawierania) i wyodrębnianie wariantów (za pomocą związków rozszerzania) to bardzo ważny etap procesu tworzenia prostego, zrównoważonego i przystępnego zbioru przypadków użycia systemu.

Inne właściwości

Przypadki użycia są klasyfikatorami, a zatem, podobnie jak klasy, mogą mieć atrybuty i operacje. Atrybuty możesz traktować jak wewnętrzne obiekty przypadku użycia, które są potrzebne do opisania jego zachowania widzianego z zewnątrz. Operacje z kolei możesz uważać za akcje systemu, które są potrzebne do opisania ciągu zdarzeń. Obiektów i operacji możesz użyć na diagramach interakcji do określenia zachowania przypadku użycia.
Ponieważ przypadki użycia są klasyfikatorami, możesz skojarzyć z nimi maszyny stanowe. Jest to jeszcze jeden sposób opisania zachowania reprezentowanego przez przypadek użycia.

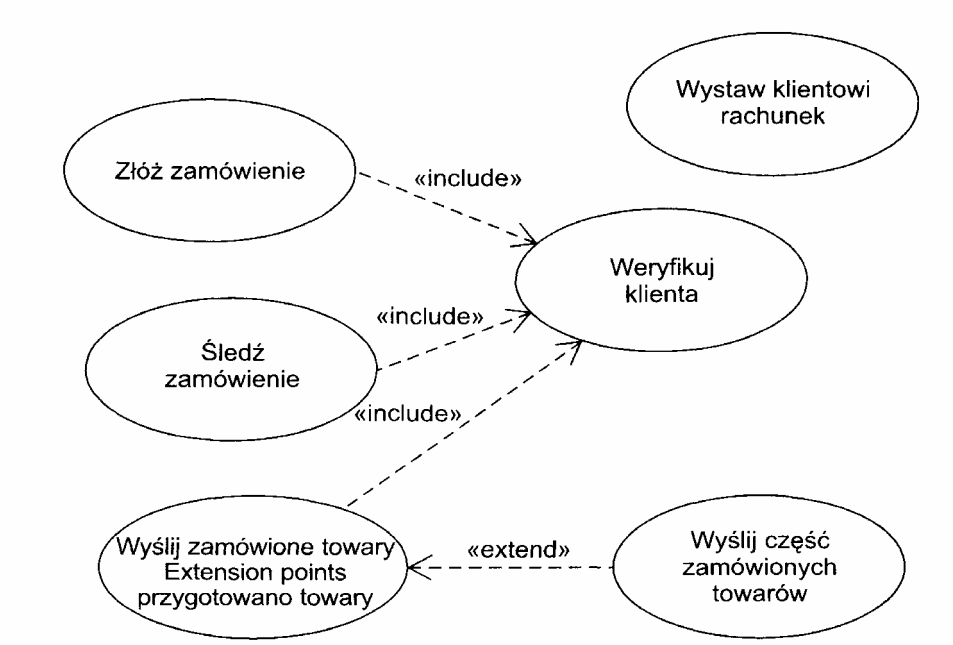
Przydatne techniki modelowania

Modelowanie zachowania bytu

Przypadków użycia używa się najczęściej do modelowania zachowania bytów - czy to systemu jako całości, czy to podsystemu, czy to klasy. Modelując działanie takiego elementu, należy skoncentrować się na tym, co on robi, a nie na tym, jak to robi.
Takie podejście do stosowania przypadków użycia w odniesieniu do bytów jest ważne z trzech powodów. Po pierwsze, modelując zachowanie bytu za pomocą przypadków użycia, eksperci w dziedzinie problemu mogą wyspecyfikować ten byt od zewnątrz na tyle precyzyjnie, żeby programiści mogli zająć się nim od środka. Przypadki użycia stanowią swego rodzaju środek komunikowania się między ekspertami w dziedzinie problemu, użytkownikami i programistami. Po drugie, dzięki zastosowaniu przypadków użycia poszczególne byty stają się bardziej przystępne i zrozumiałe dla programistów. System, podsystem lub klasa może być niezwykle złożonym bytem, obejmującym wiele operacji i innych składowych. Definiując przypadki użycia bytu, sprawiasz, że użytkownicy mogą się z nim zapoznać w taki sposób, w jaki prawdopodobnie będą z niego korzystać. Gdyby nie było przypadków użycia, użytkownicy sami musieliby dochodzić, jak dany byt stosować. Przypadki użycia umożliwiają twórcy bytu przestawienie zamierzonego trybu korzystania z niego. Po trzecie, przypadki użycia są podstawą do testowania każdego bytu w miarę rozwijania go. Testując każdy byt pod kątem przypadków użycia, cały czas weryfikujesz jego implementację. Przypadki użycia są nie tylko źródłem testów regresywnych. Gdy dorzucasz nowy przypadek użycia bytu, musisz ponownie rozważyć jego implementację, żeby sprawdzić, czy jest odporny na taką zmianę. Jeśli nie jest, musisz odpowiednio ulepszyć architekturę systemu.
Modelując zachowanie bytu, postępuj zgodnie z podanymi tu wytycznymi.

- Zidentyfikuj aktorów będących w interakcji z danym bytem. Kandydatami są między innymi grupy wymagające od niego pewnych działań niezbędnych do realizacji ich zadań, a także grupy potrzebne mu bezpośrednio lub pośrednio do spełnienia jego funkcji.
- Uporządkuj aktorów przez wyznaczenie ról bardziej ogólnych i bardziej szczegółowych.
- W wypadku każdego aktora rozważ podstawowe sposoby jego interakcji z danym bytem. Weź pod uwagę także te interakcje, które *zmieniają* stan bytu lub jego otoczenie albo są *związane z reakcją* na pewne zdarzenie.
- Rozważ również sytuacje wyjątkowe, w których dochodzi do interakcji każdego aktora z bytem.
- Usystematyzuj te zachowania w postaci przypadków użycia; skorzystaj ze związków zawierania i rozszerzania, aby wydzielić wspólne i wyróżnić wyjątkowe zachowania.

System sprzedaży detalicznej będzie w interakcji z klientem, który składa zamówienia i chce znać ich stan. System ten wysyła zamówione towary i wystawia klientowi rachunki. Na rysunku 16.6 przedstawiamy model zachowania tego systemu w postaci przypadków użycia (Złóż zamówinie Siedź zamówienie, Wyślij zamówione towary, Wystaw klientowi rachunek). Jak wynika z rysunku, wydzielono wspólne fragmenty tych działań (Weryfikuj klienta) i wyróżniono warianty (Wyślij część zamówionych towarów). Do każdego z tych przypadków użycia należy dodać specyfikację zachowania w postaci zwykłego tekstu, maszyny stanowej lub interakcji.



Rys. 16.6. Modelowanie zachowania bytu

W miarę rozrastania się modelu zauważysz, że wiele przypadków użycia da się pogrupować w porcje spójne pojęciowo i znaczeniowo. Aby uwzględnić je w modelu, wystarczy skorzystać z pakietów.

Rady i wskazówki

Modelując w UML przypadki użycia, nie zapomnij, że każdy z nich powinien reprezentować pewne odrębne i dobrze określone zachowanie systemu lub jego części. Dobrze zbudowany przypadek użycia

- opisuje pojedyncze, dobrze określone i możliwie niepodzielne zachowanie systemu lub jego części;
- uwzględnia wydzielone wspólne działania z innych przypadków użycia;
- uwzględnia wydzielone warianty dodane do innych przypadków użycia;
- opisuje ciąg zdarzeń tak, że jest on zrozumiały dla laika;
- jest opisany przez minimalną liczbę scenariuszy, określających jego podstawowe i opcjonalne znaczenie.

Gdy obrazujesz przypadek użycia w UML,

- pamiętaj, że powinien on ułatwiać zrozumienie działania systemu lub jego części w danym kontekście;
- ujawnij jedynie tych aktorów, którzy są w interakcji z tym przypadkiem użycia.

Diagramy przypadków użycia

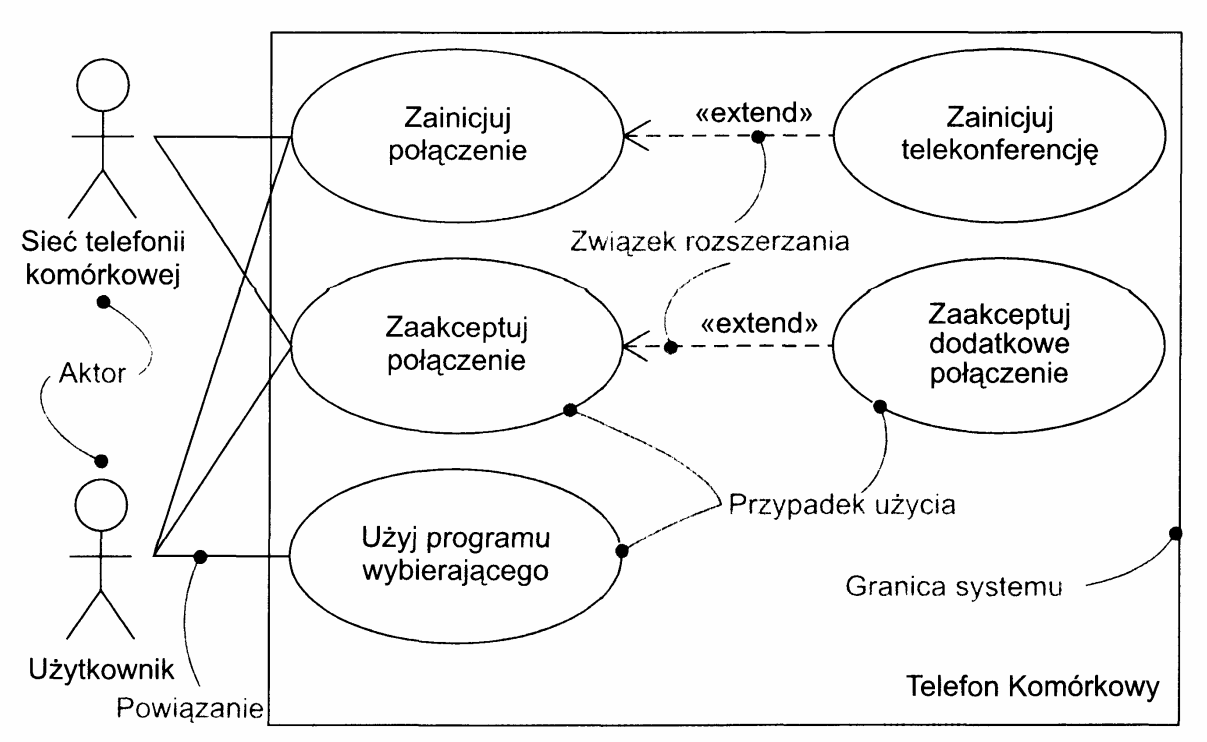
Poruszone tematy:

- Modelowanie otoczenia systemu
- Modelowanie wymagań stawianych systemowi
- Inżynieria do przodu i inżynieria wstecz

Diagram przypadków użycia to jeden z pięciu rodzajów diagramów UML, które służą do modelowania dynamiki systemu (pozostałe cztery to diagramy czynności, diagramy stanów, diagramy przebiegu i diagramy kooperacji). Diagramy przypadków użycia są głównym narzędziem do modelowania zachowania systemu, podsystemu lub klasy. Każdy z nich przedstawia zbiór przypadków użycia i aktorów oraz związki między nimi.
Diagramy przypadków użycia służą do modelowania perspektywy przypadków użycia systemu, a w tym do opisywania otoczenia systemu, podsystemu lub klasy lub określania wymagań dotyczących zachowań tych bytów.
Diagramy przypadków użycia są szczególnie przydatne w obrazowaniu, specyfikowaniu i dokumentowaniu zachowania bytu. Dzięki nim systemy, podsystemy i klasy stają się bardziej przystępne i zrozumiałe. Diagramy te, bowiem przedstawiają byt od zewnątrz. Są też przydatne w testowaniu gotowego systemu z wykorzystaniem inżynierii do przodu i w rozpoznawaniu funkcji działającego systemu z wykorzystaniem inżynierii wstecz.

Wprowadzenie

Wyobraź sobie, że ktoś wręczył ci skrzynkę. Na jednym z boków są przyciski i niewielki wyświetlacz ciekłokrystaliczny. Nic więcej o tej skrzynce nie wiesz, a w szczególności nie masz pojęcia, jak jej używać. Możesz oczywiście losowo naciskać przyciski i obserwować, co się dzieje. Trudno Ci jednak będzie dojść, co skrzynka robi i jak jej używać, jeśli nie poświęcisz sporo czasu na zbadanie jej metodą prób i błędów.
Podobnie jest z systemami informatycznymi. Przypuśćmy, że dostarczono Ci oprogramowanie i polecono go używać. Jeśli jest ono opracowane zgodnie z regułami, do jakich przywykłeś, z czasem sobie z nim poradzisz. Nigdy jednak nie zrozumiesz w ten sposób jego bardziej złożonego i subtelnego zachowania. W podobnej sytuacji znajdzie się programista, który ma wykorzystać otrzymany po kimś program lub zbiór komponentów. Trudno mu będzie dojść, jak tych elementów używać, jeśli nie opracuje modelu pojęciowego (koncepcyjnego) metod posługiwania się nimi.
W UML diagramy przypadków użycia służą do obrazowania zachowania systemu, podsystemu lub klasy w taki sposób, żeby użytkownicy mogli zrozumieć, jak z tego bytu korzystać, a programiści mogli go zaimplementować. Na rysunku 17.1 widać diagram przypadków użycia, którego można użyć do modelowania wspomnianej skrzynki. Wiele osób nazwałoby ją telefonem komórkowym.



Rys. 17.1. Diagram przypadków użycia

Pojęcia i ich definicje

Diagram przypadków użycia przedstawia zbiór przypadków użycia i aktorów oraz związki między nimi.

Ogólne właściwości

Diagram przypadków użycia to szczególny rodzaj diagramu. Ma te same właściwości co inne diagramy, to znaczy ma nazwę i zawartość, ale wyróżnia go specyfika tej zawartości.

Zawartość

Diagramy przypadków użycia zawierają na ogół

- przypadki użycia,
- aktorów,
- zależności, uogólnienia i powiązania.

Podobnie jak inne diagramy mogą zawierać także notatki i ograniczenia.
Na diagramach przypadków użycia mogą się znaleźć również pakiety, które służą do grupowania bytów modelu w większe porcje. Czasem na takim diagramie będziesz chciał umieścić także egzemplarze przypadków użycia - zwłaszcza wtedy, kiedy będziesz chciał zobrazować konkretny działający system.

Najczęstsze zastosowania

Diagramy przypadków użycia służą do obrazowania statycznych aspektów perspektywy przypadków użycia systemu. W perspektywie tej bierze się pod uwagę zachowanie systemu, to znaczy rozpoznawane z zewnątrz usługi, jakie system udostępnia bytom ze swego otoczenia.
Modelując statyczne aspekty perspektywy przypadków użycia systemu, diagramy przypadków użycia będziesz wykorzystywać do dwóch celów.

1. Modelowanie otoczenia systemu

To zadanie polega między innymi na wyznaczeniu granicy wokół całego systemu i na wskazaniu leżących poza nią aktorów, którzy wchodzą w interakcję z systemem. Diagramy przypadków użycia służą w tym wypadku do zdefiniowania aktorów i znaczenia ich ról.

2. Modelowanie wymagań stawianych systemowi

To zadanie polega na określeniu, co system powinien robić (z punktu widzenia jego otoczenia) - niezależnie od tego, jak ma to zrobić. Diagramy przypadków użycia służą w tym wypadku do zdefiniowania oczekiwanego działania systemu. Możesz do systemu podejść jak do czarnej skrzynki - znane jest otoczenie i sposób porozumienia się z bytami leżącymi poza nim, ale nie to, co dzieje się w jego wnętrzu.

Przydatne techniki modelowania

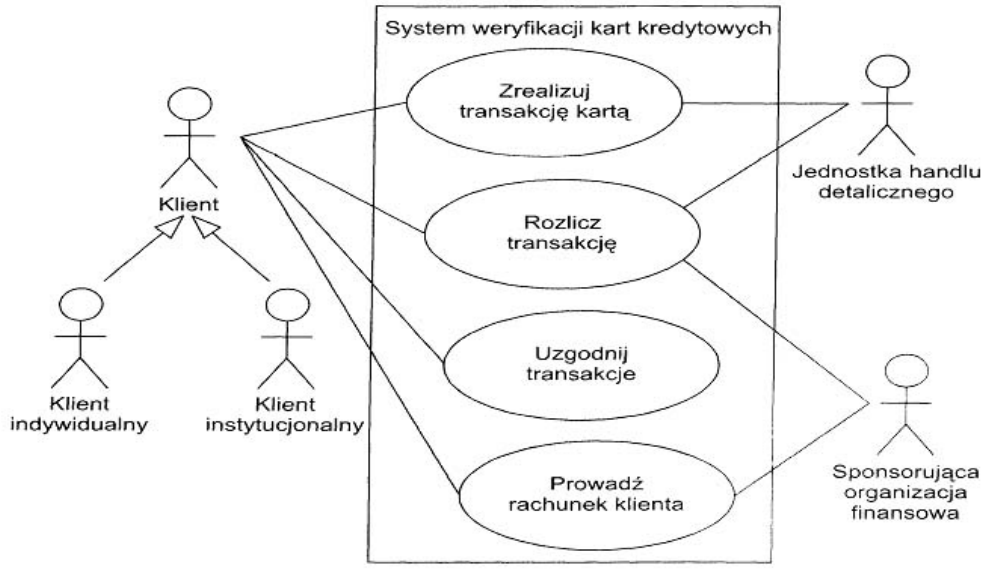
Modelowanie otoczenia systemu

W wypadku każdego systemu pewne elementy żyją wewnątrz niego, a pewne poza nim. Jeśli chodzi na przykład o system weryfikacji kart kredytowych, w jego wnętrzu znajdują się takie elementy, jak konta, transakcje i programy wykrywające oszustwa, natomiast poza nim - użytkownicy kart kredytowych i instytucje sprzedające takie karty. Elementy żyjące wewnątrz systemu są odpowiedzialne za wykonanie działań, których elementy zewnętrzne oczekują od systemu. Wszystkie byty istniejące poza systemem, a będące z nim w interakcji, stanowią jego otoczenie. Otoczenie definiuje środowisko, w którym system żyje.

Otoczenie systemu modelujemy za pomocą diagramu przypadków użycia, na którym kładzie się nacisk na aktorów działających wokół systemu. Decyzja, co ma być aktorem, jest bardzo ważna, ponieważ w ten sposób określasz klasę elementów będących w interakcji z systemem. Decyzja, co ma nim nie być, jest równie ważna (jeśli nawet nie ważniejsza), ponieważ zmusza środowisko systemu do uwzględnienia tylko tych aktorów, którzy są niezbędni do właściwego działania systemu.
Modelując otoczenie systemu, postępuj zgodnie z podanymi tu wytycznymi

- Zidentyfikuj aktorów działających wokół systemu. W tym celu rozważ, które grupy potrzebują pomocy systemu do realizacji swoich zadań, są niezbędne do realizacji funkcji systemu, są w interakcji z urządzeniami zewnętrznymi lub innymi systemami informatycznymi, wypełniają drugorzędne funkcje, takie jak zarządzanie i pielęgnacja.
- Uporządkuj podobnych aktorów za pomocą uogólnień.
- Aby zwiększyć czytelność modelu, dodaj - jeśli trzeba - stereotypy do aktorów.

„Zaludnij" tymi aktorami diagram przypadków użycia i zdefiniuj ścieżki komunikacyjne od każdego aktora do przypadków użycia systemu.
Na rysunku 17.2 przedstawiamy otoczenie systemu weryfikacji kart kredytowych. Na diagramie uwypuklono aktorów działających wokół tego systemu. Znajdziesz tu Klienta oraz dwa jego uszczegółowienia: Klient indywidualny i Klient instytucjonalny. Ci aktorzy to role odgrywane przez ludzi podczas interakcji z systemem. W otoczeniu systemu występują także aktorzy będący innymi instytucjami: Jednostka handlu detalicznego (to jej Klient płaci kartą za towary lub usługi) i Sponsorująca organizacja finansowa (działająca jako izba rozrachunkowa transakcji z użyciem karty kredytowej). W świecie rzeczywistym ci dwaj ostatni aktorzy będą prawdopodobnie systemami



Rys. 17.2. Modelowanie otoczenia systemu

Tę samą metodę można stosować do modelowania otoczenia podsystemów System na jednym poziomie abstrakcji traktuje się często jak podsystem większego systemu na innym poziomie abstrakcji. Modelowanie otoczenia podsystemu jest zatem przydatne, gdy tworzy się system złożony ze wzajemnie powiązanych systemów.

Modelowanie wymagań stawianych systemowi

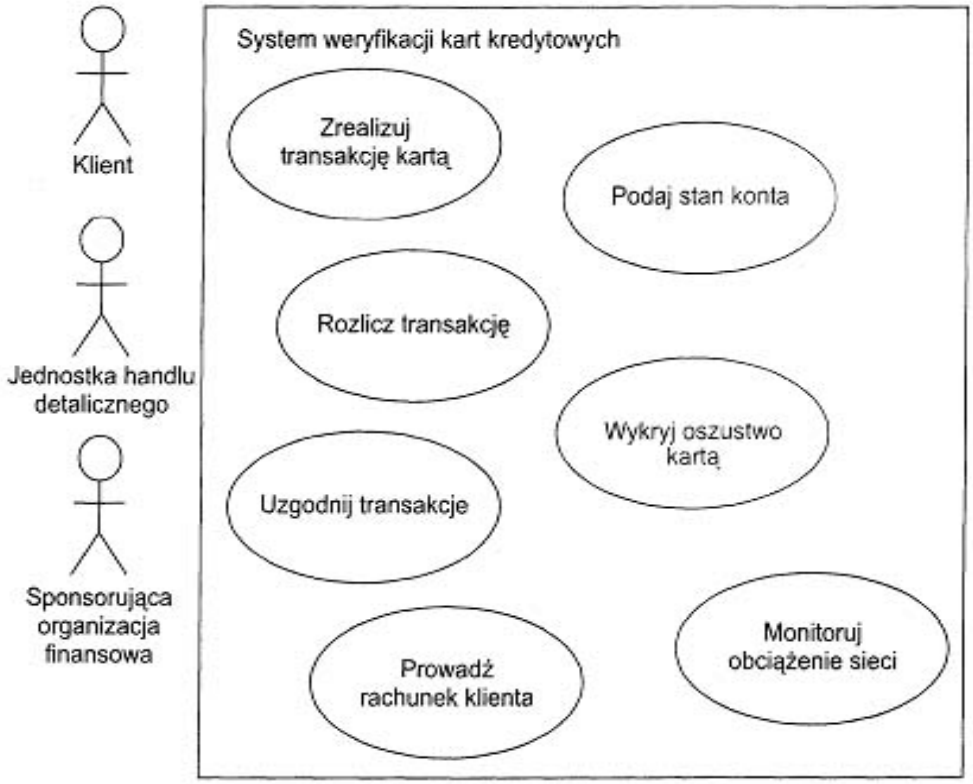
Wymaganie to element projektu, właściwość lub zachowanie systemu. Ustalając wymagania, spisuje się kontrakt między bytami z otoczenia a samym systemem, określający, co system ma robić. W większości wypadków nie przywiązuje się wagi do tego, jak system ma to robić, a jedynie do tego, że *ma* to robić. Poprawnie działający system spełnia wszystkie wymagania dokładnie, niezawodnie i w sposób przewidywalny. Przystępując do jego budowy, twórca systemu musi zacząć od uzgodnienia jego oczekiwanego zachowania, choć z pewnością wymagania będą się zmieniać w miarę przyrostowego i iteracyjnego implementowania systemu. Sytuacja użytkownika systemu jest podobna - znajomość zachowania systemu jest niezbędna do właściwego posługiwania się nim.
Wymagania mogą być zapisane w różnej postaci - od niestrukturalnego tekstu, do wyrażeń w języku formalnym. Większość wymagań funkcjonalnych, jeśl: nie wszystkie, może być określona w postaci przypadków użycia. Zdefiniowany w UML diagram przypadków użycia jest bardzo ważnym narzędziem do zarządzania tymi wymaganiami.

Modelując wymagania stawiane systemowi, postępuj zgodnie z podanymi tu wytycznymi.

- Określ otoczenie systemu. W tym celu zidentyfikuj okalających go aktorów.
- W wypadku każdego aktora rozważ działania, których on oczekuje lub wymaga od systemu.
- Zapisz te działania w postaci przypadków użycia.
- Wyłącz powtarzające się fragmenty działań i utwórz z nich nowe przypadki użycia, które będą dołączane przez inne przypadki użycia.
- Wydziel warianty działań i umieść je w nowych przypadkach użycia, które rozszerzają główne ciągi zdarzeń innych przypadków użycia.
- Uwzględnij te przypadki użycia, aktorów i związki między nimi na diagramie przypadków użycia.

Dodaj do nich notatki określające wymagania niefunkcjonalne. Może się zdarzyć, że będziesz musiał dołączyć niektóre z tych notatek do całego systemu.

Na rysunku 17.3 przedstawiamy diagram będący rozszerzeniem poprzedniego diagramu przypadków użycia. Chociaż pominięto na nim związki między aktorami a przypadkami użycia, dodano pewne nowe przypadki użycia, które są niedostrzegalne dla zwykłego klienta, ale stanowią zasadniczą część zachowania systemu. Diagram ten ma duże znaczenie, ponieważ ułatwia porozumienie użytkowników, ekspertów w dziedzinie problemu i programistów. Mogą skorzystać z niego do obrazowania, wyspecyfikowania, utworzenia i udokumentowania swoich decyzji dotyczących wymagań funkcjonalnych. Działanie Wykryj oszustwo kartą jest bardzo istotne z punktu widzenia Jednostki handlu detalicznego i Sponsorującej organizacji finansowej. Podobnie Podaj stan konta jest działaniem wymaganym przez różne instytucje w otoczeniu systemu.
Wymaganie modelowane przez przypadek użycia Monitoruj obciążenie sieci ma trochę inną naturę, ponieważ definiuje drugorzędne działania systemu niezbędne do jego niezawodnego i nieprzerwanego funkcjonowania.
Opisany tu sposób postępowania odnosi się również do modelowania wymagań stawianych podsystemom.



Rys. 17.3. Modelowanie wymagań stawianych systemowi

Inżynieria do przodu i inżynieria wstecz

Diagramy UML, w tym diagramy klas, diagramy komponentów, diagramy stanów, są w większości dobrymi kandydatami do zastosowania inżynierii do przodu i inżynierii wstecz, ponieważ przedstawiane na nich byty mają swoje odpowiedniki w systemie wykonywalnym. Diagramy przypadków użycia różnią się jednak nieco od pozostałych diagramów, gdyż raczej odzwierciedlają, a nie definiują implementację systemu, podsystemu lub klasy. Przypadki użycia opisują, jak dany byt się zachowuje, a nie, jak zachowanie bytu jest implementowane, a zatem nie mogą bezpośrednio podlegać inżynierii do przodu albo inżynierii wstecz.

Inżynieria do przodu to proces transformacji modelu na kod za pomocą przekształcenia do danego języka implementacji. Wynikiem tego procesu zastosowanego względem diagramu przypadków użycia pewnego bytu może być zbiór przeznaczonych dla niego danych testowych. Każdy przypadek użycia na diagramie określa pewien ciąg zdarzeń (i jego warianty), a te ciągi opisują, jak byt powinien się zachowywać, a zatem coś, co warto testować. Starannie opracowany przypadek użycia może nawet zawierać warunek wstępny i warunek końcowy, które mogą posłużyć - odpowiednio - za definicje stanu początkowego i kryterium powodzenia testu. Każdy

przypadek użycia na diagramie może być podstawą do przygotowania danych testowych wykorzystywanych zawsze wtedy, kiedy pojawia się nowa wersja bytu. Dzięki temu można sprawdzić, czy byt działa zgodnie z wymaganiami - zanim jeszcze inne składniki systemu zaczną na nim polegać.
Aby zastosować inżynierię do przodu do diagramu przypadków użycia, postępuj zgodnie z podanymi tu wytycznymi.

- Zidentyfikuj główny i nadzwyczajne ciągi zdarzeń każdego przypadku użycia na tym diagramie.
- Zależnie do założonego poziomu szczegółowości testowania wygeneruj dla każdego ciągu zdarzeń plan testu, używając warunku wstępnego jako stanu początkowego testu, a warunku końcowego jako kryterium powodzenia testu.
- Jeśli to konieczne, wygeneruj konstrukcję pomocniczą do testów dla każdego aktora będącego w interakcji z tym przypadkiem użycia. Aktorzy, którzy przekazują informacje bytowi lub podlegają działaniom tego bytu, mogą być symulowani lub zastąpieni przez swoje odpowiedniki ze świata rzeczywistego.
- Przeprowadź te testy, kiedy tylko pojawi się nowa wersja tego bytu Użyj w tym celu odpowiednich narzędzi.

Inżynieria wstecz to proces transformacji kodu na model za pomocą przekształcenia z danego języka implementacji. Automatyczne przeprowadzenie tego procesu względem diagramu przypadków użycia jest niezgodne z regułami sztuki. W wyniku, bowiem przejścia od specyfikacji zachowania elementu do implementacji tego zachowania dochodzi do utraty informacji. Badanie gotowego systemu i rozpoznawanie jego oczekiwanego zachowania można jednak przeprowadzić bez automatycznego wspomagania. Osiągnięte wyniki można potem utrwalić w postaci diagramu przypadków użycia. Trzeba to też zrobić, kiedy ma się do czynienia z nieudokumentowanym oprogramowaniem. Zdefiniowany w UML diagram przypadków użycia udostępnia po prostu standardowy i wyrazisty język do wyrażenia tego, co odkryłeś.
Aby przeprowadzić inżynierię wstecz do diagramu przypadków użycia, postępuj zgodnie z podanymi tu wytycznymi.

- Zidentyfikuj wszystkich aktorów będących w interakcji z systemem.
- W wypadku każdego autora zbadaj, w jaki sposób przebiega jego interakcja z systemem, jak zmienia on stan systemu lub jego środowisko, jak reaguje na zdarzenia.
- Prześledź ciągi zdarzeń w działającym systemie z punktu widzenia każdego aktora. Zacznij od ciągu głównego, a potem zajmij się ciągami nadzwyczajnymi.

- Zgrupuj ciągi pokrewne, deklarując odpowiedni przypadek użycia. Rozważ modelowanie wariantów za pomocą związków rozszerzania. a modelowanie wspólnych podciągów za pomocą związków zawierania.
- Zobrazuj tych aktorów i te przypadki użycia na diagramie przypadków użycia; określ związki między nimi.

Rady i wskazówki

Pracując nad diagramami przypadków użycia w UML. pamiętaj, że każdy z nich obrazuje statyczne aspekty perspektywy przypadków użycia systemu. Nie musisz przedstawiać ich wszystkich na jednym diagramie. Łącznie diagramy przypadków użycia reprezentują pełny statyczny obraz perspektywy przypadków użycia systemu, podczas gdy każdy z osobna uwypukla tylko jeden statyczny aspekt.
Dobrze zbudowany diagram przypadków użycia

- uwypukla jeden statyczny aspekt perspektywy przypadków użycia systemu;
- uwzględnia tylko te przypadki użycia i tych aktorów, którzy są niezbędni do zrozumienia tego aspektu.
- uwzględnia szczegóły odpowiednie do przyjętego poziomu abstrakcji, z dodatkami (np. miejsca rozszerzania), które są niezbędne do zrozumienia tego, na czym Ci zależy;
- nie jest zbyt ogólny, a zatem czytelnik nie zostanie wprowadzony w błąd co do istotnego znaczenia.

Gdy rysujesz diagram przypadków użycia,

- nadaj mu nazwę, która określa jego przeznaczenie;
- tak ułóż elementy, żeby zminimalizować liczbę przecinających się linii
- poukładaj zbliżone znaczeniowo działania i role tak, żeby były także blisko siebie na płaszczyźnie;
- skorzystaj z notatek i kolorów, żeby zwrócić uwagę czytelnika na to, na czym Ci zależy;
- postaraj się nie umieszczać na nim zbyt wielu rodzajów związków; jeśli sieć związków zawierania i rozszerzania jest bardzo złożona, przenieś związane z nimi byty na odrębne diagramy.