Technologia i rozwiązania

UML 2.0 w akcji

Przewodnik oparty na projektach

Poznaj język UML i wykorzystaj jego możliwości

- Opanuj podstawy języka
- Stwórz modele systemów biznesowych i informatycznych
- Zaplanuj integrację systemów przy użyciu języka UML

Patrick Graessle
Henriette Baumann
Philippe Baumann

Helion
PACKT PUBLISHING

Tytuł oryginału: Uml 2.0 in Action: A Project-based Tutorial

Tłumaczenie: Marek Pętlicki

ISBN: 978-83-246-4732-3

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. (32) 231-22-19, (32) 230-98-63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie?umlakc_ebook
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

Wstęp

Zaleta tej książki polega na tym, że jej treść opiera się na zagadnieniach praktycznych. Rozwój technik UML-a jest procesem bardzo dynamicznym, tak jak w przypadku większości innowacyjnych technologii informatyki biznesowej. Zainteresowanie producentów narzędzi, konsultantów i autorów jest z oczywistych przyczyn skupione na zwiększaniu zróżnicowania i szczegółowości przez wyrażanie indywidualnych opinii. Taka postawa skutkuje jednak rozmyciem granic kluczowych zagadnień, co w przypadku UML-a jest szczególnie niekorzystne. Sukces tej metodologii opiera się właśnie na jej prostocie i łatwości integrowania jej z innymi rozwiązaniami. Zrozumiała terminologia i precyzyjna metodyka tworzenia rozwiązań wynikają z obserwacji realnych problemów. Takie podejście zmusza do spojrzenia na problem pod różnymi kątami (co w przypadku każdego projektu uwzględnia przynajmniej punkty widzenia klienta oraz twórcy rozwiązań), dzięki czemu współpraca między stronami przedsięwzięcia jest bardziej produktywna w długim okresie. Nadmiernie rozbudowane metodologie nie sprzyjają produktywności, ponieważ są zrozumiałe jedynie dla ekspertów. Z tego powodu rzadko bywają wykorzystywane na szeroką skalę.

Początek książki dotyczy właśnie tej problematyki. Zaczynamy od zapoznania się z oczywistymi zaletami UML-a wraz z jego podstawowymi założeniami. Autorzy odważnie zawężają UML do zakresu, który według ich doświadczenia ma zastosowania praktyczne. Fundamenty wiedzy w tym zakresie autorzy budują w oparciu o subiektywny punkt widzenia, wynikający z doświadczeń związanych z wdrażanymi projektami. Dobór ten jest jednak słuszny, co mogą potwierdzić Czytelnicy, którzy mieli okazję brać czynny udział we wdrażaniu projektów systemowych. Opierając się na tym doświadczeniu, autorzy ocenili UML w sposób krytyczny. W efekcie powstał użyteczny podręcznik UML-a zawierający liczne praktyczne wskazówki dotyczące rozwiązywania problemów nieodłącznie związanych z projektami. Jednym z narzędzi proponowanych przez autorów są listy kontrolne (ang. *checklists*) służące weryfikacji wykonania postawionych celów. Całości przyświeca założenie, aby unikać nadmiaru metodologii. W związku z tym prezentowane przykłady są bardzo proste, dzięki czemu UML mógł zostać sprowadzony tylko do swoich niezbędnych elementów.

Lider projektu zainteresowany profesjonalnym modelowaniem, koordynacją i kontrolą projektów wykorzystujących metodologię UML oczywiście nie znajdzie w tej książce gotowych rozwiązań wszystkich potencjalnych problemów. Z pewnością znacznie ułatwi ona jednak dobór literatury i narzędzi pomocnych we wdrażaniu konkretnych projektów. Dla realisty nastawionego na rozwiązania praktyczne niniejsza książka będzie stanowiła znaczną pomoc w rozwiązywaniu codziennych problemów.

Prof. Dr Rainer Thome
Wydział Administracji Biznesowej i Informatyki w Biznesie
Uniwersytet w Würzburgu

O autorach

Patrick Grässle jest współzałożycielem i członkiem zarządu KnowGravity Inc. (*www.knowgravity.com*), firmy mającej swoją siedzibę w Zurychu. Firma ta specjalizuje się w konsultingu z dziedziny MDA i reguł biznesowych.

Patrick studiował informatykę i ekonomię na uniwersytecie w Zurychu. W 1986 roku zbudował swój pierwszy model systemu IT wykorzystujący analizę strukturalną i od tamtej pory nie zaprzestał prac nad modelowaniem systemów. W wielu projektach miał okazję używać UML-a. Wykorzystywał strukturalne i obiektowe metody specyfikowania systemów i świadczył konsultacje z zakresu ich stosowania. W latach dziewięćdziesiątych zorganizował pierwsze w Szwajcarii szkolenia z UML-a.

Obecnie jest mocno zaangażowany w rozwój UML-owych zagadnień określanych jako architektury zarządzane modelowo (ang. *Model Driven Architecture*) oraz podejścia oparte o reguły biznesowe (ang. *Business Rules Approach*). Znajduje jednak czas na prowadzenie szkoleń i konsultacji w dziedzinie UML-a.

Kontakt: *ck.graessle@knowgravity.com*.

Henriette Baumann jest współzałożycielką i członkinią rady nadzorczej firmy integratio GmbH (*www.integratio.com*) z siedzibą w Zurychu.

Henriette studiowała informatykę i ekonomię, a programowaniem i inżynierią oprogramowania zajmuje się od połowy lat osiemdziesiątych. Szczególnie mocno interesuje się zagadnieniem transformacji wymagań biznesowych w systemach programowych. W 1998 roku po raz pierwszy zastosowała UML w modelowaniu biznesowym i od tamtej pory wykorzystała ten język w kilku innych projektach. Obecnie jej uwaga skupia się głównie na zarządzaniu projektowym. Henriette zajmuje się też doradztwem w dziedzinach analizy biznesowej, inżynierii wymagań biznesowych i specyfikacji biznesowych z użyciem UML-a. Współpracuje głównie z firmami z sektora usług finansowych.

Kontakt: *henriette.baumann@integratio.com*.

Philippe Baumann jest współzałożycielem i członkiem rady nadzorczej firmy integratio GmbH (*www.integratio.com*) z siedzibą w Zurychu.

Philippe studiował informatykę na uniwersytecie w Hagen (Niemcy), a w połowie lat osiemdziesiątych zainteresował się rozwojem oprogramowania i integracją aplikacji. W 1998 roku zaczął stosować UML do modelowania integracji systemów i elektronicznej wymiany danych między firmami.

Obecnie jest menedżerem projektów i konsultantem w dziedzinie zagadnień technicznych oraz implementacji integracji oprogramowania z użyciem UML-a. Specjalizuje się również w dziedzinie implementacji i integracji oprogramowania biznesowego (jak ERP i RM) na licencji open source.

Kontakt: *philippe.baumanrv@integratio.com*.

O książce

W *OMG Specification* można znaleźć między innymi taki cytat:

> „Unified Modeling Language (UML) jest graficznym językiem służącym do wizualizacji, specyfikowania, konstruowania i dokumentowania artefaktów systemów wykorzystujących oprogramowanie komputerowe".

Kluczowym elementem dużych projektów programistycznych jest modelowanie, pomocne również w przypadku średnich i małych projektów. Język UML można wykorzystać do modelowania różnych systemów — programistycznych, biznesowych lub dowolnych innych. Zmiany i rozszerzenia, które weszły do wydania 2.0 tego języka, spowodowały, że UML znacznie lepiej nadaje się dziś do tych celów.

Tematyka książki

Autorzy tej książki starają się udowodnić, że dzięki językowi UML można tworzyć i odczytywać proste modele procesów biznesowych oraz modele specyfikacji projektowych. Większość książek na temat UML-a omawia go całościowo. Często jednak zdarza się, że z braku czasu, odpowiedniej wiedzy lub motywacji do zgłębiania wszystkich zagadnień na odpowiednim poziomie szczegółowości nie udaje się Czytelnikowi zrozumieć przedstawianego materiału w zakresie umożliwiającym praktyczne zastosowanie. Ta książka ma na celu wypełnienie tej luki. UML jest w niej zaprezentowany jedynie częściowo i w sposób uproszczony. Omawiamy jedynie te jego części, które uznaliśmy za przydatne z praktycznego punktu widzenia.

Rozdział 1. stanowi wprowadzenie do UML-a i omawia zalety stosowania go w charakterze języka do modelowania.

Rozdział 2. zawiera wstępne omówienie studium przypadku. Celem jest tu zastosowanie spójnych przykładów związanych z treścią poszczególnych rozdziałów. Ta część książki zawiera ponadto omówienie podstawowych zagadnień i koncepcji, takich jak modele, perspektywy,

diagramy, systemy informacyjne, metody i notacje. Modele i perspektywy tutaj wykorzystane mają pomóc Czytelnikowi w doborze odpowiedniego dla niego modelu do zbudowania specyfikacji wymagań.

Rozdział 3. obejmuje konstrukcję modeli systemów biznesowych. Można znaleźć tu uzasadnienie dla stosowania wielu perspektyw tego samego modelu systemu biznesowego oraz omówienie najważniejszych elementów każdej z perspektyw. W tym rozdziale znajduje się również instrukcja przygotowania diagramu przypadków użycia.

Rozdział 4. ilustruje proces tworzenia koncepcyjnego modelu systemu informatycznego przygotowanego z użyciem UML-a.

Rozdział 5. opisuje integrację systemu informatycznego ze środowiskiem, w którym ma być on użytkowany. Zostaną tu omówione sposoby modelowania komunikatów przesyłanych między różnymi systemami informatycznymi oraz procesów niezbędnych do wymiany tych komunikatów.

Konwencje

W książce wykorzystany został szereg konwencji typograficznych, które pozwalają nam wyodrębnić różne typy informacji.

Nowe i ważne terminy są zapisywane **pogrubioną czcionką**. Informacje wypisywane na ekranie, w menu i oknach dialogowych są wyróżnione w następujący sposób: „Po kliknięciu przycisku *Next* zostanie wyświetlona następna strona kreatora".

Wskazówki, sugestie lub ważne uwagi są wyróżnione przez umieszczenie tekstu w ramce.

ROZDZIAŁ 1

Wprowadzenie

Nadal jestem zmieszany, lecz teraz na znacznie wyższym poziomie abstrakcji.

Unified Modeling Language (**UML**) służy do opisywania systemów z użyciem słów i diagramów. Może być używany do modelowania różnorodnych systemów: informatycznych, biznesowych lub dowolnych innych. Warto tu wspomnieć o diagramach graficznych, między innymi o diagramach przypadków użycia z charakterystycznymi schematycznymi figurami ludzików. Diagramy tego typu nie są żadną nowością, natomiast jest nią unifikacja języków modelowania w oparciu o język UML. Sam język jest standaryzowany przez organizację **Object Management Group** (**OMG**), międzynarodowe stowarzyszenie zajmujące się promocją otwartych standardów wykorzystywanych w zastosowaniach technik obiektowych (*http://www.omg.org*).

Większość książek poświęconych językowi UML omawia go w całości. Z naszego doświadczenia wynika natomiast, że potencjalnym użytkownikom z reguły brakuje czasu, wiedzy lub motywacji do poznawania narzędzia o tej skali możliwości w pełnym ich zakresie. W takich przypadkach materiał nie jest w pełni zrozumiany i wykorzystywany w praktyce. Ta książka stanowi propozycję dla osób, które znajdują się właśnie w takiej sytuacji. Omawiamy w niej te aspekty UML-a, które okazały się najbardziej użyteczne. Dzięki temu każdy Czytelnik z niewielką dozą samozaparcia może w swojej pracy wykorzystać ten język.

Istnieje kilka argumentów przemawiających za wykorzystaniem UML-a w charakterze języka modelowania. Oto one:

- UML pozwala na unifikację terminologii i standaryzację notacji, co prowadzi do znacznego uproszczenia komunikacji pomiędzy wszystkimi zaangażowanymi w projekt grupami ludzi. Dzięki niemu możliwa jest bezproblemowa wymiana modeli systemu między departamentami lub firmami. Co więcej, projekty mogą być z łatwością przekazywane między zespołami projektowymi lub ich członkami.

- UML rozrasta się wraz ze wzrostem oczekiwań w stosunku do technik modelowania systemów. Mimo że jest on językiem modelowania o dużych możliwościach, znajomość z nim można spokojnie zacząć od prostych modeli; można w nim też tworzyć rozbudowane i szczegółowe projekty systemów. Jeśli standardowe możliwości UML-a nie wystarczają, można go rozbudować, używając do tego celu stereotypów.
- UML wykorzystuje szereg znanych i akceptowanych w branży metodologii. Oznacza to, że nie został on zaprojektowany od zera jako projekt akademicki — wręcz przeciwnie, miał stanowić odpowiedź na z gruntu praktyczne problemy i został oparty o istniejące języki modelowania. Taka geneza gwarantuje jego użyteczność i nadaje mu bardzo praktyczną wartość.
- UML jest językiem obsługiwanym przez wiele narzędzi.
- W przypadku opracowania w UML-u kilku konkurencyjnych projektów tego samego systemu informatycznego można je z łatwością porównywać.

Niniejsza książka jest oparta o UML w wersji 2.0, zgodnie ze specyfikacją Object Management Group (*OMG Unified Modeling Language: Superstructure, Version 2.0, Revised Final Adopted Specification*, październik 2004; specyfikacja ta jest dostępna na stronie *http://www.omg.org*).

W czasie pisania tej książki proces standaryzacji wersji 2.0 języka UML nie był jeszcze zakończony. Dzięki przyjętemu tu uproszczonemu podejściu do niego nie ma jednak podstaw do większych obaw, że zmiany wprowadzone w ostatecznej jego wersji będą miały znaczący wpływ na kwestie merytoryczne.

Nasza książka była pisana z myślą o naukowcach z dziedziny informatyki oraz osobach zaangażowanych we wdrażanie systemów informatycznych (analitykach, decydentach, użytkownikach i ekspertach). Pokazujemy w niej, że dzięki UML-owi proste modele procesów biznesowych i modele specyfikacji mogą być tworzone i odczytywane bez większego wysiłku. Nasze doświadczenie z projektami nasuwa nam następujące spostrzeżenia:

- Często tworzone są jedynie poszczególne elementy modelu.
- Przez większą część procesu wdrożenia można obejść się bez kompletnego modelu systemu.
- Na szkolenia z metodologii i języka modelowania z reguły poświęca się bardzo mało czasu.
- Samo modelowanie również jest zajęciem o niskim priorytecie podczas wdrażania systemu.

Niewiele projektów wykorzystuje pełny model UML, a nawet jeśli jest on używany, w bardzo małej części takich projektów język ten jest stosowany poprawnie. Istnieje natomiast wiele projektów, w których UML i inne języki modelowania są wykorzystywane jedynie w wąskim zakresie. Na początku prac nad projektem wiele osób podchodzi do niego entuzjastycznie, natomiast wraz ze zbliżaniem się terminów i nasileniem prac modelowanie i dokumentacja pierwsze padają ofiarą cięć i kompromisów.

Niestety, nie możemy w żaden sposób wyeliminować tego niekorzystnego zjawiska. Jednak pamiętając o takiej tendencji, staraliśmy się wybrać i zaprezentować w książce jedynie pewien podzbiór UML-a, dzięki czemu użytkownik będzie mógł korzystać z tego języka w sposób bardziej wydajny i poprawny, z niewielkim nakładem czasu poświęconego na naukę.

Doświadczenie uczy, że opanowanie kilku dobrze dobranych podstawowych elementów UML-a pozwala osiągnąć lepsze rezultaty niż nadgorliwe wykorzystywanie zbyt dużej liczby elementów tego języka. Pozwoliliśmy sobie zatem wybrać pewien podzbiór tych elementów. Oczywiście, wybór ten jest po części czysto subiektywny. Nie wspominamy wcale o wielu elementach UML-a, inne zaś objaśniamy dość powierzchownie. Nawet jeśli taki sposób nie jest zgodny z założeniami twórców tego języka, z naszej praktyki wynika, że jest on skuteczny.

UML 2.0 dzięki wprowadzonym zmianom i usprawnieniom jest znacznie lepiej od poprzednich wersji przystosowany do modelowania systemów biznesowych. Zwiększenie rozmiaru specyfikacji języka UML 2.0 dowodzi, że nie wystarczy omówienie poszczególnych elementów języka — należy również zademonstrować ich wykorzystanie w specjalistycznych zastosowaniach, takich jak integracja systemów czy hurtownie danych.

W tej książce staraliśmy się omówić różne zastosowania języka w oparciu o profile. W ten sposób przy okazji omawiania poszczególnych zastosowań języka UML 2.0 mamy okazję omówić nowe jego elementy. Każdy z najważniejszych profili omawiamy w osobnym rozdziale książki.

Struktura książki została ułożona w taki sposób, aby odpowiadać poszczególnym fazom pracy nad projektem. Na początku zajmujemy się modelowaniem systemu biznesowego (rozdział 3., „Modelowanie systemów biznesowych"). Następnie przechodzimy do specyfikowania systemu informatycznego, który ma być osadzony w systemie biznesowym (rozdział 4., „Modelowanie systemów informatycznych"). Na końcu opisujemy techniki integracji systemu IT w jego środowisku docelowym (rozdział 5., „Modelowanie integracji systemów"). Każdy z tych rozdziałów stanowi niezależną całość. Czytelnik może zapoznać się z tymi, które w danej chwili będą mu najbardziej potrzebne. Niezależnie od tego na początku należy jednak przeczytać rozdział 2., noszący tytuł „Podstawy i tło historyczne". Stanowi on wstęp do omawianego studium przypadku. Ten rozdział omawia również podstawowe terminy i koncepcje wykorzystywane w kolejnych rozdziałach.

Cała książka oparta jest o studium przypadku, dzięki czemu teoretyczna wiedza na temat UML-a może być pokazana na praktycznych przykładach. To studium przypadku spełnia rolę ilustracji zastosowań UML-a, ponieważ na konkretnym przykładzie często udaje się wytłumaczyć coś lepiej niż za pomocą abstrakcyjnej definicji. Czytelnik może dzięki temu „poczuć" UML-a. Oczywiście, nie mieliśmy możliwości przeprowadzenia pełnego studium przypadku i omówienia wszystkich diagramów i ich elementów. Nie jest to również konieczne do tego, aby Czytelnik zrozumiał język UML. Studium przypadku (obsługa pasażerów na fikcyjnym „lotnisku UML") nie zawsze odwzorowuje rzeczywistą specyfikę działania lotniska: w wielu miejscach tę analizę znacznie uprościliśmy. Nie ma to jednak negatywnego wpływu na jakość przekazywanej wiedzy dotyczącej podstaw UML-a.

Od Czytelnika tej książki nie oczekujemy wcześniejszej znajomości UML-a ani innych metodologii programowania zorientowanego obiektowo. Oczekujemy jednak podstawowej wiedzy na temat procedur wdrożeniowych systemów informatycznych.

W tym miejscu chcemy podziękować członkom załogi lotniska Unique Zurich Airport, którzy cierpliwie i kompetentnie pomagali nam zrozumieć szczegóły techniczne ich pracy przydatne w konstruowaniu naszego studium przypadku. Dziękujemy wszystkim, którzy pomogli nam w przygotowaniu i poprawianiu tej książki. Szczególnie chcemy podziękować redaktorce naszego niemieckiego wydania, Judith Stevens-Lemoine, i jej zespołowi w wydawnictwie Galileo za kompetentną i przyjacielską współpracę. Bardzo użyteczne okazały się również słowa krytyki i sugestie od naszych Czytelników oraz przyjaciół z firm Integratio i KnowGravity. Na końcu, lecz z nie mniejszą wdzięcznością, chcemy podziękować wszystkim, który zakupili egzemplarze pierwszych dwóch wydań tej książki w języku niemieckim. Dzięki Wam możliwe stało się wydrukowanie kolejnego wydania tej książki.

Henriette i Philippe Baumann oraz Patrick Grässle

ROZDZIAŁ 2

Podstawy i tło historyczne

Wiele zagadnień omawianych w kolejnych rozdziałach opiera się na pewnych podstawach, które wprowadzimy w tym rozdziale.

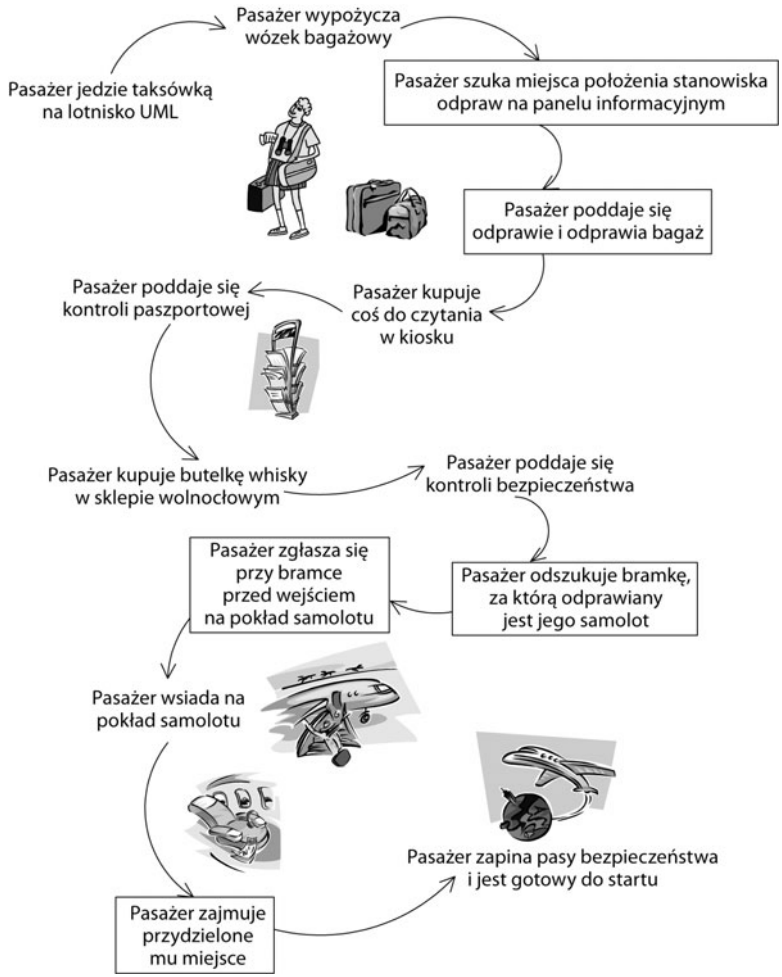
2.1. Wprowadzenie do studium przypadku

Do naszych analiz wykorzystamy lotnisko — nazwiemy je „lotniskiem UML". Dzięki temu każdy Czytelnik, który miał okazję lecieć samolotem, bez problemu zrozumie wszystkie przykłady.

Rozważania ograniczymy do tych obszarów pracy lotniska, z którymi podróżny może zetknąć się na co dzień, zatem największą uwagę poświęcimy procedurom **odprawy pasażera** i jego **wejścia na pokład** samolotu. Rysunek 2.1 ilustruje, w jaki sposób da się wyróżnić usługi związane z pasażerami spośród innych związanych z pracą lotniska. Rysunek ten przedstawia schemat operacji, w których musi wziąć udział pasażer, zanim znajdzie się na swoim fotelu w samolocie i zapnie pasy, a samolot będzie gotów do startu. Nie wszystkie etapy działań pasażera wiążą się jednak z interesującymi nas obszarami pracy lotniska. Te, które nas interesują, zostały na rysunku ujęte w ramkę.

Tego typu sekwencja nosi nazwę **scenariusza** (ang. *scenario*). Scenariusz przedstawiony na rysunku jest tylko jednym z możliwych, ponieważ mogą wystąpić liczne sytuacje odbiegające od tego schematu, na przykład takie:

- pasażer posiada jedynie bagaż podręczny;
- pasażer nie dokonuje zakupów w kiosku z prasą;
- pasażer przybył na lotnisko za późno i jest zmuszony poddać się przyspieszonej odprawie;



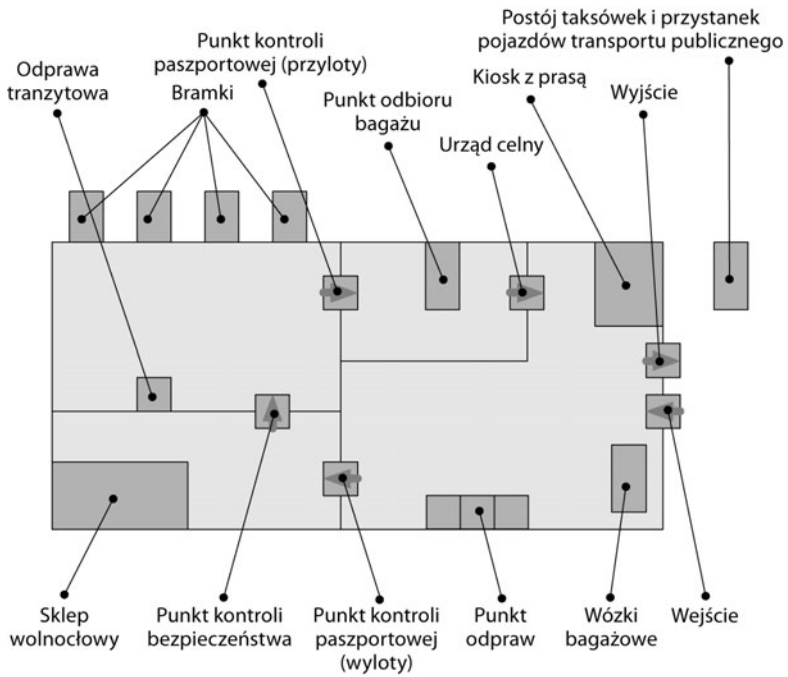
Rysunek 2.1. Studium przypadku — pasażer wybiera się w podróż samolotem

- pasażer zgubił swój bilet;
- pasażer przyleciał innym samolotem i musi po prostu dokonać przesiadki, czyli pozostaje w obszarze tranzytowym;
- pasażer poddał się odprawie, ale zasypia w poczekalni i pomimo wielokrotnego wywoływania przez megafon nie zgłasza się, więc samolot odlatuje bez niego;
- pasażer zostaje zatrzymany przy odprawie paszportowej, ponieważ jego paszport jest nieważny.

Warto zastanowić się, które z powyższych przypadków mają rzeczywiste znaczenie podczas procedury odprawy pasażerów oraz czy można sobie wyobrazić inne sytuacje, ważniejsze od wymienionych.

Schemat hipotetycznego lotniska przedstawiony na rysunku 2.2 powinien być pomocny przy wyobrażaniu sobie zdarzenia z naszego studium przypadku. Z procedurą odprawy łączy się także wiele innych obszarów pracy lotniska, sąsiadujących z tym dotyczącym samego pasażera. Wśród nich można wskazać działanie:

- kas biletowych;
- kiosku z prasą;
- sklepu wolnocłowego;
- stanowiska odprawy paszportowej;
- kontroli lotu;
- stanowiska informacyjnego;
- stanowiska obsługi i transportu bagażu.



Rysunek 2.2. Schemat lotniska UML

Procedura odprawy pasażerów wymaga wymiany danych z tymi obszarami. Konieczna jest też komunikacja z innymi sferami pracy lotniska. Będą one omawiane przy okazji definiowania modelu biznesowego oraz integracji systemu. Studium przypadku będzie rozbudowywane w kolejnych rozdziałach książki przy okazji wprowadzania kolejnych zagadnień.

Lotnisko UML jest niewielkim portem lotniczym, więc i nasze studium przypadku nie będzie nazbyt skomplikowane.

Celem tego studium przypadku jest dostarczenie spójnego przykładu zastosowania UML-a. W każdym z kolejnych rozdziałów model będzie rozszerzany o kolejne kwestie. Na początek należy jednak wyjaśnić kilka szczegółów:

- Bilet lotniczy składa się z samego biletu oraz z kilku dodatków (do czterech). **Bilet** ma postać książeczki, w której każdemu etapowi podróży jest poświęcona osobna sekcja. Na przykład może on zawierać jeden kupon na lot z Zurychu do Frankfurtu, drugi na lot z Frankfurtu do Londynu, a trzeci na powrót z Londynu do Zurychu. Podczas każdej odprawy odpowiedni kupon zostaje zamieniony na odpowiednią kartę pokładową. Sam bilet natomiast pozostaje w posiadaniu pasażera.
- Należy rozróżnić pojęcia lot i numer lotu. **Numer lotu** może na przykład mieć postać LH435 lub LX016. Numer ten określa regularny przelot z lotniska początkowego do lotniska przeznaczenia. **Lot** natomiast określa się przy użyciu numeru lotu i daty, na przykład LH435 26 czerwca 2006 roku. Można zatem powiedzieć, że lot to rzeczywista realizacja numeru lotu. Lot może być również odwołany z powodu niekorzystnych warunków pogodowych. Numer lotu jest stosowany przez cały okres regularnego wykonywania przez linię lotniczą przelotów na określonej trasie.
- Należy rozróżnić także trzy formy odprawy:
  - **zwykła odprawa** z bagażem w punkcie odpraw;
  - **ekspresowa odprawa** bez bagaży w specjalnym punkcie odpraw;
  - **automatyczna odprawa** bez bagaży.

2.2. Modele, perspektywy i diagramy

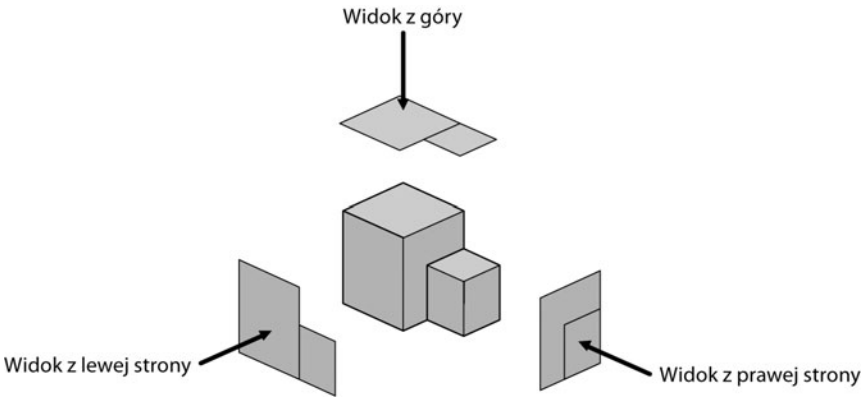
2.2.1. Czym jest model?

Modele są budowane w kontekście biznesowym lub informatycznym w celu lepszego zobrazowania istniejących lub przyszłych systemów. Należy pamiętać, że model nigdy nie będzie w pełni odpowiadał rzeczywistości. Modelowanie jest nieodłącznie związane z **podkreślaniem** i **pomijaniem**: podkreślane są szczegóły istotne, pomijane zaś szczegóły zbędne. Jak jednak można odróżnić jedne od drugich? Nie ma, niestety, uniwersalnej odpowiedzi na to pytanie. Odpowiedź zależy od tego, **po co** model jest tworzony i **kto** ma być jego odbiorcą.

Zastanówmy się, jakie szczegóły są podkreślane, a jakie pomijane w następujących modelach:

- model samochodu do badań aerodynamicznych
- model budynku w skali 1:50
- plan trasy metra
- mapa
- schemat organizacyjny

Im więcej informacji wprowadzi się do modelu, tym bardziej stanie się on skomplikowany, a przez to trudniejszy do zrozumienia. Na przykład mapa Europy zawierająca informacje polityczne, geologiczne, demograficzne i związane z transportem będzie z pewnością bardzo trudna do odczytania. Najprostszym rozwiązaniem tego problemu będzie z pewnością poświęcenie poszczególnym zagadnieniom osobnych, wyspecjalizowanych map. Z tego powodu analizowany obiekt jest często ujmowany w różnych **perspektywach**. Perspektywy te są ze sobą powiązane na wiele sposobów. Jeśli jedna z nich ulega zmianie, pozostałe również należy odpowiednio zmodyfikować. Jeśli na przykład Holandia powiększy swój obszar o kolejny fragment lądu odebrany Morzu Północnemu, należy odpowiednio zmodyfikować wszystkie wyspecjalizowane mapy tego obszaru.



Rysunek 2.3. Różne perspektywy postrzegania tego samego obiektu

Podobne zasady dotyczą modelu budynku. Jeśli do istniejącego budynku zostanie dodane nowe skrzydło, należy odpowiednio zmodyfikować poszczególne ujęcia, w tym rzuty pięter, poszczególne rzuty perspektywiczne oraz trójwymiarowy, drewniany model. Zasadę tę w sposób schematyczny ilustruje rysunek 2.3. W podrozdziale 2.4 („Modele studium przypadku") bliżej zajmiemy się powiązaniami między modelami systemu analizowanego w tej książce. Poszczególne perspektywy w każdym z modeli zostaną omówione w rozdziałach 3., 4. i 5.

2.2.2. Do czego potrzebne są modele?

Model systemu realizuje następujące funkcje:

- **Komunikacja** między elementami systemu. Aby system działał poprawnie, wszystkie jego elementy muszą być skonstruowane zgodnie z pewnym zasadniczym, jednolitym założeniem. Szczególnie ważne jest to, aby każdy użytkownik systemu rozumiał zastosowaną terminologię, żeby klienci zgadzali się z przyjętymi założeniami, a twórcy systemu rozumieli te założenia w taki sam sposób. Równie istotna jest kwestia tego, by decyzje były później tak samo czytelne, jak w momencie ich podejmowania.

- **Wizualizacja** wszystkich procedur z punktu widzenia klientów, ekspertów i użytkowników. Wszystkie skumulowane procedury związane z systemem powinny być zaprezentowane w taki sposób, aby każdy zainteresowany mógł je zrozumieć. Wiemy jednak z doświadczenia, że często podczas prezentacji projektu w postaci diagramu (zamiast opisu słownego) odbiorca odruchowo reaguje negatywnie. Ważne jest zatem pokonanie tej reakcji. Jest ona bowiem zwykle spowodowana strachem przed nieznanym, a diagramy na pierwszy rzut oka mogą sprawiać wrażenie czegoś skomplikowanego. Ta książka zawiera wskazówki dotyczące również sposobu analizy diagramów.
- **Weryfikacja** faktów pod kątem kompletności, spójności i poprawności. Szczególnie jasna definicja wzajemnych związków umożliwia zadawanie pytań i udzielanie odpowiedzi. Przy każdym prezentowanym diagramie będziemy przedstawiać stosowną listę pytań.

Proponujemy Czytelnikowi, aby odpowiedział sobie na następujące pytania:

- Kiedy ostatnio podczas dyskusji nad systemem okazało się, że cele klienta i dostawcy są zupełnie sprzeczne?
- Kiedy ostatnio miałeś wrażenie, że po raz kolejny omawiasz ten sam temat?
- Kiedy ostatnio żałowałeś, że podczas ustalania ważnego konsensusu w dyskusji nie miałeś włączonego dyktafonu?

2.2.3. Cel i grupa docelowa modelu

W życiu codziennym często okazuje się, że wyniki ciężkiej pracy nad projektem zwyczajnie grzęzną na czyimś biurku pod zwałami papieru. Można zapytać, dlaczego tak się dzieje. Efekt końcowy prac nad modelem jest uzależniony od dwóch podstawowych czynników: odbiorcy modelu (**dla kogo?**) i rzeczywistego jego przeznaczenia (**po co?**). Jeśli obydwa te czynniki nie zostaną ustalone w sposób jednoznaczny, istnieje realne ryzyko, że w wyniku prac powstaną modele niezawierające informacji ważnych dla odbiorcy. A jeśli model nie podkreśla tego, co istotne, i nie pomija tego, co zbędne, jego odbiorca z pewnością uzna, że jest on bezużyteczny.

Aby zdefiniować grupę docelową modelu, należy odpowiedzieć na następujące pytania:

- Jakiego poziomu **zaawansowania biznesowego** należy odczekiwać od odbiorców? Czy należy założyć, że mają oni podstawową wiedzę na temat obiektu, czy też model ma zawierać fundamentalne szczegóły dotyczące modelowanych zdarzeń i procesów?
- Jaki poziom **szczegółowości** jest potrzebny odbiorcom? Jaki poziom komplikacji jest dopuszczalny w modelu? Jeśli procesy i systemy będą podatne na ciągłe zmiany, bardzo szczegółowy model może być mało realistyczny. Dzieje się tak z tego powodu, że w większości przypadków nie ma możliwości stałego utrzymania takiego modelu w aktualnej postaci. Mniej szczegółowy model wymaga mniejszych nakładów pracy, jeśli chodzi o jego utrzymanie, jest jednak oczywiście mniej dokładny.

- Ile **czasu** grupa docelowa może poświęcić na analizę i interpretację modelu? Aby uniknąć przysypania modelu stosem papierów, należy zadbać o odpowiedni poziom szczegółowości i komplikacji — w przeciwnym wypadku nikt nie będzie zaprzątał sobie nim głowy.
- Jakim **językiem** posłużyć się przy definicji modelu? Czy grupa docelowa zrozumie biznesową terminologię techniczną? Czy zrozumie terminologię informatyczną? Oto prosty przykład: jeśli na butelce z wodą widnieje etykieta z napisem „woda", mamy pewność, że prawie każdy, kto ją przeczyta, domyśli się, jaka jest zawartość naczynia. Jeśli jednak na etykiecie napisać „H_2O", to nawet pomimo technicznej poprawności tego opisu mamy pewność, że przekaz dotrze do węższej grupy odbiorców, na przykład pracowników laboratorium chemicznego. Takie rozwiązanie ma jednak swoje zalety: odbiorca uzyskuje dodatkową, precyzyjną informację na temat składu chemicznego substancji. Z tego przykładu wynika wniosek, że „etykiety" (czyli nomenklaturę) należy zawsze uważnie dobierać pod kątem założonej grupy odbiorców przekazu.
- Jaki zastosować poziom **abstrakcji**? Im mniej abstrakcyjny jest model, tym bardziej czytelny i zrozumiały będzie dla odbiorcy. Dzieje się tak z tego powodu, że niższy poziom abstrakcji jest bliższy percepcji użytkownika i stosowanemu przez niego językowi. Z drugiej strony modele o wysokim poziomie abstrakcji są bardziej uniwersalne i łatwiej je przełożyć na system informatyczny. Takie modele również łatwiej analizować pod kątem poprawności formalnej. Specjaliści do spraw informatyki z reguły preferują modele abstrakcyjne. Użytkownicy natomiast często zupełnie gubią się w zetknięciu z modelami tego typu.

Wskazówki praktyczne

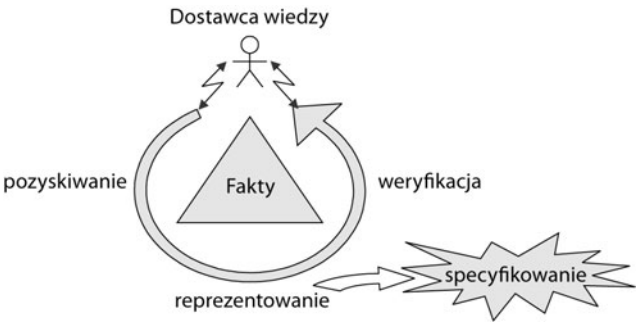
Przy wyborze poziomu abstrakcji, czytelności i szczegółowości modelu często jesteśmy zmuszeni do **kompromisów**. Można oczywiście stworzyć kilka modeli różniących się poziomem formalności i szczegółowości, optymalizując je dla różnych grup odbiorców. W ten sposób komunikacja między twórcami modelu, klientami, użytkownikami i deweloperami będzie odbywać się sprawniej. Należy jednak unikać przesady — model powinien być dostosowany do grupy odbiorców i założonego wykorzystania.

Analiza i **wzorce projektowe** są przykładami modeli opisujących powszechnie stosowane techniki projektowania i modelowania. Należy, o ile to możliwe, poszukiwać tego typu modeli w internecie, książkach[1], prasie fachowej. Można też po prostu konsultować się ze współpracownikami.

[1] Jako przykład można tu wymienić książkę Martina Fowlera, *Analysis Patterns: Reusable Object Models*, Addison-Wesley, 1999[0].

2.2.4. Proces analizy

Rysunek 2.4 przedstawia proces analizy składający się z faz **pozyskiwania**, **reprezentowania** i **weryfikowania** faktów.



Rysunek 2.4. Proces analizy

Ten etap jest realizowany przez analityka. W procesie analizy powstaje specyfikacja wynikająca z modelu i innych dostępnych dokumentów. Analityk współpracuje z osobami posiadającymi wiedzę na temat modelowanego systemu: klientami, użytkownikami i ekspertami w analizowanej dziedzinie.

- Fakty są **pozyskiwane** w wyniku współpracy pomiędzy analitykiem a ekspertami. Ta współpraca polega na tym, że eksperci są dostawcami wiedzy w danej dziedzinie, a analitycy są dostawcami wiedzy metodologicznej.
- Fakty są **reprezentowane** w postaci diagramów i dokumentów przygotowywanych z reguły przez analityka.
- Fakty są **weryfikowane** wyłącznie przez dostawców wiedzy w danej dziedzinie, ponieważ tylko oni mogą zdecydować, czy są one poprawne. Weryfikacja faktów jest etapem absolutnie koniecznym. Bez niego twórcy systemu otrzymają piękne diagramy, lecz będzie istniało ryzyko, że prezentowane informacje są nieprawdziwe. Innymi słowy, praca nad modelem bez weryfikacji jego poprawności jest zupełnie bezużyteczna!

Porady praktyczne

Niemożliwe jest stworzenie i zweryfikowanie użytecznego modelu bez opanowania technicznych podstaw analizowanego problemu. Gdzie szukać **dostawców wiedzy**, którzy będą służyć informacjami na temat modelowanego systemu? Z naszych doświadczeń wynika, że szczególnie dobre wyniki można uzyskać, korzystając ze współpracy z grupami osób takich, jak:

- uczestnicy i kontrolerzy procesów biznesowych;
- użytkownicy systemów informatycznych o funkcjonalności zbliżonej do modelowanego systemu lub związanej z nim;

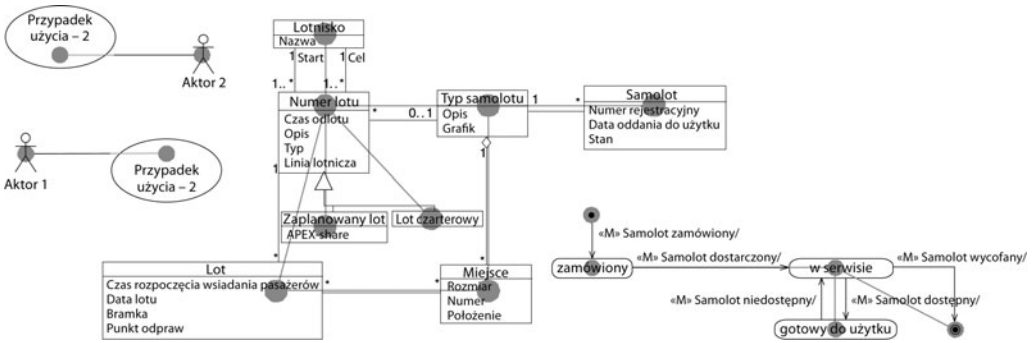
- klienci, którzy często bywają krytycznymi, lecz również kreatywnymi dostawcami wiedzy;
- partnerzy biznesowi;
- eksperci w analizowanej dziedzinie;
- zarząd firmy;
- niezależni obserwatorzy.

W procesie analizy i poznawania procesów biznesowych pomocne bywają następujące techniki:

- obserwacja pracowników przy pracy;
- branie udziału w analizowanych procesach biznesowych;
- przyjęcie roli uczestnika zewnętrznego (na przykład klienta);
- ankiety;
- przeprowadzanie wywiadów;
- organizowanie burz mózgów z udziałem wszystkich zaangażowanych grup;
- prowadzenie dyskusji z ekspertami;
- analiza istniejących formularzy, dokumentacji, specyfikacji i narzędzi pracy;
- opisywanie struktury organizacyjnej i zasad przepływu informacji (diagramy organizacyjne itp.).

2.2.5. Diagramy w roli perspektyw

Każdy diagram UML odgrywa rolę jednej **perspektywy** modelu systemu. W zależności od typu diagramu na pewne zagadnienia kładzie się nacisk, inne zaś są pomijane. Większość diagramów występuje w postaci graficznej (przykład na rysunku 2.5), co znaczy, że składają się one z elementów graficznych połączonych liniami:

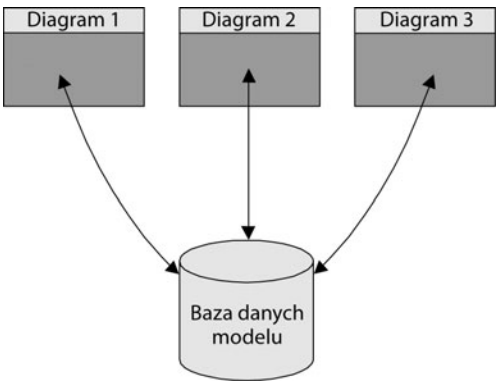


Rysunek 2.5. Diagram graficzny

W celu odczytu diagramów należy zapoznać się ze znaczeniem elementów i linii. Poszczególne rodzaje diagramów oraz występujące w nich symbole będziemy omawiać stopniowo w kolejnych rozdziałach książki.

Nawet w przypadku **komputerowych narzędzi wspomagania inżynierii oprogramowania** (ang. *Computer Aided Software Engineering* — CASE) diagramy UML są traktowane jako perspektywy. Narzędzia te do przechowywania informacji o modelu wykorzystują bazy danych. Każdy diagram (w roli perspektywy) zawiera część informacji na temat całości. Dzięki temu narzędzia CASE pomagają zachować spójność każdej perspektywy. Jeśli na przykład nazwa klasy ulega zmianie w diagramie klasy, diagram stanów wykorzystujący tę klasę również ulega zmianie.

Baza danych zawierająca elementy modelu to główny element odróżniający narzędzie CASE od zwykłego programu graficznego (rysunek 2.6). Każdy diagram UML można narysować za pomocą ołówka i kartki papieru lub dowolnego programu graficznego. W takim przypadku jednak każdy diagram będzie po prostu rysunkiem. Dopiero zastosowanie narzędzia CASE wyposażonego w bazę danych i zgodnego ze specyfikacją UML pozwala na stworzenie kolekcji danych, zarządzanie nią i modyfikowanie informacji bez obawy o naruszenie spójności tej kolekcji.



Rysunek 2.6. Narzędzie CASE jako baza danych

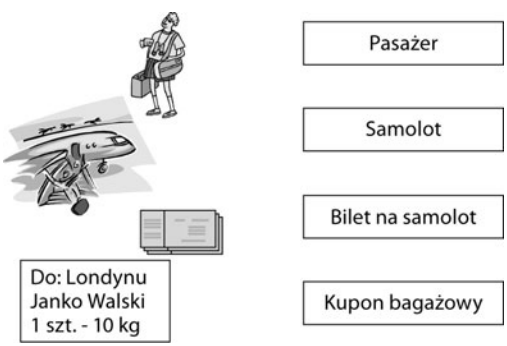
UML zawiera własny model bazy danych — jest to metamodel UML, element składowy specyfikacji UML (*OMG: Unified Modeling Language: Infrastructure, Version 2.0, Final Adopted Specification* — wrzesień 2003 i *OMG: Unified Modeling Language: Superstructure, Version 2.0, Revised Final Adopted Specification* — wrzesień 2004; obydwa dokumenty można znaleźć w serwisie *http://www.omg.org*). Wszystkie elementy występujące w diagramach UML oraz w opisach elementów są wyszczególnione w metamodelu UML. Można w nim znaleźć na przykład informację o tym, że klasa może zawierać atrybuty i metody. Ten model danych języka UML jest fundamentem baz danych narzędzi modelujących, czyli na przykład wszystkich narzędzi CASE. Niestety, większość narzędzi CASE jest zasobożerna, kosztowna, niedopracowana, kłopotliwa w obsłudze i wymagająca czasochłonnego szkolenia. Mimo tych wad w większości przypadków (z wyjątkiem naprawdę małych projektów) ich wykorzystanie daje niepodważalne korzyści.

2.3. Systemy informacyjne a systemy informatyczne

Praktycznie w każdym zawodzie przynajmniej część obowiązków jest związana z obróbką informacji. Tak dzieje się od tysiącleci i to było jedną z przyczyn powstania pisma. Jeden z najstarszych zapisków w dziejach Europy dotyczył listy magazynowej zaopatrzenia pałacu w Knossos na Krecie. Gdybyśmy byli w stanie poznać procedury działania pracowników zaopatrzenia sprzed 3500 lat, moglibyśmy odwzorować procesy biznesowe ich działań. Moglibyśmy na przykład zamodelować kontakty z dostawcami i odbiorcami w procesie wymiany towarów oraz przeanalizować zapisy ich działalności biznesowej. To samo dotyczy handlarzy oliwą w starożytnym Rzymie 1500 lat później, kupców Hanzy w piętnastowiecznych północnych Niemczech czy też firmy Lloyd's w Londynie na początku dwudziestego wieku.

W każdym z powyższych przykładów do zarządzania zadaniami wykorzystywane były mniej lub bardziej skomplikowane systemy informacyjne. Celem tych systemów było (i nadal jest) zarządzanie informacjami niezbędnymi do prowadzenia interesów. Oczywiście, wszystko odbywało się bez udziału komputerów. Systemy informacyjne były oparte na innych technikach, takich jak kreda i tablica czy też kartoteki. Obecnie komputery pozwalają na implementowanie systemów informacyjnych w postaci systemów informatycznych. To daje nowe możliwości, niewyobrażalne dla handlarza oliwą w starożytnym Rzymie. Główny cel tych systemów pozostał jednak ten sam — jest nim udostępnianie danych niezbędnych w codziennych procesach biznesowych. W tej książce będziemy zajmować się przede wszystkim systemami informatycznymi, ponieważ przyjmujemy założenie, że systemy informacyjne modelowane za pomocą techniki UML są implementowane z użyciem technologii informatycznych.

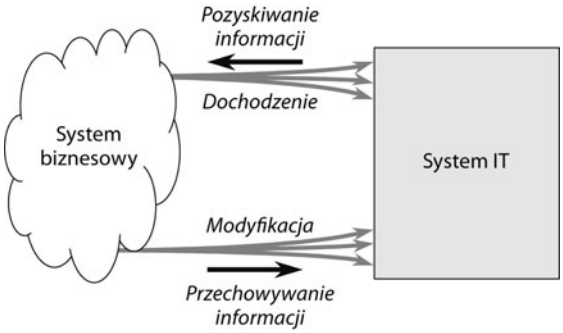
W naszej analizie systemu **obsługi pasażerów** na lotnisku zajmujemy się pasażerami, biletami lotniczymi i prawdziwymi odlotami. W systemie informacyjnym mamy jednak do czynienia z reprezentacjami (**odzwierciedleniami**) pasażerów, biletów i odlotów. Te reprezentacje składają się z **informacji** na temat pasażerów, biletów i odlotów zapisanych w systemie informatycznym. Są to informacje niezbędne do obsługi procesów biznesowych lotniska. Sytuację tę przedstawia rysunek 2.7.



Rysunek 2.7. Obiekty świata rzeczywistego i ich reprezentacje

System informatyczny działa z użyciem komputerów — jest to system, który dostarcza informacji niezbędnych do prowadzenia interesów. Informacje te są najczęściej dostarczane w rezultacie zapytania zadanego przez użytkownika. Oczywiście, aby odpowiadać na zapytania, system informatyczny musi być zasilony informacjami.

Rysunek 2.8 przedstawia schemat współpracy między systemami biznesowymi i informatycznymi. W zakresie procesów biznesowych systemu biznesowego informacja jest odczytywana i zapisywana w systemie informatycznym.



Rysunek 2.8. System informatyczny

Techniki modelowania omawiane w tej książce przydadzą się nie tylko na etapie tworzenia systemu informatycznego — będą również pomocne na etapie analizy systemów informacyjnych. Aby to zademonstrować, posłużymy się drugim przykładem (obok przeważającego w książce przykładu lotniska UML). Do tego drugiego przykładu będziemy wracać w różnych miejscach książki.

Tym razem proszę wyobrazić sobie biuro średniowiecznego kupca należącego do Hanzy. Kupiec ten nazywa się Hafenstein. Hanza była potężnym, działającym w okresie średniowiecza stowarzyszeniem zrzeszającym kupców w Niemczech północnych i w innych krajach bałtyckich.

Szefem biura jest skrupulatny sekretarz Hildebrand. W biurze przechowywane są różne księgi, między innymi księga rozliczeń dziennych, bilans sprzedaży oraz spis klientów. Każdą księgą opiekuje się inny księgowy. Nikt oprócz uprawnionego księgowego nie może wprowadzać zmian w księdze i tylko on wie dokładnie, w którym miejscu można znaleźć konkretną informację.

W naszej terminologii biuro, Hildebrandt, księgowi i księgi składają się na system informacyjny. Dzięki temu przykładowi chcemy w różnych miejscach książki zademonstrować, że chociaż system informacyjny jest dziś zwykle zaimplementowany w postaci systemu informatycznego, w rzeczywistości koncepcyjnie niewiele ma wspólnego z komputerami. Można go fizycznie zaimplementować na wiele różnych sposobów.

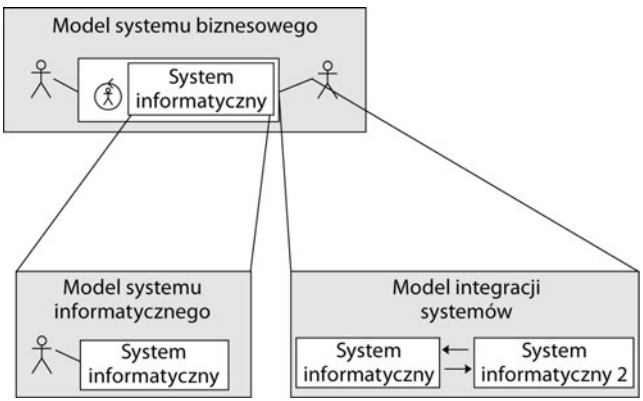
2.4. Modele studium przypadku

W naszym studium przypadku skonstruujemy trzy modele różnych systemów:

1. Model systemu biznesowego obsługi pasażerów, czyli zagadnienia biznesowe związane z systemem informatycznym. Model ten będzie obrazował procesy biznesowe, pasażerów, partnerów biznesowych, pracowników itp. Omówimy go w rozdziale 3.
2. Model systemu informatycznego obsługi pasażerów. Będzie on oparty o model systemu biznesowego obsługi pasażerów. Omówimy go w rozdziale 4.
3. Model integracji opisujący integrację systemów z otoczeniem, w szczególności interfejsy łączące system ze światem zewnętrznym. Również w tym przypadku fundamentem będzie model systemu biznesowego obsługi pasażerów. Omówimy go w rozdziale 5.

Aby zbudować i zintegrować system informatyczny, niezbędne będą wszystkie trzy wymienione wyżej modele: sam model systemu informatycznego nie wystarczy. Taka zależność zachodzi nie tylko w naszym studium przypadku, lecz również we wszystkich innych systemach.

Jak widać na rysunku 2.9, model systemu biznesowego stanowi podbudowę dla dwóch pozostałych modeli. Z tego powodu stanowi punkt wyjścia większości działań w ramach projektu. Zastosowanie **ujednoliconego języka modelowania** ma zatem ogromne zalety, ponieważ dzięki niemu jeden model będzie zrozumiały i użyteczny zarówno dla pracowników różnych działów biznesowych, jak i dla informatyków. Takie podejście umożliwia płynną wymianę modeli pomiędzy różnymi obszarami prac nad projektem. Również weryfikacja poprawności modeli jest dzięki temu znacznie uproszczona. Jesteśmy przekonani, że UML służy jako łącze między wymogami technicznymi a sferą funkcjonalną systemów informatycznych.



Rysunek 2.9. Modele studium przypadku

2.5. Historia UML-a — metody i notacje

Technologia informacyjna w swojej krótkiej historii zdążyła się już dorobić bardzo wielu **metod** i **notacji**. Istnieją metody i notacje dla projektu, struktury, przetwarzania i przechowywania informacji. Istnieją również metody planowania, modelowania, implementacji, składania, testowania, dokumentacji, dostosowywania itp. Niektóre ze stosowanych koncepcji są dość podstawowe i z tego powodu można je postrzegać jako wykraczające poza zakres technologii informacyjnej. Jednym z przykładów takich cech jest dziedziczenie, zjawisko znane ze świata rzeczywistego, będące jednocześnie podstawowym założeniem programowania zorientowanego obiektowo.

Mniej więcej do lat siedemdziesiątych twórcy oprogramowania traktowali proces programowania jako działalność niemal artystyczną. Jednakże systemy stawały się coraz bardziej skomplikowane, co spowodowało, że tworzenie oprogramowania i jego rozwój ze względów praktycznych nie mogły już być traktowane jako zadanie dla kreatywnych indywidualistów. Taka indywidualistyczna postawa doprowadziła stopniowo do **kryzysu branży**.

Kryzys doprowadził do wypracowania **metodologii inżynierskich** (inżynierii oprogramowania) i technik programowania strukturalnego. Zostały opracowane metody strukturalizacji systemów oraz procesu ich projektowania, tworzenia i rozwijania. Podejścia procesowe, na przykład metoda **HIPO** (ang. **Hierarchy Input Processing Output**), kładły główny nacisk na funkcjonalność systemów. Dzięki tej metodzie system jest dzielony na mniejsze elementy za pomocą rozkładu funkcjonalnego.

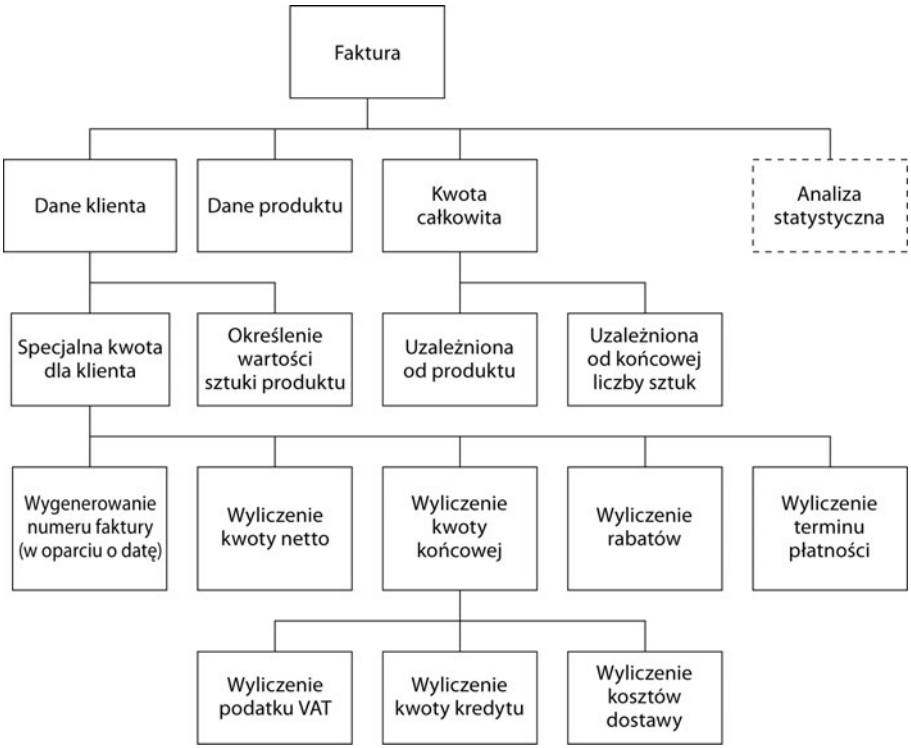
Rysunek 2.10 przedstawia graficzną reprezentację (diagram hierarchiczny) podfunkcji systemu wystawiania faktur. Każdy element jest opisany za pomocą schematu: *dane wejściowe-przetwarzanie-dane wyjściowe*.

Mniej więcej w tym samym okresie opracowano metodologie zorientowane na struktury danych. Jedną z nich jest metoda Jacksona, gdzie struktura programu jest oparta na graficznej reprezentacji struktur danych.

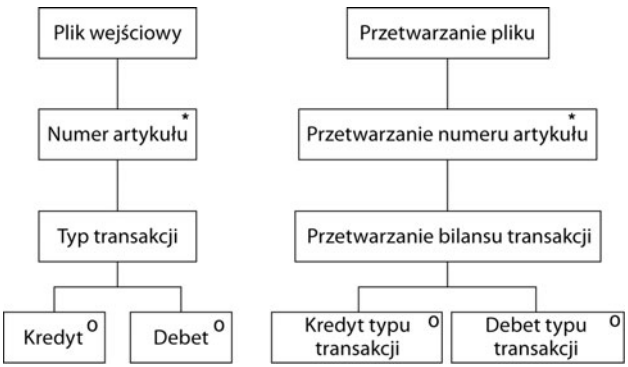
Na rysunku 2.11 w lewej kolumnie została przedstawiona struktura danych listy magazynowej. W prawej kolumnie przedstawiono zaś strukturę programu wypracowaną w oparciu o strukturę danych.

We wszystkich tych metodach i notacjach systemy są podzielone na dwie części — sekcję danych i sekcję procedur. Ten podział jest bardzo wyraźny w starszych językach programowania, takich jak COBOL. Diagramy przepływowe, strukturalne, HIPO i Jacksona są wykorzystywane do reprezentowania różnych funkcji. Oczywiście, te wczesne metody były jedynie podstawą do opracowania nowych systemów.

W latach osiemdziesiątych znacznie rozwinięto klasyczną analizę strukturalną. Twórcy oprogramowania uzyskali nowe narzędzia w postaci diagramów związków encji do modelowania danych oraz sieci Petriego do modelowania procesów.



Rysunek 2.10. Diagram HIPO



Rysunek 2.11. Diagram Jacksona

Wraz ze wzrostem poziomu komplikacji systemów projektowanie od zera stawało się metodą coraz mniej uzasadnioną ekonomicznie. Na znaczeniu zyskały takie cechy, jak łatwość utrzymania i możliwość ponownego użycia (ang. *reusability*). Opracowano języki programowania zorientowane obiektowo, a wraz z nimi pierwsze języki modelowania zorientowanego obiektowo (w latach siedemdziesiątych i osiemdziesiątych). W latach dziewięćdziesiątych pojawiły się

pierwsze publikacje na temat analizy obiektowej i projektowania obiektowego. W połowie lat dziewięćdziesiątych istniało ponad 50 metod zorientowanych obiektowo i tyle samo formatów projektowych. Potrzeba opracowania **ujednoliconego** języka modelowania stawała się coraz bardziej oczywista.

Na początku lat dziewięćdziesiątych szeroko rozpowszechnione były metodologie zorientowane obiektowo autorstwa Grady'ego Boocha i Jamesa Rumbaugha. W październiku 1994 roku firma Rational Software Corporation (od lutego 2003 wchodząca w skład korporacji IBM) rozpoczęła pracę nad ujednoliconym językiem modelowania. Na pierwszym etapie prac wprowadzono standardy dla notacji (język), ponieważ to zagadnienie uznano za najłatwiejsze w opracowaniu. Wykorzystano między innymi **metodę Boocha** (ang. **Booch Method** autorstwa Grady'ego Boocha), **OMT** (ang. **Object Modeling Technique** autorstwa Jamesa Rumbaugha) oraz **OOSE** (ang. **Object-Oriented Software Engineering** autorstwa Ivara Jacobsena) wraz z elementami innych metodyk. Opracowaną w ten sposób notację opublikowano jako język UML w wersji 0.9. Celem tych prac nie było sformułowanie zupełnie nowej notacji, lecz zaadaptowanie, rozwinięcie i uproszczenie istniejących metodyk, takich jak diagramy klas, diagramy użycia Jacobsona czy diagramy stanu Harela. W UML-u wykorzystano środki reprezentacji opracowane na potrzeby metod strukturalnych. Z tego powodu w diagramach aktywności języka UML znajdujemy wpływy diagramów przepływowych i sieci Petriego.

UML nie wyróżnia się zatem swoją nietypową formą, lecz tym, że narzuca daleko idącą standaryzację języka o formalnie zdefiniowanym znaczeniu stosowanych symboli.

W dalszych pracach nad językiem UML udział wzięły znane firmy, takie jak IBM, Oracle, Microsoft, Digital, Hewlett-Packard i Unisys. W 1997 roku opublikowano wersję 1.1 języka UML. Została ona zaakceptowana przez OMG. UML w wersji 1.2, zawierający pewne zmiany i poprawki, pojawił się w roku 1998, rok później powstała wersja 1.3, a UML 1.5 wypuszczono w marcu 2003. Tymczasem od roku 2000 trwały już prace nad wersją 2.0 UML-a, która została zaakceptowana przez OMG w postaci Final Adopted Specification w czerwcu 2003. Gdy oryginalna wersja tej książki trafiała do druku (czerwiec 2005), jeszcze nie zakończyła się ostatnia faza wdrażania standardu przez OMG (nie nastąpiło jeszcze przyjęcie specyfikacji jako **obowiązującej**).

2.6. Specyfikacja wymagań

Modele tworzonego systemu są integralną częścią każdej specyfikacji wymagań. W tej książce można znaleźć podstawy potrzebne do tworzenia tego typu modeli. Niestety, nie istnieje uniwersalna receptura tworzenia specyfikacji wymagań. Dobór szczegółów i poziom szczegółowości modelu zależą od wielu czynników. Z naszego doświadczenia wynika, że najważniejsze są odpowiedzi na następujące pytania:

- Kto definiuje specyfikację?
- Dla kogo jest ona przygotowywana?
- Jaki jest jej podmiot?

2.6.1. Wspomaganie procesów decyzyjnych

Modele i perspektywy zaprezentowane w tej książce stanowią podstawowe elementy składowe, z których — jak z cegieł — konstruuje się modele zdefiniowane w specyfikacjach wymagań. Poniższa tabela może być pomocna przy podejmowaniu decyzji dotyczących modeli i perspektyw.

Model (co?)	Perspektywa	Twórca (kto?)	Odbiorcy (dla kogo?)	Cel (po co?)
System biznesowy	Perspektywa zewnętrzna	Użytkownik	Użytkownik	Dokumentacja biznesowa
			Informatyk	Podstawy specyfikacji systemu informatycznego
	Perspektywa wewnętrzna	Użytkownik	Użytkownik	Dokumentacja biznesowa, opis procedur
			Informatyk	Podstawy specyfikacji systemu informatycznego
System informatyczny	Perspektywa zewnętrzna	Użytkownik	Informatyk Użytkownik	Wymagania w stosunku do systemu informatycznego
	Perspektywa strukturalna	Informatyk	Informatyk	Specyfikacja systemu informatycznego
	Perspektywa wydajności	Użytkownik Informatyk	Informatyk	Specyfikacja systemu informatycznego
	Perspektywa interakcji	Użytkownik Informatyk	Informatyk	Specyfikacja systemu informatycznego
Integracja systemów	Perspektywa procesów	Użytkownik	Informatyk	Specyfikacja integracji systemu informatycznego
	Perspektywa statyczna	Informatyk	Informatyk	Specyfikacja integracji systemu informatycznego

2.6.2. Weryfikacja

Wszystkie perspektywy omówione w tej książce opisują model, dokumentując wymagania w stosunku do niego z punktu widzenia użytkownika. Oznacza to, że wszystkie omawiane modele i perspektywy mają następujące wspólne cechy:

- mogą być skonstruowane wyłącznie przy ścisłej współpracy z użytkownikami;
- ich poprawność z punktu widzenia biznesowego może być zweryfikowana wyłącznie przez użytkowników,

Pomimo że model systemu informatycznego jest tworzony z myślą o docelowych odbiorcach, czyli informatykach zaangażowanych w implementację, nie ma możliwości jego realizacji bez udziału użytkowników, którzy muszą dostarczyć wymagania w stosunku do modelu. Reprezentują oni punkt widzenia użytkownika i są dostawcami wiedzy z dziedzin biznesowych.

W proces tworzenia i weryfikacji specyfikacji wymagań zaangażowane bywają różne grupy użytkowników, dlatego ważne jest wykorzystanie do tych celów ujednoliconego języka modelowania. Dzięki temu łatwiej uniknąć nieporozumień wynikających z błędnej interpretacji specyfikacji.

2.7. UML 2.0

2.7.1. Przegląd cech języka UML 2.0

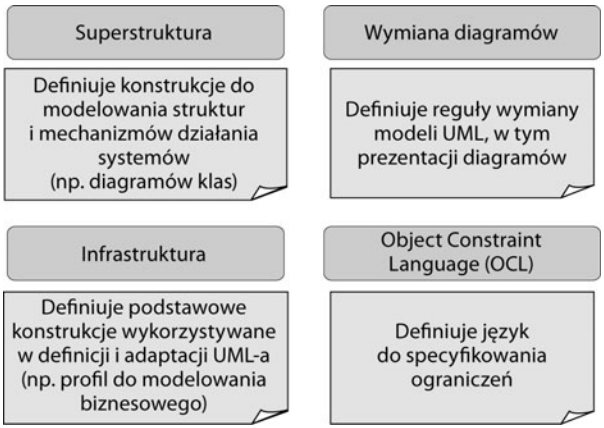
Książka *UML 2.0 w akcji. Przewodnik oparty na projektach* jest oparta na nowej wersji języka UML, czyli UML 2.0. W tej wersji struktura dokumentacji języka została zupełnie zmieniona. Obecnie istnieją dwa dokumenty opisujące UML:

- *UML 2.0 Infrastructure* opisuje podstawowe konstrukcje stanowiące fundament języka UML. Ten dokument nie przyda się bezpośrednio użytkownikowi (lub odbiorcy) języka UML; jest on przeznaczony raczej dla twórców narzędzi modelujących.
- *UML 2.0 Superstructure* definiuje konstrukcje języka UML 2.0 dostępne dla użytkownika — czyli te elementy UML-a, z którymi użytkownicy mają bezpośredni kontakt.

Ta wersja UML-a została opracowana przede wszystkim (ale nie tylko) przy następujących założeniach:

- zmiana struktury i przedefiniowanie UML w taki sposób, by uprościć jego wykorzystanie, implementację i adaptację;
- infrastruktura UML-a zawierająca takie cechy, jak:
  - wspólny rdzeń w postaci metajęzyka, za pomocą którego UML może zdefiniować sam siebie;
  - mechanizm umożliwiający doskonalenie języka.
- nadstruktura (ang. *superstructure*) UML-a posiadająca takie cechy, jak:
  - możliwość prostego tworzenia modeli w oparciu o komponenty;
  - możliwość udoskonalania elementów specyfikacji architektury;
  - możliwość dostarczania lepszych narzędzi do modelowania zachowań.

Oprócz specyfikacji *UML Infrastructure* i *UML Superstructure* opublikowane zostały dodatkowe dokumentacje, między innymi język **Object Constraint Language** (**OCL**) oraz diagram Interchange (dotyczący wymienności diagramów). Obecnie dokumenty te wchodzą w skład pakietu UML 2.0, co obrazuje rysunek 2.12.



Rysunek 2.12. Pełny pakiet specyfikacji UML 2.0

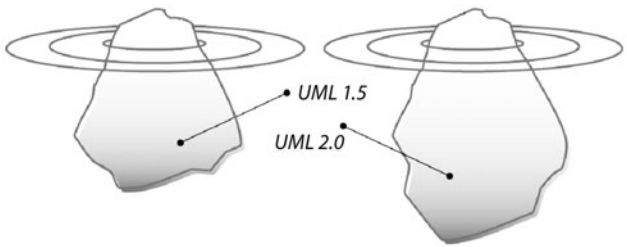
UML 2.0 jako całość jest bardziej rozbudowany i skomplikowany od swoich poprzednich wersji. Zakres dokumentacji dotyczącej tego języka również znacznie się poszerzył. Dokumentacja UML 1.5 wraz ze specyfikacją OCL miała rozmiar 730 stron, natomiast dokumentacja UML 2.0 — również z OCL — ma rozmiar ponad 1050 stron.

Pomijając już nawet fakt, że część dokumentacji „nie dotyczy" zwykłego użytkownika, przeczytanie całości przekracza możliwości przeciętnego członka zespołu twórców systemu. Nie chodzi tu wyłącznie o rozmiar mierzony liczbą stron, lecz przede wszystkim o poziom komplikacji konstrukcji UML-a. Z tego powodu bardzo ważnym zagadnieniem stało się opracowanie uproszczonego podzbioru elementów tego języka.

Konsekwencją takiego podejścia są dwa ważne spostrzeżenia dotyczące książki *UML 2.0 w akcji. Przewodnik oparty na projektach*. Książka ma za zadanie wykreować uproszczony obraz UML-a. Taki cel jest uzasadniony tym bardziej, że wersja 2.0 nie jest wcale bardziej przystępna dla użytkownika od wersji wcześniejszych.

Na szczęście naprawdę niewiele cech UML 2.0 ma wpływ na poziom szczegółowości zagadnień przedstawionych w tej książce. Zawiera ona niewiele zmian w porównaniu z wcześniejszymi wydaniami (w języku niemieckim), które oparte były na poprzednich wersjach UML-a. Ograniczony zakres tematyki pozwala na zachowanie stabilności merytorycznej tej publikacji dla każdej nowej wersji UML-a.

W książce świadomie prezentujemy jedynie wierzchołek góry lodowej i robimy to z pełną konsekwencją. Prezentowany wierzchołek (rysunek 2.13) ma ciągle ten sam rozmiar, niezależnie od tego, że część góry ukryta pod wodą stale się powiększa. Im bardziej ta sytuacja się rozwija, tym większe mamy przekonanie, że ten wierzchołek w zupełności wystarcza oczekiwanym odbiorcom książki, czyli członkom zespołów informatycznych wdrażających projekty.



Rysunek 2.13. Góra lodowa UML-a

Chcielibyśmy również podkreślić nowe możliwości, jakie daje UML 2.0. Jednym z założeń przyjętych przy tworzeniu UML 2.0 było stworzenie formalnej i kompletnej semantyki. Wykorzystanie tej cechy UML 2.0 pozwala w automatyczny sposób generować z modeli szkielety systemów. Taka cecha UML-a daje następujące możliwości:

- model opisany za pomocą UML-a nie jest abstrakcją, a odzwierciedleniem rzeczywistego systemu;
- błędy popełnione podczas konstrukcji modelu można wykrywać i poprawiać w sposób ciągły już od wczesnych etapów rozwoju;
- nie ma konieczności stosowania etapów pośrednich, takich jak modyfikowanie kodu niewchodzącego w skład projektu modelu;
- ten sam model może działać na różnych platformach (sprzętowych i programowych).

Te zalety okupione są jednak koniecznością uzyskania dogłębnej wiedzy na temat UML-a i włożenia dużego wysiłku w opracowanie modeli w sposób umożliwiający wykorzystanie pozytywnych cech tego języka.

2.7.2. Wpływ zmian w UML-u na modelowanie systemu biznesowego

Możliwości UML-a w zakresie modelowania systemów biznesowych zostały zwiększone dzięki zmianom wprowadzonym w elementach modelowania aktywności. Oto kilka przykładów zmian i usprawnień.

Diagramy aktywności nie są już po prostu szczególnymi przypadkami diagramów stanu. Ta zmiana sama w sobie nie ma większego znaczenia dla użytkownika UML-a. Jednak w połączeniu z nową autonomią metamodelu oferuje nowe, interesujące możliwości.

- Do tej pory osobne etapy w diagramie aktywności były interpretowane jako aktywności. Obecnie cały diagram interpretuje się jako **aktywność**, natomiast etapy (nazywane wcześniej aktywnościami) obecnie określa się mianem **akcji**. Akcja może wykorzystywać prostą operację lub wywoływać inną aktywność. Dzięki temu przy tworzeniu modelu można przyjąć zasadę zstępującą (od ogółu do szczegółu).
- Podział modelu na składowe nie musi wiązać się z koniecznością synchronizacji.
- Aktywność może mieć więcej niż jeden stan początkowy. Dzięki temu można zaplanować jednoczesne uruchomienie kilku zdarzeń.
- Aktywność może mieć zdefiniowane parametry wejściowe i wyjściowe.

Jedno z usprawnień diagramu sekwencji polega na uzupełnieniu go o koncepcję **operatorów**. Operatory umożliwiają połączenie kilku akcji lub aktywności w ramach jednego diagramu sekwencyjnego. Mogą one na przykład zostać użyte do zamodelowania wpływu innych diagramów sekwencyjnych na poszczególne sekwencje diagramu. Odpowiednie operatory można również wykorzystać do reprezentacji iteracji. Dzięki nowo wprowadzonym operatorom diagramy sekwencji również można tworzyć metodą zstępującą.

OCL jest obecnie integralną częścią UML-a. Tego języka można użyć do opisu argumentów, inwariantów, warunków wstępnych i wyjściowych w ramach modeli UML. Dzięki temu modele systemów i procesów biznesowych można projektować o wiele precyzyjniej.

2.7.3. Wpływ zmian w UML-u na modelowanie systemu informatycznego

Diagramy prezentowane w tej książce, reprezentujące różne perspektywy systemów IT, nie uległy znaczącym zmianom.

Największe zmiany dotyczyły notacji diagramów sekwencji. Tutaj nowa jest możliwość modularyzacji dzięki referencjom interakcji. W książce nie wprowadziliśmy jednak żadnych zmian, jeśli chodzi o poziom szczegółowości diagramów sekwencji. To samo dotyczy diagramów klas oraz przypadków użycia.

Najciekawszym zmianom w przypadku modelowania systemów informatycznych uległy diagramy stanu: punkty połączeń pozwalają na przykład na lepszy poziom modularyzacji diagramów stanu. Ze względu na założenie dotyczące uproszczenia prezentowanych zagadnień UML-a zdecydowaliśmy się jednak nie omawiać szczegółowo tego zagadnienia.

2.7.4. Wpływ zmian w UML-u na modelowanie integracji systemów

Oczywiście, usprawnienia w modelowaniu zachowań miały wpływ na perspektywę procesu w modelu integracji systemów. Zostały wprowadzone znaczące udoskonalenia polegające na wprowadzeniu parametrów wejściowych i wyjściowych dla aktywności (zobacz punkt 2.7.2, „Wpływ zmian w UML-u na modelowanie systemu biznesowego").

Niewielkie zmiany nastąpiły także w obszarze perspektyw statycznych, co ma wpływ na projektowanie obiektów biznesowych z diagramami klas.

Oprócz zmian wprowadzonych wraz z wersją 2.0 na znaczeniu zyskał profil **Enterprise Application Integration** (**EAI**), bardzo użyteczny przy integracji systemów. Oprócz podstawowych operacji niezbędnych przy integracji systemów profil EAI pozwala na prezentację metamodeli danych również dla nieobiektowych języków programowania. Zagadnienia te są jednak wykorzystywane na znacznie wyższym poziomie szczegółowości niż przyjęty w tej książce.

2.7.5. Podsumowanie

Dla zwykłego użytkownika UML 2.0 nie stanowi znacznej rewolucji w stosunku do wcześniejszych wersji, z pewnością jest jednak postrzegany jako usprawnienie koncepcji istniejących dotychczas. Warto rozważyć wykorzystanie UML-a 2.0 w przyszłych wdrożeniach projektów. Z drugiej strony, nic nie stoi na przeszkodzie, aby w istniejących projektach nadal stosować wykorzystaną wcześniej terminologię. W przypadku nowych projektów zalety UML-a 2.0 (dokładniejsze modelowanie) na pewno zrekompensują jego wady (na przykład większy nakład pracy).

3

ROZDZIAŁ

Modelowanie systemów biznesowych

Komercyjne systemy informatyczne są wykorzystywane głównie do obsługi systemów biznesowych. Z tego powodu tworzenie i integracja systemów IT są uzależnione od perspektyw procesów biznesowych. Za fundament systemu informatycznego służy model systemu biznesowego oraz jego procesów. W tym rozdziale omówimy zasady tworzenia modeli systemów biznesowych.

Aby zapewnić płynną wymianę danych biznesowych między systemami informatycznymi, koniecznie trzeba zrozumieć **środowisko biznesowe**, w którym działają systemy IT. Z tego powodu analiza i modelowanie procesów biznesowych stanowią ważne elementy procesu tworzenia i integracji systemów informatycznych.

Obecnie systemy informatyczne nie tylko są osadzone w środowisku biznesowym, lecz często bywają połączone z innymi systemami informatycznymi. Z tego powodu każdy system informatyczny musi być dopasowany nie do jednego, lecz do różnych środowisk docelowych.

- **Integracja na poziomie procesów biznesowych** — każdy system informatyczny musi mieć aktywności systemu biznesowego zaimplementowane w sposób umożliwiający poprawne i wydajne wykonywanie procesów biznesowych związanych ze wszystkimi elementami.
- **Integracja na poziomie systemu informatycznego** — komunikacja z innymi systemami informatycznymi na poziomie wymaganym przez system biznesowy musi odbywać się w sposób płynny. Do tego niezbędne są doskonałe interfejsy — doskonałe zarówno z punktu widzenia semantycznego, jak i technicznego. Integracja na poziomie systemu informatycznego zostanie omówiona w rozdziale 5.

Proponujemy, aby Czytelnik samodzielnie odpowiedział na poniższe pytania:

- Kiedy ostatnio zdarzyło Ci się napotkać funkcjonalną rozbieżność między nowym systemem informatycznym a jego środowiskiem działania, powstałą w trakcie procesu tworzenia?
- Jak wiele znanych Ci systemów informatycznych nie wspomaga w sposób optymalny działań użytkowników (a ile wręcz im przeszkadza)?
- Kiedy ostatnio zdarzyło Ci się napotkać system informatyczny, który musiał być zatrzymany w dniu swojej premiery, ponieważ okazało się, że błąd funkcjonalny w interfejsach uniemożliwiał jego wykorzystanie?

Nie chcemy skupiać się jedynie na dynamicznej sferze naszego modelu; powiemy również co nieco o jego elementach statycznych. Z tego powodu skonstruujemy model systemu biznesowego, który będzie reprezentował zarówno procesy, jak i struktury biznesowe.

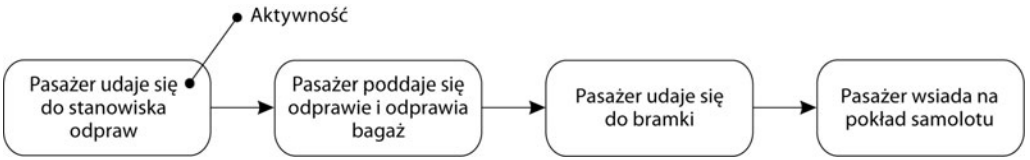
3.1. Procesy i systemy biznesowe

3.1.1. Czym jest proces biznesowy?

Większość ludzi intuicyjnie definiuje pojęcie **procesu biznesowego** jako **procedurę** lub wręcz **zdarzenie** realizowane z myślą o osiągnięciu **celu**. Na przykładzie lotniska UML możemy wyodrębnić wiele procesów biznesowych i celów:

- Celem pasażera jest wyjazd na wakacje. Aby go osiągnąć, pasażer musi zarezerwować lot i hotel, spakować bagaże, dojechać do lotniska UML, poddać się odprawie i wsiąść do samolotu, a następnie wysiąść z niego na lotnisku docelowym, dostać się do swojego pokoju hotelowego i rozpakować bagaże.
- Właściciel kiosku z prasą na lotnisku UML chce sprzedawać swój towar. W tym celu kupuje towar jak najtaniej i sprzedaje go pasażerom lotniska po wyższej cenie.
- Aby pasażerowie mogli wsiąść do samolotu, muszą poddać się odprawie. W tym celu pracownicy odpraw przyjmują ich bilety i bagaże, zasięgają informacji na temat preferencji dotyczących miejsca w samolocie i wykorzystują system informatyczny. Na końcu procedury pasażerowie otrzymują swoje karty pokładowe, na których mają zapisane numery foteli oraz bramek.

Procesy biznesowe są często realizowane wieloetapowo. Te etapy określa się mianem **aktywności**. Muszą być one zrealizowane w określonej kolejności. Właściciel kiosku nie może sprzedawać towaru, jeśli wcześniej go nie zakupi. Pasażer pakuje walizkę przed wyjazdem z domu na lotnisko. Pracownik odpraw może wręczyć pasażerowi kartę pokładową dopiero po zakończeniu odprawy (rysunek 3.1).



Rysunek 3.1. Aktywność procesu biznesowego „odprawa pasażerów" (wersja uproszczona)

Aktywności mogą być realizowane **sekwencyjnie** lub **równolegle**. Na przykład w czasie, gdy bagaże są transportowane do samolotu, pasażer może zakupić jakiś towar w sklepie wolnocłowym.

Indywidualne aktywności mogą być odpowiednio **rozproszone**. Procedura odprawy jest przeprowadzana przez pracownika odpraw, natomiast wsiadanie do samolotu odbywa się w innym miejscu i jest koordynowane przez innych pracowników lotniska.

Z reguły aktywności procesów biznesowych są od siebie zależne. Takie zależności powstają wskutek interakcji między aktywnościami należącymi do tego samego procesu biznesowego, czyli między aktywnościami prowadzącymi do tego samego celu.

Które z poniższych aktywności naszego studium przypadku nie są od siebie zależne, ponieważ nie prowadzą do tego samego celu, który można zdefiniować tak: „polecieć na wakacje Airbusem 320".

- ładowanie żywności i napojów na pokład samolotu Airbus 320;
- tankowanie paliwa do Boeinga 737;
- sprzątanie toalet na lotnisku UML;
- awansowanie na stanowisko wiceprezesa jednego z pracowników lotniska UML.

3.1.2. Definicja autorstwa Workflow Management Coalition

Oficjalne definicje terminów **proces** i **proces biznesowy** zostały przyjęte przez **Workflow Management Coalition**. Można je znaleźć w słowniku ***Workflow Reference Model*** autorstwa Workflow Management Coalition (*The Workflow Reference Model*, luty 1999: *http://www.wfmc.org*):

> „Proces jest skoordynowanym (równoległym lub sekwencyjnym) zbiorem aktywności wykonywanych z myślą o osiągnięciu wspólnego celu. Tego typu aktywności mogą być manualne oraz (lub) zautomatyzowane".

Zgodnie z tą definicją proces jest zbiorem aktywności wykonywanych w sposób skoordynowany, jednocześnie lub jedna po drugiej, zmierzających do jednego celu. Te aktywności można wykonywać ręcznie lub za pomocą systemu informatycznego.

> „Proces biznesowy jest formą procesu występującego w ramach struktury organizacyjnej i podporządkowanego osiągnięciu celów biznesowych".

3.1.3. Systemy biznesowe

W poprzednim punkcie wyjaśniliśmy pojęcie procesów biznesowych. Są one dynamiczne z natury i wykorzystują aktywności. Jeśli jednak chcemy spojrzeć na system biznesowy całościowo, musimy uwzględnić również elementy statyczne, czyli między innymi struktury organizacyjne, w ramach których odbywają się procesy biznesowe. Musimy również wziąć pod uwagę obiekty biznesowe oraz informacyjne, takie jak bilety i zlecenia. Terminu „system biznesowy" używamy właśnie w odniesieniu do połączenia elementów statycznych i dynamicznych.

W terminologii biznesowej **systemem biznesowym** nazywamy łańcuch elementów generujący wartość dodaną. W skład tego łańcucha wchodzi proces generujący tę wartość — może nim być na przykład dostawa towarów lub usług. Biznes może wykorzystywać jeden lub kilka systemów biznesowych.

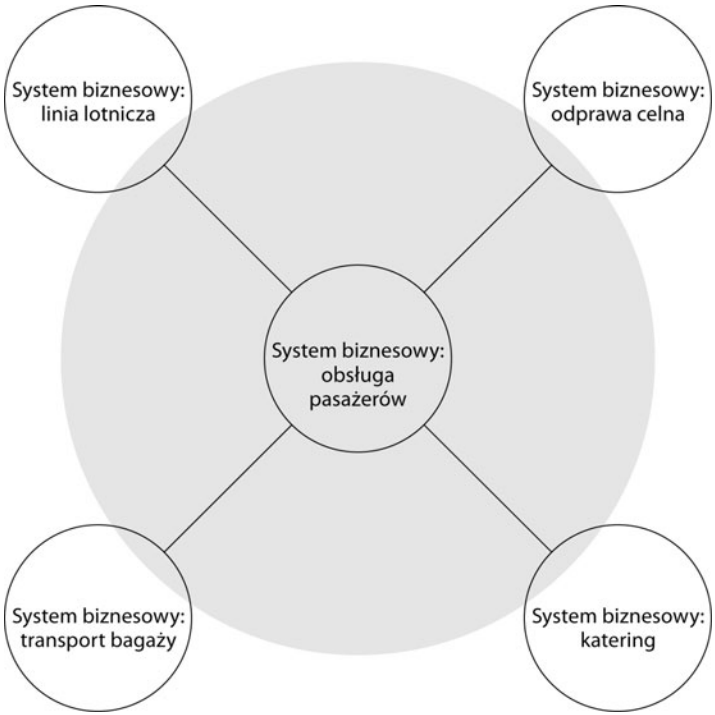
Każdy system biznesowy sam w sobie generuje korzyść ekonomiczną. Należy założyć, że osoba patrząca z informatycznego punktu widzenia powinna postrzegać system biznesowy w podobny sposób, jak osoba patrząca z biznesowego punktu widzenia. Efekty działania systemu biznesowego określa się również mianem funkcji.

Analiza i modelowanie systemu biznesowego narzucają potrzebę określenia ograniczeń systemowych. Modelowany system biznesowy może obejmować całą organizację. W takim przypadku mamy do czynienia z **modelem organizacyjnym**.

Można również analizować i zamodelować jedynie część organizacji. W naszym studium przypadku system informatyczny ma być zintegrowany w sferze działań lotniska związanej z obsługą pasażerów. W związku z tym wystarczy, jeśli ograniczymy się w analizie i konstruowaniu modelu jedynie do tej sfery pracy lotniska.

Obsługa pasażerów jest realizowana przez określony dział lotniska UML. W tym dziale są zatrudnieni pracownicy, ma on zdefiniowaną strukturę organizacyjną, posiada system informatyczny i ma też jasno wyznaczone zadania (rysunek 3.2). Działy wspomagające, takie jak dział transportu bagaży czy katering, również należą do lotniska UML, lecz nie wchodzą w skład naszego systemu biznesowego. Będziemy zatem traktować je tak samo, jak inne, zewnętrzne systemy biznesowe.

Nie interesują nas zewnętrzne systemy biznesowe jako całość, a jedynie **interfejsy** między nimi i naszym systemem biznesowym. Na przykład pracownicy obsługi pasażerów muszą wiedzieć o tym, że bagaże pasażerów powinny być dostarczone do punktu transportu bagaży, aby stamtąd mogły być załadowane do samolotu. Oczywiście, w związku z tym pracownicy obsługi pasażerów muszą wiedzieć, **w jaki** sposób bagaż ma być dostarczony do transportu, aby procedura została zakończona bez problemów. Uzasadnione jest zatem połączenie systemów informatycznych obsługi pasażerów oraz transportu bagaży. Oznacza to, że muszą zostać stworzone odpowiednie interfejsy między systemami. Z drugiej strony, obsługa pasażerów nie ma żadnego związku ze sposobem organizacji transportu bagaży — czyli na przykład nie ma znaczenia (z punktu widzenia obsługi pasażerów), czy każda walizka jest ręcznie przenoszona przez pracownika do luku samolotu, czy też do tego celu są wykorzystywane wózki bagażowe.



Rysunek 3.2. Granica systemu na potrzeby analizy systemu biznesowego

3.1.4. Wykorzystywanie UML-a do modelowania procesów i systemów biznesowych

Zanim przejdziemy do modelowania procesów i systemów biznesowych za pomocą UML-a, powinniśmy zadać sobie pytanie, czy UML w ogóle nadaje się do ich modelowania. Przyjrzymy się zatem definicji UML-a zdefiniowanej przez OMG. *Object Management Group Inc.* to międzynarodowe stowarzyszenie firm promujących otwarte standardy zastosowań technik obiektowych. Stowarzyszenie to opracowało każdą z wydanych dotychczas wersji UML-a. Aktualną wersję definicji można znaleźć na stronie *http://www.omg.org*. Definicja języka UML jest następująca:

> „Unified Modeling Language jest wizualnym językiem służącym do specyfikowania, konstruowania i dokumentowania artefaktów systemów" (*UML Unified Modeling Language: Infrastructure, Version 2.0, Final Adopted Specifications*, wrzesień 2003).

Przytoczona tu definicja jasno wskazuje, że UML jest językiem do modelowania i reprezentowania wszelakich systemów, a więc również systemów biznesowych.

UML realizuje przynajmniej jeden z wymogów modelowania systemowego: pozwala odzwierciedlić różne perspektywy systemu biznesowego, dzięki czemu możliwe jest uchwycenie jego różnych aspektów. Fakt istnienia różnych typów diagramów ustandaryzowanych w ramach UML-a dowodzi, że ten wymóg jest spełniony. Każdy typ diagramu pozwala bowiem modelować różne perspektywy działania systemu.

Możliwości UML-a są oczywiście ograniczone i w przypadku wielkich projektów biznesowych te ograniczenia mogą stanowić pewną przeszkodę. Do takich celów opracowano specjalne metodologie i narzędzia, między innymi **Architecture of Integrated IT Systems** (**ARIS**). Gdy spotkamy się z sytuacją, w której nie da się wykorzystać możliwości UML-a, warto zainteresować się tego typu rozwiązaniami. Nie oznacza to jednak, że odradzamy stosowanie UML-a w rozbudowanych projektach. Szczególnie wówczas warto dokładnie przestudiować jego specyfikację (*OMG: Unified Modeling Language: Superstructure, Version 2.0, Revised Final Adopted Specification*, wrzesień 2004) oraz stosować narzędzia CASE.

Ten tekst jest dostosowany do projektów prowadzących do tworzenia systemów informatycznych. Co więcej, skupiliśmy się na takich aspektach wdrażania systemów biznesowych, w których kluczowa jest jak najbardziej bezkonfliktowa integracja całego systemu informatycznego. Projekty tego typu mają kilka charakterystycznych cech wspólnych:

- Rozważane są te procesy biznesowe, na których działanie będzie miała wpływ integracja systemu informatycznego.
- Modelowanie procesów biznesowych nie jest kluczowym aspektem tych projektów. Model służy raczej jako fundament pod konstrukcję integracji systemów informatycznych. Ponadto często bywa tak, że integracja procesów biznesowych stanowi kluczowy czynnik sukcesu lub porażki projektów tego typu. Niezależnie od tego najważniejszym zadaniem pozostaje jednak konstrukcja systemu informatycznego.
- Budżety na projekty z reguły są ograniczone, nakład czasu na proces modelowania w tych projektach nie powinien przekraczać 5 – 10% całkowitego czasu realizacji.

3.1.5. Praktyczne wskazówki dotyczące modelowania procesów biznesowych

Często słyszy się opinie, że analiza i modelowanie procesów biznesowych to zadania skomplikowane i żmudne. Z naszych doświadczeń wynika jednak, że większość procesów biznesowych dość łatwo zrozumieć i kontrolować. Procesy te wydają się bardziej skomplikowane, niż są w rzeczywistości, z powodu słabej czytelności i braku jednoznacznej definicji ich działania.

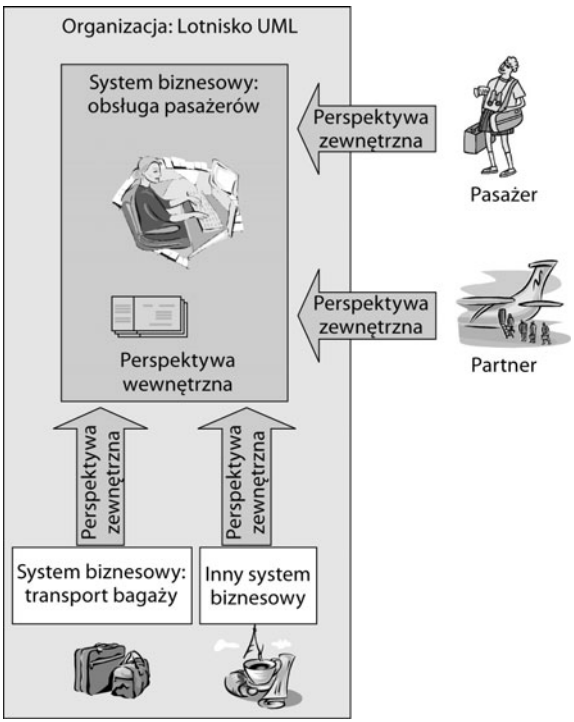
W wielu przypadkach istniejące procesy biznesowe są udokumentowane słabo lub wręcz nie są udokumentowane w ogóle. Jedną z przyczyn takiego stanu rzeczy jest pewna zaszłość: bardzo często poszczególne sfery działań biznesowych były traktowane jako osobne byty, nie zaś jako integralne elementy procesów biznesowych. Z tego powodu brakuje świadomości istnienia powiązań między aktywnościami (czyli żadna z osób zaangażowanych w jakąś działalność biznesową nie potrafi wskazać całego łańcucha procesu). Jeśli zabraknie nam tej perspektywy postrzegania systemu, cały proces biznesowy wydaje się bardziej skomplikowanym tworem.

W przypadku, gdy modelowane procesy biznesowe są już obsługiwane przez systemy informatyczne, mogą pojawić się nowe przeszkody. W większości przypadków nie jest dostępna dokumentacja przepływu informacji między poszczególnymi systemami informatycznymi. Czasem bywa też tak, że funkcje systemu informatycznego nie są znane, ponieważ jest to proces działający w sposób zautomatyzowany, ukryty przed użytkownikami, przyjmujący postać „czarnej skrzynki". Użytkownicy mają dostęp jedynie do danych wejściowych i wyjściowych systemu.

Proces modelowania mogą znacznie przyspieszyć istniejące modele architektury procesów lub modele referencyjne. Pomocne może okazać się również porównywanie procesów z podobnymi lub identycznymi procesami zamodelowanymi w innych organizacjach — szczególnie, gdy chcemy zidentyfikować odstępstwa od modelu i określić potencjalne obszary usprawnień.

3.2. Jeden model — dwie perspektywy

System biznesowy może analizować, biorąc pod uwagę różne punkty widzenia. Nasz model systemu biznesowego będzie składał się z dwóch różnych perspektyw. Każda z nich skupia się na nieco innych aspektach tego systemu, ale są one wzajemnie ze sobą powiązane. Perspektywy te prezentuje rysunek 3.3.



Rysunek 3.3. Zewnętrzna i wewnętrzna perspektywa systemu biznesowego

Patrząc na system biznesowy od zewnątrz, przyjmujemy rolę klienta, partnera biznesowego, dostawcy lub innego systemu biznesowego. Z takiej **zewnętrznej perspektywy** interesujące są tylko te procesy, które uwzględniają obiekty zewnętrzne w stosunku do systemu. Perspektywa zewnętrzna opisuje zatem środowisko systemu biznesowego. Sam system biznesowy jest z tej perspektywy postrzegany jako czarna skrzynka.

Patrząc na system biznesowy od wewnątrz, widzimy **pracowników** i **narzędzia** biorące udział w realizacji potrzeb otoczenia i obsłudze niezbędnych procesów biznesowych. Oprócz procesów biznesowych mamy do czynienia z procesami **przepływu informacji** (ang. *workflow*) oraz **systemami informatycznymi**. Każdy pracownik jest częścią **struktury organizacyjnej**. W normalnej sytuacji **perspektywa wewnętrzna** jest ukryta przed światem zewnętrznym.

Rzućmy okiem na nasze studium przypadku z punktu widzenia pasażera:

- Jakie usługi są świadczone przez dział obsługi pasażerów?
- Z którymi pracownikami działu obsługi pasażerów mamy szansę się spotykać?
- Czy widoczna jest dla nas procedura realizowana przez pracownika odpraw podczas wydawania karty pokładowej? Czy interesuje nas sposób dostarczenia bagażu na pokład, czy też nie ma to większego znaczenia przy założeniu, że nie ulegnie on zgubieniu lub uszkodzeniu?

Na pierwszy ogień podczas naszych prac modelowych pójdzie perspektywa zewnętrzna. Innymi słowy, zaczniemy od opisu systemu biznesowego z punktu widzenia klienta, partnera biznesowego lub dostawcy.

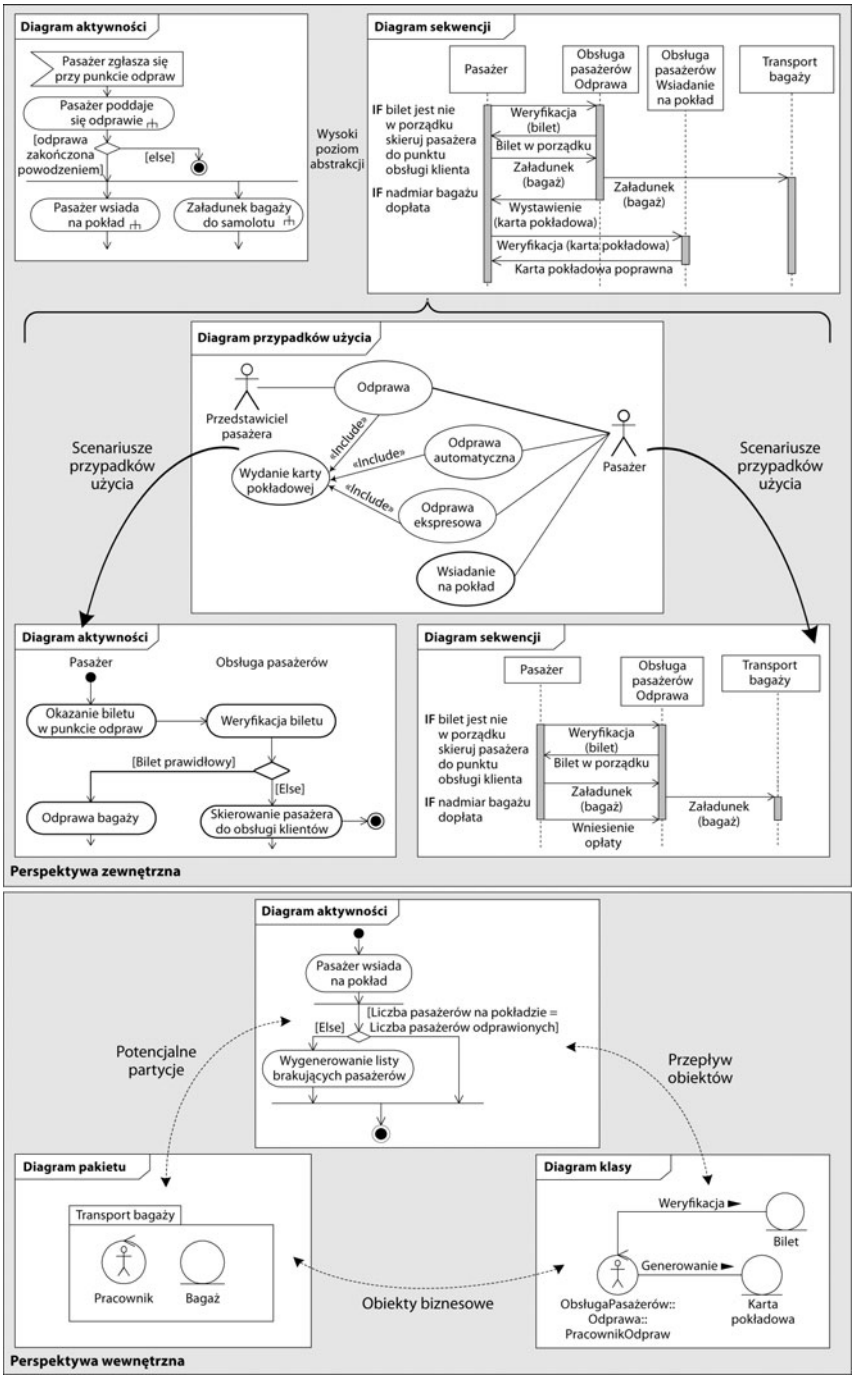
Perspektywa wewnętrzna stanowi odpowiedź na pytanie, **jak** system biznesowy świadczy te usługi. Przypadki użycia zdefiniowane w ramach perspektywy zewnętrznej spełniają rolę fundamentu dla scenariuszy testowych niezbędnych do testowania gotowych systemów informatycznych.

Rysunek 3.4 przedstawia indywidualne perspektywy wykorzystywane do modelowania systemu biznesowego oraz diagramy UML opracowane na ich podstawie.

3.3. Perspektywa zewnętrzna

3.3.1. Jakie korzyści przynosi modelowany system biznesowy?

Klient lub partner biznesowy organizacji nie jest zainteresowany szczegółami jej funkcjonowania, takimi jak to, czy przekazywanie informacji odbywa się w niej drogą informatyczną, czy nieinformatyczną. Nie ma również znaczenia liczba i zawartość formularzy wypełnianych przez pracowników. Klienci i partnerzy biznesowi są zainteresowani wyłącznie tym, jakie rodzaje towarów lub usług oferuje organizacja i jaki mogą mieć z nich pożytek. Perspektywa klienta opisuje interakcje systemu z obiektami zewnętrznymi, takimi jak klienci i partnerzy, a system biznesowy jest przez nią interpretowany jako czarna skrzynka.



Rysunek 3.4. Różne perspektywy i diagramy

Weźmy pod uwagę system biznesowy, na przykład obsługę pasażerów lub kiosk z prasą — jak są one postrzegane od zewnątrz? Które wyniki ich działania mają znaczenie dla klientów i partnerów biznesowych? Czy te wyniki działania są towarami, czy usługami?

Z punktu widzenia zarządu organizacji biznesowej celem systemu biznesowego jest generowanie wyniku działania (przynoszącego zysk — zobacz również punkt 3.1.3, „Systemy biznesowe"). Wyniki działania można ogólnie podzielić na towary i usługi. Wytwarzanie towarów to na przykład produkcja tabliczek szwajcarskiej czekolady o najwyższej jakości.

W jaki jednak sposób wyróżnić usługi? Usługi są towarem niematerialnym — usługą jest na przykład rezerwacja miejsca lub transport bagażu na pokład samolotu. W przeciwieństwie do dóbr materialnych usługi nie są materializowane aż do momentu, gdy między usługodawcą i klientem dojdzie do kontaktu. Usługa może jednak wykorzystywać towar. Jeśli w kiosku z prasą sprzedawane są tabliczki szwajcarskiej czekolady, sama transakcja sprzedaży jest usługą. Można stąd wyciągnąć wniosek, że transakcja wykonana na dobrach materialnych jest traktowana jako usługa, ponieważ bierze w niej udział klient.

Z uwagi na to, że w dostarczaniu dóbr materialnych i usług bierze udział klient, proces ten ma znaczenie z punktu widzenia perspektywy zewnętrznej. Perspektywa ta nie opisuje mechanizmów obsługi danej transakcji przez pracowników organizacji i jej system informatyczny; nie opisuje też tego, w jaki sposób procesy biznesowe są realizowane w ramach systemu biznesowego. W perspektywie zewnętrznej znaczenie mają jedynie te aktywności, które odbywają się z udziałem obiektów zewnętrznych w stosunku do systemu.

W naszym studium przypadku pasażer powinien wiedzieć, że z ważnym biletem na lot musi zgłosić się przy stanowisku odpraw oraz że po pomyślnej odprawie otrzyma kartę pokładową. Czynności, które muszą wykonać pracownicy i system informatyczny lotniska, aby ta karta pokładowa została wygenerowana, są z reguły ukryte przed pasażerem. W większości przypadków nie jest on nawet w żadnym stopniu tym zainteresowany.

W praktyce okazuje się, że perspektywa zewnętrzna jest trudna do zobrazowania, jeśli model jest opracowywany przez pracowników organizacji, z natury rzeczy będących jej **wewnętrznymi** obserwatorami. Osobie obserwującej system biznesowy od środka, znającej wewnętrzne zależności, trudno jest wczuć się w rolę klienta, który z kolei nie bierze pod uwagę żadnych wewnętrznych powiązań. Jeśli pomieszalibyśmy perspektywy zewnętrzną i wewnętrzną, powstałby model obrazujący system od zewnątrz oraz uwzględniający jego procesy wewnętrzne. W taki właśnie sposób powstają systemy mało przyjazne użytkownikowi. Dlatego do prac nad perspektywą zewnętrzną systemu należy zaangażować uczestników systemu o niskim stopniu nasycenia szczegółami dotyczącymi wewnętrznych mechanizmów biznesowych, na przykład pracowników innych działów lub zewnętrznych konsultantów.

Biznesowe przypadki użycia

Zanim przejdziemy do omawiania **biznesowych przypadków użycia**, warto zapoznać się z ogólną definicją przypadku użycia wykorzystywaną w UML-u. Przypadek użycia jest specyfikacją takiego zbioru transakcji realizowanych przez system, który prowadzi do uzyskania wyraźnego wyniku. Wynik ten najczęściej ma określoną wartość dla jednego lub większej liczby aktorów lub innych odbiorców systemu (*OMG: Unified Modeling Language: Superstructure, Version 2.0, Revised Final Adopted Specification*, wrzesień 2004).

Co to jest „wyraźny wynik o określonej wartości" w przypadku systemu biznesowego? Odpowiedź na to pytanie stanowi również podstawę tworzenia diagramów przypadków użycia. Problem ten jest tematem dyskusji analityków i projektantów od momentu powstania definicji po dzień dzisiejszy. Przypadkami użycia w naszym systemie biznesowym są usługi świadczone na rzecz klientów, partnerów biznesowych lub innych systemów biznesowych. Natomiast wewnętrzne funkcje systemu, które nie są widoczne ani dostępne dla zewnętrznych obserwatorów, to zbiór aktywności wewnętrznych składających się na wewnętrzny proces biznesowy.

Na poziomie modelowania biznesowego w miejsce pojęcia przypadku użycia posługujemy się pojęciem biznesowego przypadku użycia. To rozróżnienie wprowadzono w celu wyraźnego oddzielenia tych zagadnień i uniknięcia niejednoznaczności na poszczególnych etapach prac między modelowaniem systemu biznesowego a modelowaniem systemu informatycznego. Biznesowy przypadek użycia jest stosowany wyłącznie w modelu systemu biznesowego. Poza tym nie ma jednak żadnych różnic merytorycznych między przypadkiem użycia a biznesowym przypadkiem użycia.

Procesy biznesowe mogą być realizowane metodami nieinformatycznymi lub informatycznymi. Dzięki współczesnej technice całe systemy biznesowe mogą funkcjonować zupełnie bez udziału człowieka. Biznesowe przypadki użycia mogą zatem być obsługiwane bezpośrednio przez człowieka lub wspomagane komputerowo.

Jeśli przyjrzymy się działaniu naszego biura handlarza zrzeszonego w Hanzie, znajdziemy sporo biznesowych przypadków użycia — wykonywanych oczywiście metodami tradycyjnymi. Jeśli klient biura handlowego zamówi futra z Rosji, urzędnik bierze pióro i kałamarz i wpisuje odpowiednie zamówienie do książki zamówień. Biznesowe przypadki użycia istniały już w średniowieczu.

W naszym studium przypadku w dziale obsługi pasażerów realizowane są zarówno aktywności manualne, jak również te wspomagane informatycznie. Na przykład system informatyczny jest wykorzystywany przy rezerwacji miejsc dla pasażerów, natomiast operacje związane z weryfikacją biletów są realizowane w sposób ręczny.

Pasażer poddający się automatycznej odprawie nie spotka żadnego pracownika departamentu odpraw. Maszyna odprawiająca pasażerów sama realizuje cały biznesowy przypadek użycia.

Aktorzy

Zewnętrznymi uczestnikami systemu (ang. *outsiders*) są na przykład klienci lub partnerzy biznesowi, którzy wykorzystują wyniki działania analizowanego systemu. Nie ma potrzeby, aby ci **uczestnicy zewnętrzni** znali szczegółowo procedury realizacji biznesowych przypadków użycia. W przypadku pasażerów ważne jest, aby wiedzieli, że mogą kupić butelkę whisky w sklepie wolnocłowym. Butelka whisky jest **dobrem materialnym** oferowanym przez sklep wolnocłowy. Sama sprzedaż butelki klientowi jest już **usługą**. Pasażera nie interesuje sposób realizacji tej usługi przez pracownika sklepu. Zewnętrzni uczestnicy systemu określani są terminem **aktorów** (rysunek 3.5).

Rysunek 3.5. Uczestnicy zewnętrzni i członkowie załogi

Biznesowy przypadek użycia zawsze jest inicjowany przez aktora, co oznacza, że klient lub partner biznesowy wykorzystuje usługi systemu. Nasz pasażer przechadzający się po sklepie wolnocłowym spostrzega Scottish Malt Whisky i decyduje, że jej cena mu odpowiada, więc kupi jedną butelkę. W tym momencie staje się inicjatorem sprzedaży. Podczas transakcji biznesowego przypadku użycia aktorzy mogą wchodzić w interakcje z pracownikami i systemami informatycznymi składającymi się na system. Pracownicy wraz z tymi systemami są odpowiedzialni za realizację transakcji. Na przykład nasz pasażer musi wręczyć pracownikowi sklepu wolnocłowego określoną kwotę pieniędzy w zamian za kupowaną butelkę whisky.

Aktywności inicjowane przez pracowników lub system informatyczny w ramach systemu biznesowego nie stanowią biznesowych przypadków użycia w perspektywie zewnętrznej, lecz są biznesowymi przypadkami użycia w perspektywie wewnętrznej i powinny być uwzględnione przy tworzeniu diagramu aktywności lub diagramu sekwencji w perspektywie wewnętrznej.

Jak można zauważyć, aktorami systemu biznesowego mogą być ludzie, organizacje lub systemy informatyczne. Nawet organizacje są na diagramach reprezentowane tym samym symbolem aktora. Na przykład w przypadku transportu bagaży aktorzy stanowią reprezentację osób

biorących udział w przypadkach użycia — w ich inicjalizacji i realizacji. W naszych modelach bardzo duże znaczenie odgrywają **role** realizowane przez tych aktorów. Na poziomie modelu systemu biznesowego nie ma znaczenia, czy daną rolę odgrywa osoba, system informatyczny, organizacja, departament organizacji, maszyna lub inny system.

Przyjrzyjmy się ponownie naszemu studium przypadku i spróbujmy zlokalizować zaangażowane osoby, jednostki organizacyjne i systemy informatyczne. Następnie spróbujmy ustrukturalizować je według następujących kryteriów:

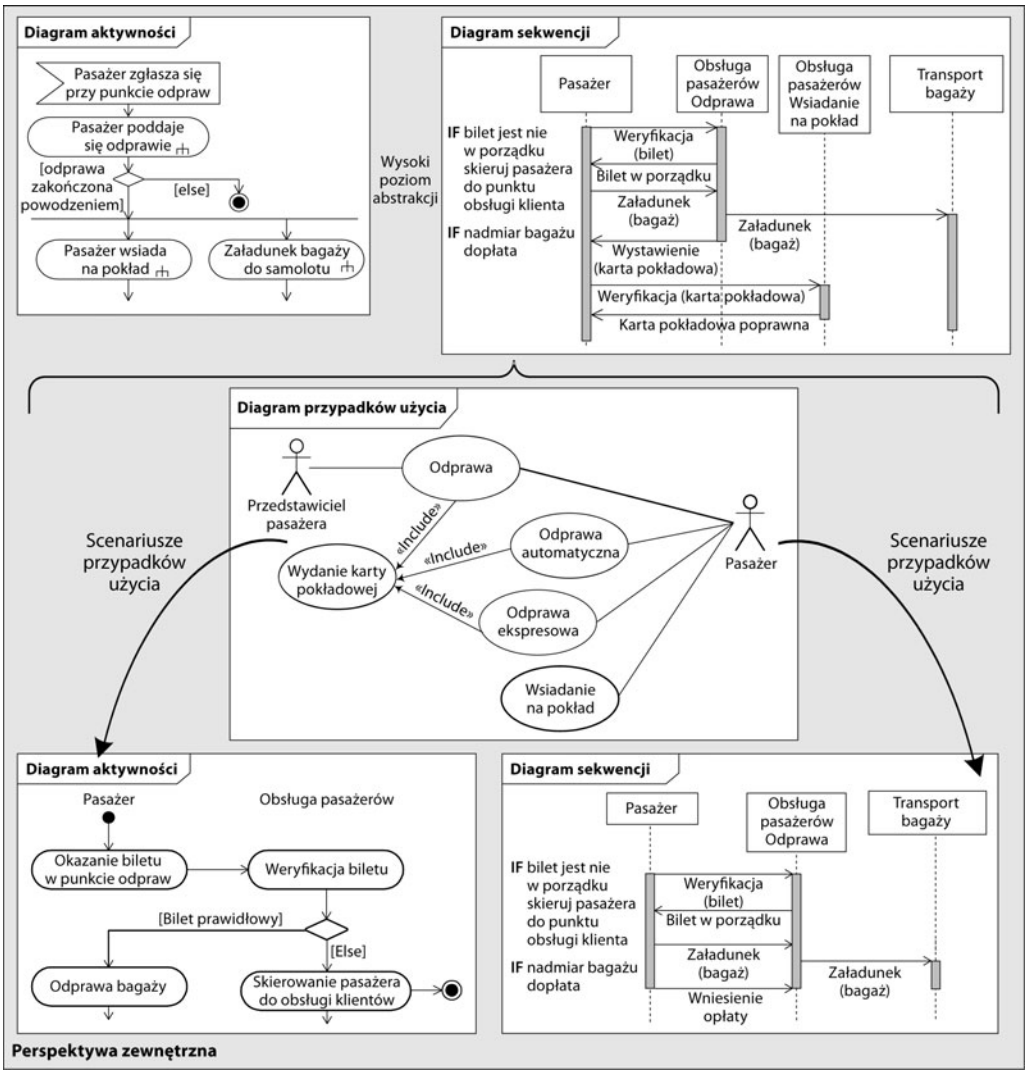
- Które z nich są uczestnikami zewnętrznymi (klient, partner biznesowy itp.) i jakie wykorzystują efekty działania systemu?
- Jakie osoby są oddelegowane do obsługi pasażerów w charakterze pracowników i jakie zadania realizują?
- Jakie systemy informatyczne są zaangażowane w działanie systemu?
- Aby pasażer mógł kupić butelkę whisky w sklepie wolnocłowym, pracownik tego sklepu musi sprawdzić jego kartę pokładową, przyjąć należność za towar, zapakować butelkę do torby i wręczyć paragon. Które z tych aktywności należą do perspektywy wewnętrznej, a które do zewnętrznej?

3.3.2. Elementy perspektywy

Perspektywa zewnętrzna składa się z diagramów następujących typów:

- **Diagramy przypadków użycia** przedstawiają aktorów, biznesowe przypadki użycia oraz ich wzajemne związki. Diagramy przypadków użycia nie opisują procedur. Alternatywne scenariusze również pozostają ukryte. Diagramy tego typu dają dobry ogląd funkcji systemu biznesowego.
- **Diagramy aktywności** opisują procedury, a w naszym przypadku procedury biznesowe obowiązujące w systemie biznesowym. Podmiotami tych diagramów są interakcje między aktorami a systemem biznesowym, czyli towary i usługi oferowane klientom i partnerom biznesowym. Na bazie diagramów aktywności zewnętrzni uczestnicy mogą zidentyfikować sposoby interakcji z systemem biznesowym. Diagramy te są szczególnie użyteczne do ilustrowania sekwencji, alternatyw i zdarzeń odbywających się równolegle. Diagramy aktywności mogą być tworzone na różnych poziomach szczegółowości.
- **Diagramy sekwencji** opisują chronologiczny przebieg interakcji. Nie opisują poszczególnych zdarzeń wraz z ich rozgałęzieniami i zrównoleglenieM operacji, lecz skupiają się na informacji przekazywanej pomiędzy stronami interakcji.

  Te diagramy stanowią dobrą podstawę opisu wymiany informacji między systemem a partnerami i klientami (rysunek 3.6).



Rysunek 3.6. Perspektywa zewnętrzna

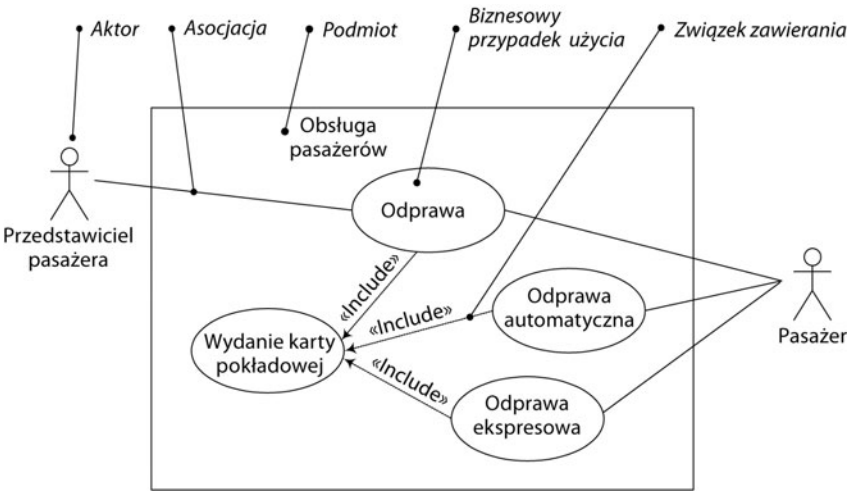
Diagramy UML opisujące biznesowe przypadki użycia mogą być dodatkowo opatrzone opisami i rysunkami uzupełniającymi. Nie trzeba używać za każdym razem wszystkich dostępnych typów diagramów. Ich wybór jest uzależniony od tego, jakie cechy systemu mają zostać podkreślone. Niezależnie od wyboru typów diagramów zalecamy, aby w każdym przypadku tworzyć diagramy przypadków użycia. Ten typ diagramów jest bowiem szczególnie dobrze przystosowany do komunikacji z partnerami systemowymi i ekspertami w dziedzinie funkcji i kontekstu pracy analizowanego systemu. Do tego celu dobrze nadają się również diagramy aktywności na wysokim poziomie abstrakcji (o niskim poziomie szczegółowości).

Na etapie uszczegóławiania biznesowych przypadków użycia i identyfikowania różnych scenariuszy pojawia się potrzeba opisania różnych aktywności i do tego właśnie służy diagram aktywności.

Diagram sekwencji prezentuje drogę wymiany informacji z partnerami i klientami (zobacz rozdział 5., „Modelowanie integracji systemów"). Z naszego doświadczenia wynika, że diagramy sekwencji są dość często stosowane w zadaniach modelowania procesów biznesowych. Powodem tego stanu rzeczy jest fakt, że diagramy te są bardzo czytelne i wykorzystują niewielką różnorodność elementów graficznych. Dopóki na temat zdarzeń technicznych mamy jedynie podstawową wiedzę, diagramy sekwencji są często bardziej od diagramów aktywności przydatne podczas prezentowania wymiany informacji.

3.3.3. Diagramy przypadków użycia

Diagramy przypadków użycia przedstawiają biznesowe przypadki użycia, aktorów i powiązania między nimi. Powiązanie pomiędzy aktorem a biznesowym przypadkiem użycia występuje wówczas, gdy aktor ma możliwość wykorzystania określonej funkcji systemu. Diagramy przypadków użycia nie zawierają żadnej informacji o tym, jaka jest chronologia podejmowanych działań (rysunek 3.7).



Rysunek 3.7. Elementy diagramu przypadku użycia

W diagramach przypadków użycia wykorzystywane są następujące elementy:

Aktor: reprezentuje rolę, jaką zewnętrzny użytkownik systemu przyjmuje w interakcji z systemem biznesowym. Na przykład aktor może być klientem, partnerem biznesowym, dostawcą lub innym systemem.

Każdy aktor ma jakąś nazwę:

Ikonką aktora nie musi być ludzik z patyków, dopuszczalne są również inne symbole — pod warunkiem, że lepiej oddają charakterystykę aktora i w rezultacie pozwalają zbudować bardziej czytelny diagram.

Asocjacja to relacja między aktorem a biznesowym przypadkiem użycia. Asocjacja reprezentuje możliwość wykorzystania przez aktora określonej funkcji systemu biznesowego, czyli właśnie biznesowego przypadku użycia.

Niestety, asocjacja nie zawiera żadnej informacji na temat sposobu użycia danej funkcji systemu. Jeśli biznesowy przypadek użycia ma zdefiniowane asocjacje z kilkoma aktorami, z diagramu nie wynika jednoznacznie, czy aktor może korzystać z tej funkcji samodzielnie, czy też potrzebni są do tego wszyscy aktorzy. Asocjacja oznacza w rzeczywistości jedynie to, że aktor ma „coś wspólnego" z biznesowym przypadkiem użycia.

Biznesowy przypadek użycia opisuje interakcję między aktorem a systemem biznesowym, to znaczy opisuje funkcję systemu biznesowego wykorzystywaną przez aktora:

Biznesowy przypadek użycia jest określany z perspektywy aktora. Jest on uszczegółowieniem przypadku użycia do elementów występujących w systemach biznesowych; nie ma większych różnic merytorycznych między biznesowymi a zwykłymi przypadkami użycia.

Związek zawierania jest to związek między dwoma przypadkami użycia informujący o tym, że biznesowy przypadek użycia wskazywany grotem strzałki jest zawarty w biznesowym przypadku użycia, od którego wychodzi strzałka. Oznacza to, że jakaś funkcja systemu biznesowego jest realizowana w ramach innej, nadrzędnej funkcji systemu biznesowego.

Dzięki temu funkcje, które są wykorzystywane przy różnych okazjach, mogą być wyodrębnione jako osobne biznesowe przypadki użycia:

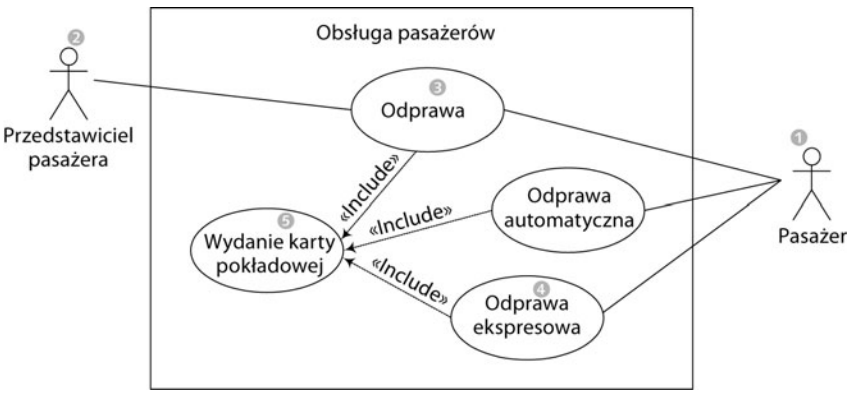
Zwrot strzałki może początkowo być mylący, intuicyjnie oczekuje się bowiem, że grot sugeruje kierunek działania związku (na przykład odprawa zawiera w sobie procedurę wystawiania karty pokładowej).

Podmiot to system biznesowy, w którym występuje jeden lub większa liczba biznesowych przypadków użycia. Podmiot jest reprezentowany przez prostokąt zawierający w sobie występujące w danym systemie przypadki użycia. U góry prostokąta zapisuje się nazwę systemu:

Podmiot (tak jak i określenie zasięgu systemu) jest opcjonalny.

Czytanie diagramów przypadków użycia

Rysunek 3.8 przedstawia diagram przypadków użycia zawierający aktorów takich, jak **pasażer** (1) i **przedstawiciel**, (2) oraz biznesowe przypadki użycia takie, jak **odprawa** (3) i **odprawa ekspresowa** (4).



Rysunek 3.8. Diagram przypadków użycia

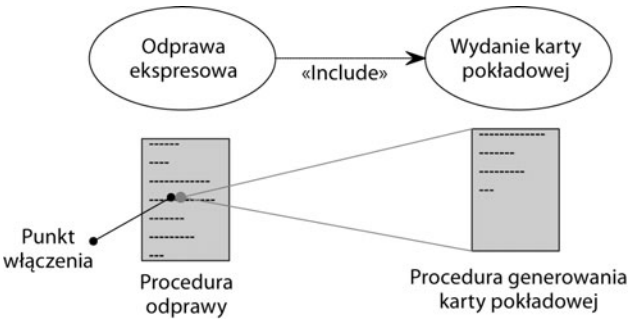
W zależności od potrzeb diagram można czytać, przyjmując za początek aktora lub biznesowy przypadek użycia. Jeśli przyjmiemy za początek aktora **pasażer** (1), znajdziemy dwa związki z biznesowymi przypadkami użycia: będą to **odprawa** (3) i **odprawa ekspresowa** (4). Oznacza to, że pasażerowie mogą dokonać odprawy lub odprawy ekspresowej, która ma miejsce w przypadku, gdy pasażer nie posiada bagażu.

Występowanie na diagramie jednego biznesowego przypadku użycia pod drugim nie ma żadnego znaczenia. Diagram przypadków użycia **nie** dokumentuje kolejności występowania biznesowych przypadków użycia. Oczywiście, kolejność ma znaczenie w przypadku opisu i powiązań procesów biznesowych. Tego typu zagadnienia są dokumentowane za pomocą diagramów aktywności (zobacz punkt 3.3.5, „Diagramy aktywności").

Aktor **przedstawiciel** (2) również ma związek z biznesowym przypadkiem użycia **odprawa** (3). Oznacza to, że nie tylko pasażer, lecz również jego przedstawiciel może poddać się odprawie. Również aktor **pasażer** (1) ma związek z biznesowym przypadkiem użycia **odprawa** (3), **nie** oznacza to jednak, że zarówno pasażer, jak i jego przedstawiciel muszą dokonywać odprawy jednocześnie. Taki fakt (gdyby miał miejsce) można by odczytać z innego diagramu (zobacz punkt 3.3.5, „Diagramy aktywności") lub z komentarza do diagramu, w którym można zawrzeć tekst uzupełniający (na przykład objaśnienie).

Z faktu, że aktor **przedstawiciel** (2) ma związek tylko z biznesowym przypadkiem użycia **odprawa** (3), wynika, że na lotnisku UML przedstawiciel pasażera nie może dokonać **odprawy ekspresowej** (4).

Jak widać, tak proste diagramy mogą zawierać spore dawki informacji. Biznesowy przypadek użycia **odprawa** (3) oraz biznesowy przypadek użycia **odprawa ekspresowa** (4) pozostają w związku zawierania z biznesowym przypadkiem użycia **wydanie karty pokładowej** (5). Oznacza to, że obydwa typy odprawy w pewnym momencie powodują wydanie karty pokładowej (przypadki użycia nie pokazują jednak tego, **w jakim momencie** następują zawarte w nich przypadki użycia). Wiadomo jedynie, że w jednym z etapów odprawy pasażerowi lub jego przedstawicielowi jest wręczana karta pokładowa. Rysunek 3.9 przedstawia kolejną próbę uszczegółowienia tej procedury.



Rysunek 3.9. Związek zawierania jednego przypadku użycia w innym

3.3.4. Konstruowanie diagramów przypadków użycia

Poniższa lista zawiera kroki niezbędne do skonstruowania diagramu przypadków użycia. Każdy z tych kroków opiszemy w dalszej części rozdziału.

Lista kontrolna 3.1. Konstruowanie diagramu przypadków użycia w perspektywie zewnętrznej składa się z poniżej opisanych etapów

- Gromadzenie źródeł informacji — skąd mam to wiedzieć?
- Identyfikowanie potencjalnych aktorów — którzy partnerzy i klienci wykorzystują towary lub usługi świadczone przez system biznesowy?

- Identyfikowanie potencjalnych biznesowych przypadków użycia — które towary i usługi są dostępne dla aktorów?
- Łączenie biznesowych przypadków użycia — kto może wykorzystać towary i usługi świadczone przez system?
- Opis aktorów — kogo lub co reprezentują aktorzy?
- Poszukiwanie większej liczby biznesowych przypadków użycia — co jeszcze musi być zrobione?
- Edycja biznesowych przypadków użycia — jakie informacje muszą być ujęte w biznesowych przypadkach użycia?
- Udokumentowanie biznesowych przypadków użycia — co się dzieje w biznesowym przypadku użycia?
- Modelowanie powiązań między biznesowymi przypadkami użycia — jakie ważne działania odbywają się w systemie?
- Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Kolejność poszczególnych etapów dobraliśmy celowo. Nie jest to jednak kolejność wymagana, ponieważ w praktyce wiele z nich wzajemnie się pokrywa.

Do realizacji każdego z etapów jest wymagane ogólne zrozumienie systemu biznesowego i procesów biznesowych. Wiele z tych etapów wymaga również konsultacji z dostawcami wiedzy. Nie ma sensu ograniczać się do jednostronnego postrzegania systemu, jak ma to miejsce w przypadku analityka, który z reguły nie posiada wyczerpującej wiedzy dotyczącej analizowanego tematu.

Gromadzenie źródeł informacji — skąd mam to wiedzieć?

W pierwszym kroku należy zidentyfikować dostawców wiedzy, ponieważ analityk wraz z dostawcami wiedzy w kolejnych krokach wspólnie będą opracowywać podstawowe zagadnienia modelu. Do dostawców wiedzy można zaliczyć:

- osoby zaangażowane w realizację, obsługę i kontrolę procesów biznesowych;
- użytkowników obsługujących takie same lub podobne systemy;
- klientów, którzy często stanowią doskonałe krytyczne i kreatywne źródła wiedzy;
- partnerów biznesowych;
- ekspertów w analizowanej dziedzinie;
- zarząd;
- obserwatorów zewnętrznych.

Istnieje kilka technik pomocnych podczas analizy i w zrozumieniu procesów biznesowych. Są to:

- obserwacja pracowników przy pracy;
- współudział w analizowanych procesach biznesowych;
- przyjęcie roli uczestnika zewnętrznego (na przykład klienta);
- przeprowadzanie ankiet;
- przeprowadzanie wywiadów;

- organizowanie burz mózgów wśród wszystkich osób biorących udział w projekcie;
- dyskusje z ekspertami;
- analiza istniejących formularzy, specyfikacji, podręczników i narzędzi;
- opisywanie struktury organizacyjnej i zarządzania przepływem informacji;
- analiza diagramów organizacyjnych i opisów stanowisk.

- W rozdziale 2. przedstawiliśmy nasze studium przypadku. Przypomnij je sobie, aby lepiej zrozumieć procesy biznesowe.
- Spróbuj zidentyfikować wszystkie role oraz procesy biznesowe, w których mogą brać udział pasażer, sprzedawca w sklepie wolnocłowym itp.
- Jakie aktywności kojarzą się z pasażerem? W jaki sposób mógłbyś odświeżyć swoją pamięć na ten temat?

W wyniku pierwszego etapu pracy powstaje z reguły zbiór formularzy, instrukcji, gotowych ankiet, istniejących opisów procesów, obiektów biznesowych, takich jak bilet czy karta pokładowa itp. Ten przegląd informacji z reguły będzie jeszcze niekompletny, ale w kolejnych etapach ulegnie uporządkowaniu i skompletowaniu.

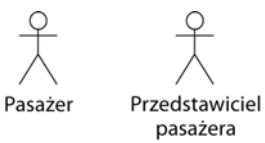
Identyfikowanie potencjalnych aktorów — którzy partnerzy i klienci wykorzystują towary lub usługi świadczone przez system biznesowy?

Ten etap służy do identyfikacji aktorów. Zasada jest prosta: im więcej, tym lepiej. Zidentyfikowani aktorzy będą wykorzystywani na dalszych etapach prac, mogą też być redukowani (usuwani drogą eliminacji lub łączenia).

Przy wyszukiwaniu aktorów można posłużyć się określonymi pytaniami (można na przykład zadawać je dostawcom wiedzy). Podczas identyfikacji aktorów warto grupować ich w zależności od kryteriów wybranych w oparciu o przykłady konkretnych osób i organizacji:

- Którzy klienci wykorzystują system biznesowy, a którzy procesy biznesowe?
- Którzy z użytkowników zewnętrznych są zewnętrznymi partnerami systemu biznesowego? Jakie towary i usługi wykorzystują ci partnerzy?
- Które stanowiska i jednostki organizacyjne wewnątrz firmy są partnerami systemu biznesowego i jakie wykorzystują towary i usługi?
- Z jakimi zewnętrznymi systemami biznesowymi modelowany system wymienia dane?

Na pierwszym etapie analizy w naszym studium przypadku wyodrębniliśmy aktorów przedstawionych na rysunku 3.10.



Rysunek 3.10. Potencjalni aktorzy

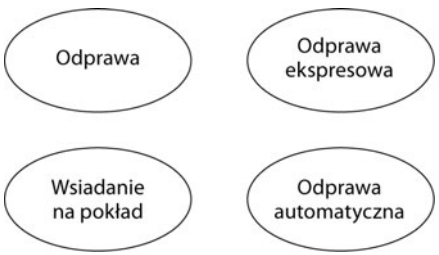
Oprócz aktora **pasażer**, który reprezentuje wszystkich podróżnych, mamy aktora **przedstawiciel pasażera**. Nie jest to pasażer, ponieważ nie odbywa podróży, a jedynie w imieniu pasażera realizuje część formalności związanych z odprawą.

Identyfikowanie potencjalnych biznesowych przypadków użycia — które towary i usługi są dostępne dla aktorów?

Na tym etapie poszukujemy potencjalnych przypadków użycia. Również tutaj ma zastosowanie reguła im więcej, tym lepiej (oczywiście w granicach rozsądku). Potencjalne biznesowe przypadki użycia można znaleźć, odpowiadając na następujące pytania:

- Jakie towary lub usługi są udostępniane klientom?
- Jakie towary lub usługi są udostępniane partnerom zewnętrznym?
- Jakie towary lub usługi udostępniane przez system biznesowy wymagają zewnętrznych dostawców (towarów i usług)?
- Jakie działania podejmują poszczególni aktorzy?
- W jaki sposób i kiedy odbywa się komunikacja z innymi systemami biznesowymi lub partnerami biznesowymi?
- Jakie zdarzenia wywołują określone aktywności?

Na pierwszym etapie analizy naszego studium przypadku wyodrębniliśmy przypadki użycia przedstawione na rysunku 3.11.



Rysunek 3.11. Potencjalne biznesowe przypadki użycia

W tym momencie przypadki użycia mogą być opisane jedynie w ogólny i nieformalny sposób:

- Procedura **odprawy** wiąże się z wydaniem biletu, odprawieniem bagażu oraz wydaniem pasażerowi karty pokładowej.
- Pasażer posiadający jedynie bagaż podręczny może skorzystać z **odprawy ekspresowej**. Nie odbywa się wtedy odprawa bagażu.
- Przed **wejściem na pokład** karta pokładowa pasażera jest sprawdzana w bramce.
- **Odprawa automatyczna** jest przeprowadzana bez udziału pracownika odpraw, pasażer przeprowadza ją samodzielnie przy użyciu komputera. W ramach odprawy automatycznej nie ma możliwości odprawienia bagażu.

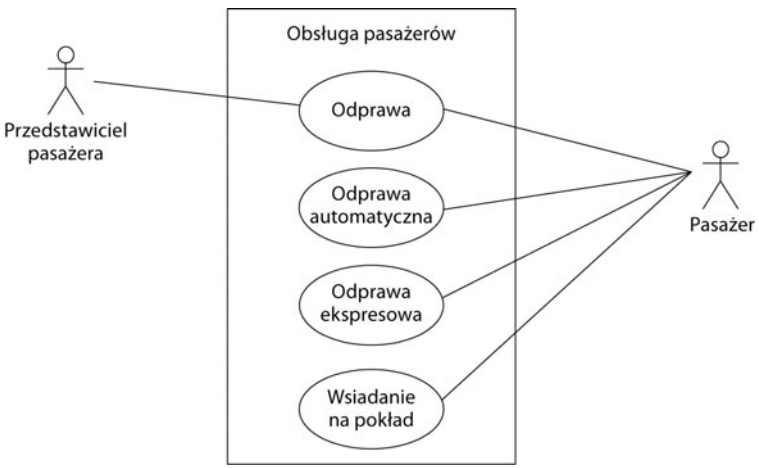
Porady praktyczne

W praktyce okazuje się, że najbardziej efektywna przy identyfikacji przypadków użycia jest technika obserwacji. Dzięki obserwacji osób zaangażowanych w proces biznesowy można sporządzić listy aktywności, które następnie mogą być grupowane w zdarzenia tworzące pierwszy zarys biznesowych przypadków użycia.

Łączenie biznesowych przypadków użycia — kto może wykorzystać towary i usługi świadczone przez system?

Poprzez przyporządkowanie biznesowych przypadków użycia odpowiednim aktorom tworzy się pierwszy szkic diagramu przypadków użycia (rysunek 3.12). Podczas tego procesu użyteczne okazuje się udzielanie odpowiedzi na pytanie:

- Do jakich funkcji systemu mają dostęp klienci lub partnerzy biznesowi?



Rysunek 3.12. Pierwszy szkic diagramu przypadków użycia

W oparciu o ten szkic możemy prowadzić dalsze prace porządkujące i udoskonalające diagram przypadków użycia.

Pasażer może wybrać odprawę zwykłą, automatyczną lub ekspresową. Podchodzi on do bramki i pokazuje swoją kartę pokładową. Przedstawiciel pasażera dokonuje zwykłej odprawy, nie może jednak poddać się odprawie automatycznej ani ekspresowej.

Opis aktorów — kogo lub co reprezentują aktorzy?

Aktor w diagramie musi mieć nazwę informującą o jego roli. Kluczowe znaczenie ma tu zastosowanie terminologii zrozumiałej dla biznesowych członków zespołu. Oprócz nazwy aktor może mieć opis. W tym przypadku warto udzielić odpowiedzi na pytanie:

- W jaki sposób można opisać aktora?

Opis może obejmować obszar obowiązków i wymagań ze strony systemu lub formalną definicję roli osoby. Nie ma powodu unikać umieszczania w tym miejscu opisów stanowisk lub profilu organizacyjnego (na przykład w przypadku firmy kateringowej), nawet wówczas, gdy te szczegóły nie są ujęte w diagramach UML.

Poszukiwanie większej liczby biznesowych przypadków użycia — co jeszcze musi być zrobione?

Gdy zostanie zidentyfikowanych kilka pierwszych biznesowych przypadków użycia, można je wykorzystać jako punkty wyjścia do dalszych pytań. Biorąc za podstawę określony biznesowy przypadek użycia, odpowiadamy na następujące pytania:

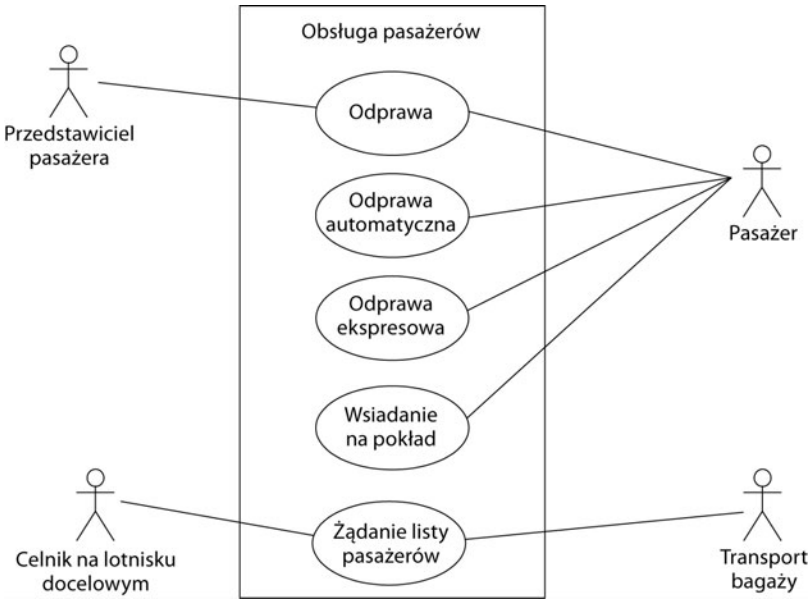
- Czy przed wykorzystaniem określonej funkcji należy zapewnić jakieś dodatkowe warunki wejściowe?
- Czy po wykonaniu określonego biznesowego przypadku użycia należy wykonać jakieś dodatkowe działania?
- Czy należy wykonać jakiekolwiek działania w przypadku, gdy dany biznesowy przypadek użycia nie będzie wykonany?

Przy powyższej analizie należy uwzględnić odpowiedni system biznesowy. Wiele zdarzeń występujących przed biznesowym przypadkiem użycia lub po nim ma miejsce poza modelowanym systemem biznesowym. W naszym studium przypadku czynnościami niezwiązanymi z modelowanym systemem są na przykład rezerwacja biletu lub przyjazd na lotnisko.

Należy zauważyć, że pasażer często podróżuje z bagażem, który należy odprawić. Za załadunek bagażu na pokład samolotu odpowiedzialny jest dział transportu bagażu. Zadanie to jest realizowane przez niezależną jednostkę organizacyjną, nazwaną u nas komórką transportową. Komórka ta jest aktorem, a dokładniej — zewnętrznym dostawcą usługi. W naszym diagramie nie ma znaczenia, że poszczególne zadania są realizowane przez różnych pracowników działu transportu bagaży.

Dziesięć minut przed odlotem komórka transportowa oczekuje od działu obsługi pasażerów dostarczenia listy, na której znajdują się nazwiska wszystkich pasażerów, którzy dokonali odprawy, lecz nie wsiedli na pokład samolotu. Na jej podstawie bagaże tych osób są wyładowywane z samolotu. Jeśli lot jest międzynarodowy, listy pasażerów oczekują również służby celne kraju docelowego.

Ta sytuacja wymaga wzięcia pod uwagę dwóch nowych aktorów: komórki transportowej i służb celnych lotniska docelowego (rysunek 3.13).



Rysunek 3.13. Rozszerzony diagram przypadku użycia

Edycja biznesowych przypadków użycia — jakie informacje muszą być ujęte w biznesowych przypadkach użycia?

Niewątpliwie trudno jest oszacować właściwy zakres szczegółowości w modelach systemów biznesowych. Jeśli prawie wszystkie aktywności aktora zostaną połączone w jeden diagram biznesowego przypadku użycia, diagram ów straci znaczenie. Jeśli aktywności zostaną odwzorowane zbyt szczegółowo, diagram stanie się skomplikowany (będzie miał za dużą liczbę aktywności i powiązań między nimi).

Na szczęście istnieją pewne kryteria, które ułatwiają wybór optymalnego poziomu szczegółowości biznesowego przypadku użycia. Ponownie należy zadać sobie kilka pytań:

- Czy biznesowy przypadek użycia składa się z sekwencji interakcji nieodłącznie powiązanych wzajemnie?

  Elementy wchodzące w skład biznesowego przypadku użycia muszą być wzajemnie powiązane, zaś na przykład wydanie karty pokładowej i poszukiwanie bagażu nie są ze sobą w ogóle związane. Biznesowe przypadki użycia niespełniające zasady wzajemnego powiązania należy rozbić na składowe. Dzięki temu można uniknąć tworzenia nadmiernie rozbudowanych diagramów.
- Jak wielu aktorów bierze udział w biznesowym przypadku użycia? Biznesowe przypadki użycia zawierające zbyt wielu aktorów muszą zostać podzielone. Dzięki temu również unika się tworzenia zbyt rozbudowanych diagramów.

- Czy biznesowy przypadek użycia dostarcza konkretne towary lub usługi? Nie powinien on opisywać etapów pośrednich, na przykład liczenia sztuk bagażu. W większości klasycznych przypadków wymaga się, aby biznesowy przypadek użycia generował konkretną korzyść z punktu widzenia klienta. Jeśli nie spełnia tej zasady, musi być połączony z innymi. Dzięki temu unika się zbytniego rozdrobnienia.
- Czy biznesowy przypadek użycia jest zawsze wykonywany w sekwencji z innymi? Nie powinien on generować towarów lub usług, które mogą być użyte tylko w połączeniu z wynikami działania innych biznesowych przypadków użycia. Jeśli nie spełnia tej zasady, musi być połączony z innymi. Dzięki temu unika się zbytniego rozdrobnienia.
- Czy biznesowy przypadek użycia jest zainicjowany przez aktora? Jeśli nie jest zainicjowany przez aktora, nie jest przypadkiem użycia, lecz wewnętrzną aktywnością, którą reprezentuje się w ramach wewnętrznej perspektywy systemu biznesowego.

Przegląd istniejących biznesowych przypadków użycia pod kątem powyższych pytań może doprowadzić do konsolidacji niektórych z nich i podziału innych.

Udokumentowanie biznesowych przypadków użycia — co się dzieje w biznesowym przypadku użycia?

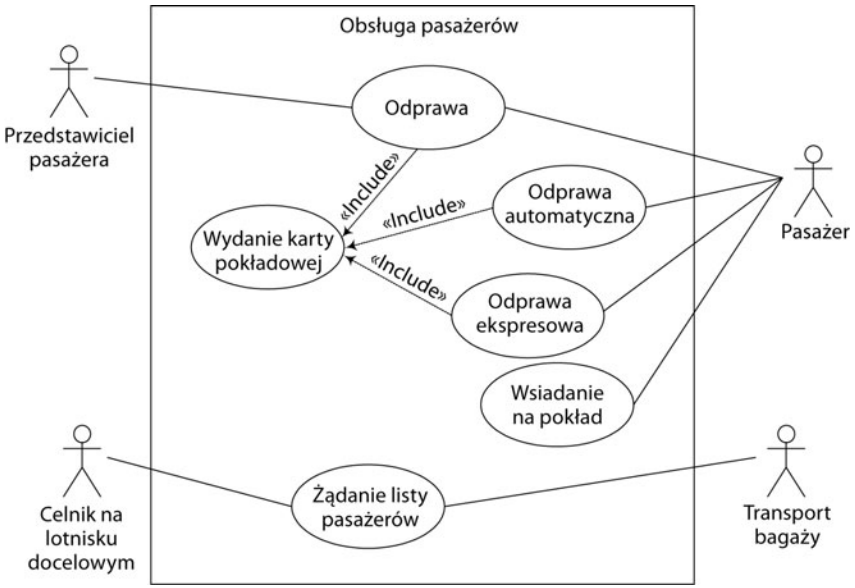
Aby zrozumieć przypadek użycia, nie wystarczy sama informacja zaczerpnięta z diagramu. Należy opisać łańcuch iteracji i różnych warunków wykonania, które cechują każdy z biznesowych przypadków użycia. Oznacza to, że musi być opisany łańcuch zdarzeń poprzedzający dostarczenie przez system każdego towaru lub usługi. Opis ten musi być wykonany z punktu widzenia klienta lub partnera biznesowego.

Oprócz czysto tekstowego opisu bardzo użyteczna okazuje się dokumentacja w postaci diagramów aktywności oraz diagramów sekwencji. Konstrukcja tych diagramów jest omówiona w punktach 3.3.5, „Diagramy aktywności" oraz 3.3.9, „Diagramy sekwencji o wysokim poziomie abstrakcji".

Modelowanie powiązań między biznesowymi przypadkami użycia — jakie ważne działania odbywają się w systemie?

Jeśli okaże się, że pewne elementy iteracji są wspólne w kilku różnych biznesowych przypadkach użycia, można je wyodrębnić i na ich podstawie zbudować indywidualne przypadki użycia. Nowy biznesowy przypadek użycia utworzony w oparciu o tę zasadę może być połączony w innym za pomocą **związku zawierania**.

W naszym studium przypadku biznesowy przypadek użycia o nazwie **wydanie karty pokładowej** nie został jeszcze przydzielony. Wiemy, że karta pokładowa jest wydawana w ramach odprawy. W pewnym momencie takich biznesowych przypadków użycia, jak odprawa zwykła, ekspresowa oraz automatyczna pasażer otrzymuje kartę pokładową (rysunek 3.14).



Rysunek 3.14. Rozszerzony diagram przypadku użycia

Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Wszystkie diagramy i zapisy muszą być zweryfikowane przez dostawców wiedzy. Należy zadać im pytanie:

- Czy informacje umieszczone na diagramie są kompletne?

Nawet w przypadku, gdy dostawcy wiedzy są w stanie przeczytać i zrozumieć diagram bez pomocy (aby go odczytać, mogą sięgnąć po wskazówki, na przykład po tę książkę), twórca diagramu powinien i tak omówić go swoimi słowami. Dopiero taka weryfikacja poprawności diagramu pozwala zamknąć cykl jego tworzenia. Wynikiem tego etapu jest zweryfikowana perspektywa, która odzwierciedla aktualny stan pojmowania systemu biznesowego i jego procesów.

Gotowy diagram przypadków użycia można zweryfikować za pomocą poniższej listy kontrolnej.

Lista kontrolna 3.2. Weryfikacja diagramu przypadków użycia w perspektywie zewnętrznej

- **Kompletność** — diagram przypadków użycia jest kompletny, jeśli w systemie nie ma już żadnych nieudokumentowanych biznesowych przypadków użycia. Wszystkie towary i usługi dostępne dla klienta i partnerów biznesowych są opisane w formie biznesowych przypadków użycia (jeśli to konieczne, mogą one być rozpisane w postaci kilku podległych diagramów).
- **Zgodność** — wszystkie biznesowe przypadki użycia uwzględnione na diagramie przypadków użycia są rzeczywiste, co oznacza, że są zgodne z definicją biznesowego przypadku użycia.

- **Poziom szczegółowości** — szczegółowość biznesowych przypadków użycia spełnia następujące wymogi:
  - biznesowy przypadek użycia reprezentuje spójną z punktu widzenia działań sekwencję iteracji;
  - biznesowy przypadek użycia jest zainicjowany przez aktora i wykorzystuje tylko kilku aktorów.
  - biznesowy przypadek użycia reprezentuje funkcję dostępną dla odbiorców i dającą wymierne wyniki.
- **Związki między przypadkami użycia** — związki zawierające są zdefiniowane właściwie.
- **Nazewnictwo i opisy** — nazwy biznesowych przypadków użycia opisują funkcje systemu biznesowego. Terminologia jest dobrana zgodnie z obowiązującą w ramach systemu biznesowego.
- **Aktorzy** — aktorzy w diagramie przypadków użycia odpowiadają rolom przyjmowanym w interakcjach przez uczestników zewnętrznych, organizacje lub inne systemy biznesowe.

Wskazówki praktyczne

Wykorzystując biznesowe przypadki użycia do modelowania systemów i procesów biznesowych, należy pamiętać o tym, że poziom abstrakcji powinien być tu stosunkowo niski. Aby zapewnić wysoki poziom zrozumiałości diagramów i jakości komunikacji między członkami zespołu pracującego nad danym projektem, lepiej uczynić ten projekt nieco nadmiarowym, niż za mocno opierać się na abstrakcji.

Kwestią zasadniczą jest tu zastosowanie terminologii używanej w już istniejących procesów biznesowych lub obowiązującej w danej organizacji. Dzięki temu opis systemu biznesowego oraz przypadków użycia będzie intuicyjnie zrozumiały.

Terminologia informatyczna nie nadaje się do stosowania w diagramach użycia na poziomie procesów biznesowych. Mieszanie pojęć procesów biznesowych z informatycznymi daje z reguły kiepskie wyniki. W rzeczywistości często napotykamy diagramy biznesowych przypadków użycia zbytnio zbliżone w swej szczegółowości do poziomu systemu informatycznego. To prowadzi do dwóch problemów:

- Użytkownicy (czyli osoby zaangażowane w proces biznesowy i niezorientowane w terminologii informatycznej) nie zrozumieją biznesowych przypadków użycia. W związku z tym, że te przypadki użycia opisują działanie systemu biznesowego, a opis nie jest zrozumiały dla jego użytkowników, model systemu nie może być zweryfikowany pod kątem poprawności. Projekt z błędnie zdefiniowanymi biznesowymi przypadkami użycia zaprezentowany użytkownikom w celu weryfikacji z reguły spotyka się z następującym komentarzem: „Czy na tym rysunku mamy oceniać ludziki rzucające strzałami?”.
- Szczegóły techniczne na poziomie biznesowych przypadków użycia będą rozbieżne z biznesowymi wymogami zdefiniowanymi dla systemu.

3.3.5. Diagramy aktywności

Diagramy aktywności mają związek z diagramami przepływów i służą do ilustrowania działań zachodzących w systemie. W perspektywie zewnętrznej diagramy aktywności są wykorzystywane do opisu procesów biznesowych realizujących funkcje systemu biznesowego.

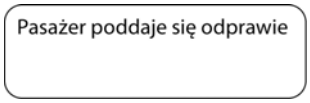
W przeciwieństwie do diagramów przypadków użycia diagramy aktywności jednoznacznie informują o tym, czy poszczególni aktorzy biorący udział w procesie wykonują określone działania samodzielnie, czy wspólnie z innymi.

Diagramy aktywności pozwalają spojrzeć na system pod kątem pełnionych w nim funkcji. Puryści podejścia obiektowego zapewne nie są z tego zadowoleni. Z naszego doświadczenia wynika jednak, że ta cecha jest dużą zaletą takich diagramów. Zastosowanie metod zorientowanych obiektowo wraz z wzorcami programowania funkcyjnego pozwala bowiem niezależnie od poziomu abstrakcji odbiorcy przekazać informacje w zrozumiały i intuicyjny sposób. To jest szczególnie ważne na etapie modelowania procesów biznesowych.

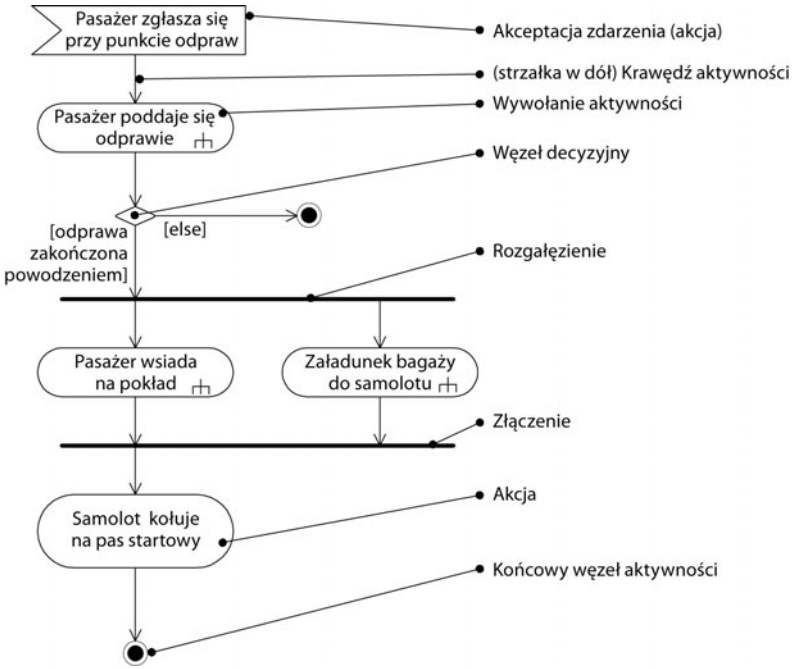
W diagramach aktywności można zaprezentować zdarzenia zachodzące równolegle. Dlatego właśnie są one szczególnie przydatne w modelowaniu procesów biznesowych — te ostatnie rzadko odbywają się w sposób sekwencyjny i nader często wykorzystują równoległe wykonywanie działań.

Diagramy aktywności mogą być tworzone na różnych poziomach szczegółowości. Można je też stopniowo uszczegóławiać. W perspektywie zewnętrznej diagramy te, podobnie jak diagramy przypadków użycia, reprezentują procesy biznesowe w ujęciu postrzeganym przez uczestników zewnętrznych. Uszczegółowianie diagramu nie wiąże się więc z uzupełnianiem go o szczegóły wewnętrzne systemu, ponieważ oznaczałoby to niekorzystne dla czytelności wprowadzanie elementów perspektywy wewnętrznej (rysunek 3.15).

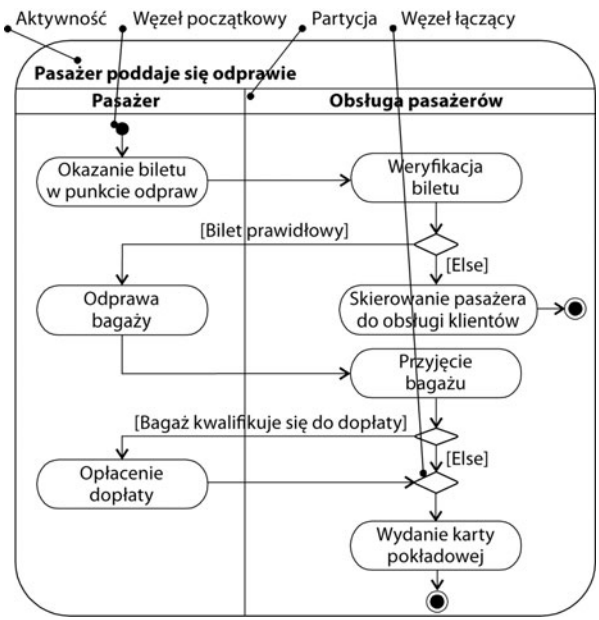
Aktywność — diagram aktywności ilustruje jedną indywidualną aktywność. W naszym kontekście aktywność identyfikuje proces biznesowy (rysunek 3.16). Podstawowymi elementami aktywności są akcje i elementy kontrolne (decyzja, podział, złączenie, zainicjowanie, zakończenie itp.):



Element są połączone w tzw. krawędzie aktywności (ang. *activity edge*) i tworzą przepływ (ang. *control flow* lub po prostu *flow*). Wykonanie aktywności może zawierać kilka równoległych przepływów. Wokół aktywności można narysować ramkę odzwierciedlającą granice diagramu aktywności.



Rysunek 3.15. Diagram aktywności „obsługa pasażerów" o niskim poziomie szczegółów (a więc o wysokim poziomie abstrakcji)

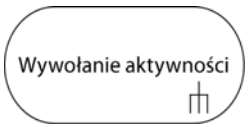


Rysunek 3.16. Diagram aktywności „pasażer dokonuje odprawy"

Akcja — pojedynczy krok aktywności, na przykład etap obliczeń, który nie musi (lub nie może) być rozłożony na mniejsze elementy. Nie oznacza to, że w rzeczywistości akcja nie może być podzielona na mniejsze elementy. Znaczy to jedynie tyle, że na potrzeby diagramu dokonywanie takiego podziału nie miałoby sensu:

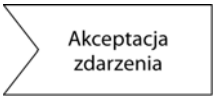
Akcja może zawierać informacje wejściowe i wyjściowe. Wyjście z akcji może stanowić wejście do innej akcji w ramach danej aktywności. Określone akcje mogą być związane z wywoływaniem innych akcji, obsługą zdarzenia i przekazywaniem sygnałów.

Wywołanie aktywności (akcja) — ten symbol informuje, że dana aktywność wywołuje inną aktywność. Wywołanie jest samo w sobie akcją, wynikiem wywołania jest podjęcie innej aktywności:



W ten sposób aktywności mogą być wzajemnie zagnieżdżane, co pozwala na reprezentację wielu poziomów szczegółowości.

Akceptacja zdarzenia (akcja) — tego typu akcja jest związana z oczekiwaniem na zdarzenie. Po zaakceptowaniu zdarzenia inicjowany jest przepływ z niej wynikający (zdefiniowany na diagramie aktywności). Akceptacja zdarzeń jest ważnym elementem procesów biznesowych w diagramach aktywności.



Wiele procesów biznesowych jest inicjowanych przez zdarzenia; może to być na przykład przetwarzanie zamówienia, przyjęcie zamówienia lub dostawa po otrzymaniu należności.

Akceptacja zdarzenia czasowego (akcja) — w określonym punkcie czasu rozpoczyna się zadana akcja wywołująca przepływ na diagramie aktywności. Do reprezentacji akceptacji zdarzenia czasowego może być użyty symbol klepsydry:

Typowym przykładem akceptacji zdarzenia czasowego może być wysyłka ponaglenia po upłynięciu terminu płatności. Więcej informacji na ten temat można znaleźć w rozdziale 5.

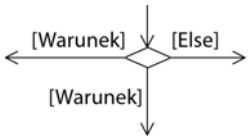
Wysyłka sygnału (akcja) — sygnały są wysyłane do innych aktywności.

Aktywność odbierająca sygnał ma zdefiniowaną akcję typu „akceptacja zdarzenia" i może po odebraniu sygnału podejmować określone reakcje.

Krawędź (kontrola przepływu) — krawędzie są reprezentowane przez strzałki łączące poszczególne elementy diagramu aktywności. Grot strzałki informuje o kierunku przepływu aktywności.

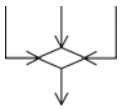
Strzałka wpływająca do aktywności rozpoczyna jej działanie, po którego zakończeniu przepływ jest kontynuowany przez strzałkę wychodzącą. Krawędzi przepływu można przypisać nazwę (etykieta w pobliżu strzałki).

Węzeł decyzyjny — poniższy romb reprezentuje takie miejsce w przepływie, w którym musi być rozstrzygnięty jakiś warunek. Węzeł decyzyjny składa się z wejścia i dwóch lub większej liczby wyjść.



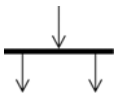
Każde wyjście z węzła ma zdefiniowany warunek zapisany w nawiasach kwadratowych. Jeśli jest on spełniony, przepływ jest kontynuowany przez związane z nim wyjście. Wyjście typu **[else]** definiuje przepływ wykonywany w przypadku niespełnienia żadnego z warunków.

Węzeł łączący — poniższy romb reprezentuje takie miejsce w przepływie, w którym łączy się kilka krawędzi. Węzeł łączący posiada kilka wejść i jedno wyjście.



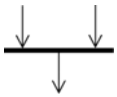
Zadaniem węzła łączącego jest łączenie kilku przepływów. Wejścia nie są zsynchronizowane — jeśli jeden przepływ dotrze do węzła, przechodzi do wyjścia niezależnie od tego, czy do pozostałych wejść dotarły inne przepływy.

Rozgałęzienie — służy do rozdzielenia przepływu na dwa lub większą liczbę równoległych przepływów. Rozgałęzienie jest reprezentowane przez grubą, poziomą linię (belkę).



Rozgałęzienie pozwala definiować równoległe przepływy w ramach aktywności. Posiada ono jedno wejście i dwa lub większą liczbę wyjść.

Złączenie — służy do konsolidacji dwóch lub większej liczby równoległych przepływów. Również w tym przypadku wykorzystuje się grubą, poziomą linię.



W przypadku konsolidacji przepływy są synchronizowane, a więc proces przechodzi do wyjścia dopiero wówczas, gdy dotrą one do **wszystkich** wejść. Złączenie ma dwa lub większą liczbę wejść i jedno wyjście.

Węzeł początkowy — sygnalizuje początek aktywności. Aktywność może mieć więcej niż jeden węzeł początkowy; w takim przypadku na początku aktywności uruchamianych jest kilka przepływów.

Aktywność nie musi mieć węzła początkowego; może być także inicjowana przez zdarzenie (akcja akceptująca zdarzenie).

Końcowy węzeł aktywności — sygnalizuje zakończenie aktywności. Diagram aktywności może mieć więcej niż jedno wyjście oznaczane węzłami końcowymi.

Jeśli w ramach aktywności występuje kilka równoległych przepływów, wszystkie kończą działanie wraz z dotarciem aktywności do końcowego węzła.

Końcowy węzeł przepływu — kończy działanie przepływu. W przeciwieństwie do końcowego węzła aktywności, który kończy całą aktywność, dotarcie do końca przepływu nie ma wpływu na inne przepływy, działające równolegle.

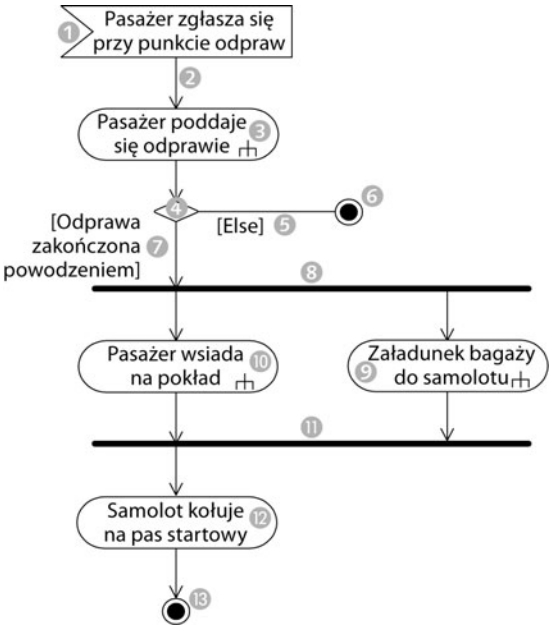
Dzięki temu równoległe przepływy mogą kończyć swoje działanie niezależnie od pozostałych.

Partycja aktywności — poszczególne elementy diagramu aktywności mogą być podzielone na indywidualne obszary zwane partycjami. Przy tworzeniu partycji brane są pod uwagę różne kryteria, takie jak jednostki organizacyjne, centra kosztowe, lokalizacje itp.

Partycjom można przypisać różne etapy aktywności. Każda z nich jest odseparowana od sąsiednich za pomocą pionowych lub poziomych linii ciągłych. Każda posiada nazwę. Partycje mogą być pokazywane w postaci siatki na płaszczyźnie.

Czytanie diagramów aktywności

Diagram czyta się, począwszy od **węzła początkowego** lub akceptacji zdarzenia — jak na rysunku 3.17, na którym aktywność rozpoczyna się od akceptacji zdarzenia **pasażer zgłasza się w punkcie odpraw** (1). Kontynuacja tego diagramu odbywa się wzdłuż strzałek przepływu (2). Kolejna akcja — **pasażer poddaje się odprawie** (3) — oznacza, że w tym miejscu zachodzi aktywność „pasażer poddaje się odprawie". Szczegóły tej aktywności są opisane na innym diagramie, co sygnalizuje symbol rozgałęzienia.

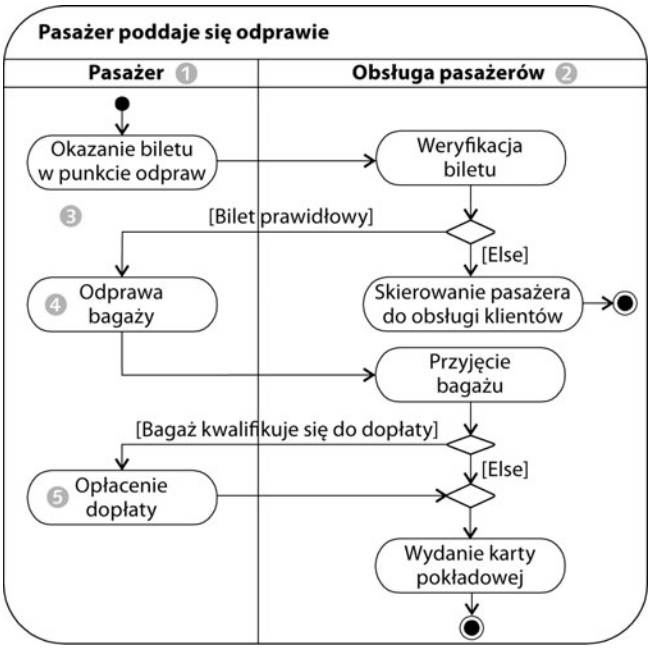


Rysunek 3.17. Diagram aktywności

W dalszej kolejności dojdziemy do rozgałęzienia, czyli węzła decyzyjnego (4). Jeśli odprawa zakończy się pomyślnie, możemy kontynuować przebieg zgodnie z przepływem. W przeciwnym wypadku (5) pasażer nie może odbyć lotu i zadanie obsługi pasażerów w tym miejscu się kończy. Jest to sygnalizowane czarną kropką z obwódką, czyli końcowym węzłem aktywności.

Po pomyślnym zakończeniu odprawy (7) docieramy do belki. Wszystkie strzałki wychodzące z tej belki (7) symbolizują przepływy realizowane równolegle. W trakcie załadunku bagażu (9) pasażer wsiada na pokład (10). Między punktami (8) i (11) przepływy są wzajemnie niezależne. Druga belka (11) łączy równoległe przepływy (9 i 10), co oznacza, że dalszy przepływ ma miejsce dopiero po tym, jak pasażer (10) **i** bagaż (9) znajdą się na pokładzie. W naszym przykładzie dodatkowo wyodrębniono jeszcze jedno działanie (12) oraz węzeł końcowy (13). Oznacza to, że po wejściu pasażerów na pokład (10) i załadowaniu ich bagaży (9) samolot może zacząć kołowanie na pas startowy. Z diagramu wynika, że kołowanie (12) jest pojedynczą akcją, choć to skomplikowany proces, na który mogłoby złożyć się wiele innych diagramów. W kontekście perspektywy zewnętrznej nie ma jednak potrzeby (a nawet nie należy) opisywać tej akcji w sposób szczegółowy.

Diagram aktywności przedstawiony na rysunku 3.18 jest podzielony na dwie partycje: **pasażer** (1) i **obsługa pasażerów** (2). Pasażer wykonuje akcje **okazanie biletu w punkcie odpraw** (3), **odprawa bagażu** (4) i **opłacenie dopłaty** (5) za nadbagaż. Wszystkie pozostałe akcje odbywają się w partycji obsługi pasażerów (2) i są realizowane przez komórkę obsługi pasażerów.



Rysunek 3.18. Diagram aktywności z partycjami

3.3.6. Konstruowanie diagramów aktywności

Jak zwykle chcemy podkreślić, że do uzyskania jak najlepszej czytelności diagramów aktywności kluczowe jest zastosowanie odpowiedniej terminologii. Nie należy załamywać się, gdy dobór odpowiedniego słownictwa zajmuje długie godziny — w większości przypadków efekt wart jest tego wysiłku.

Konstruowanie diagramów proponujemy zacząć od diagramów aktywności o niskim poziomie szczegółowości (czyli o wysokim poziomie abstrakcji), obejmujących po kilka biznesowych przypadków użycia. Dzięki temu można uzyskać dobry przegląd łańcucha powiązań między klientami i partnerami a systemem biznesowym.

Następnie można wprowadzić kolejne szczegóły i scenariusze biznesowych przypadków użycia. Jeśli biznesowy przypadek użycia jest złożony z kilku scenariuszy, każdy z nich musi być odzwierciedlony w diagramie aktywności.

Poniższa lista kontrolna zawiera etapy niezbędne przy konstruowaniu diagramów aktywności.

Lista kontrolna 3.3. Konstruowanie diagramów aktywności w perspektywie zewnętrznej

- Gromadzenie źródeł informacji — skąd mam to wiedzieć?
- Wyszukiwanie aktywności i akcji — co należy zrobić, gdy aktorzy zainicjują dostarczenie towarów lub usług?
- Przejęcie aktorów z biznesowych przypadków użycia — kto jest odpowiedzialny za jakie akcje?
- Połączenie akcji — w jakiej kolejności są one przetwarzane?
- Udoskonalanie aktywności — czy powinny być dodane inne diagramy aktywności?
- Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Kolejność powyższych etapów została dobrana celowo. Jednakże nie jest ona obowiązkowa, ponieważ w praktyce większość z tych etapów dość mocno nakłada się na siebie.

Gromadzenie źródeł informacji — skąd mam to wiedzieć?

W procesie tworzenia diagramów aktywności można wykorzystać wiedzę zgromadzoną do tej pory przy okazji tworzenia diagramów przypadków użycia. Na tym etapie można z powodzeniem zastosować te same porady, co w punkcie 3.3.4, „Konstruowanie diagramów przypadków użycia".

Wyszukiwanie aktywności i akcji — co należy zrobić, gdy aktorzy zainicjują dostarczenie towarów lub usług?

W tym przypadku jako punkt wyjścia również można wykorzystać diagramy przypadków użycia, z których da się wydobyć niezbędne informacje. Aktywności i akcje można identyfikować, odpowiadając na następujące pytania:

- Jakie etapy można wyodrębnić w biznesowych przypadkach użycia, czyli jakie kroki należy wykonać, aby dostarczyć towary i usługi?
- Co robi każdy z aktorów?
- Jeśli w danym biznesowym przypadku użycia występuje kilku aktorów, które z wyodrębnionych etapów są realizowane przez każdego z nich?
- Które zdarzenia inicjują poszczególne etapy pracy?
- Jakie akcje są na tyle skomplikowane, żeby poświęcić im osobne diagramy aktywności?

W naszym studium przypadku w ramach obsługi pasażerów można wyodrębnić następujące etapy działań:

- Odprawa pasażera (etap pozyskany z diagramu przypadku użycia). Etap ten jest związany z wydaniem karty pokładowej.
- Pasażer wsiada na pokład samolotu (etap pozyskany z diagramu przypadku użycia).

Oprócz powyższych można wyróżnić następujące etapy i zdarzenia:

- Pasażer stawia się w punkcie odpraw i okazuje swój bilet. To zdarzenie inicjuje aktywność odprawy.
- Bagaże są ładowane na pokład samolotu przez komórkę transportową.

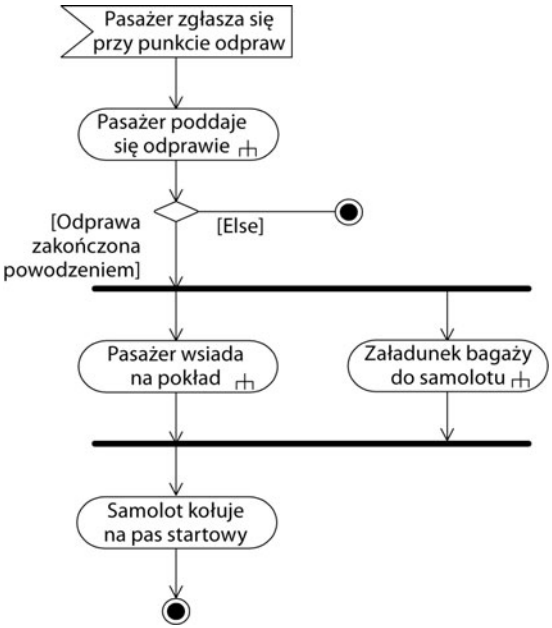
Na pierwszym etapie prac aktywności mogą być opisane w sposób nieformalny, podobnie jak to ma miejsce w kwestii przypadków użycia. W organizacji często dostępna bywa gotowa dokumentacja procesów, nieformalna lub ustrukturalizowana, którą można wykorzystać jako podstawę przy pracach nad identyfikacją aktywności i akcji.

Połączenie akcji — w jakiej kolejności są one przetwarzane?

Przez połączenie wspomnianych wyżej akcji i aktywności w przepływ tworzony jest pierwszy zrąb diagramu aktywności. Ten przepływ określa się terminem **przepływu sterowania**. W jego identyfikacji pomocne mogą okazać się odpowiedzi na następujące pytania:

- W jakiej kolejności są przetwarzane akcje?
- Jakie warunki należy spełnić, aby akcja mogła być wykonana?
- W których miejscach niezbędne są rozgałęzienia?
- Które akcje odbywają się równolegle?
- Czy kompletne wykonanie akcji jest niezbędne, zanim przepływ przejdzie do kolejnych akcji?

Pasażer poddaje się odprawie, **pasażer wsiada na pokład**, **ładowanie bagaży na pokład samolotu** — to przykłady aktywności złożonych. Każda z nich jest opisana w osobnym diagramie aktywności. Sygnalizuje to symbol rozgałęzienia w ikonkach akcji (rysunek 3.19).



Rysunek 3.19. Diagram aktywności na wysokim poziomie abstrakcji, obejmujący kilka przypadków użycia

Udoskonalanie aktywności — czy powinny być dodane inne diagramy aktywności?

Jak widać na rysunku 3.19, niektóre z etapów procesów trzeba poprawić. Chcielibyśmy na przykład nieco bardziej szczegółowo zaprezentować aktywność **pasażer poddaje się odprawie**.

Gdy pasażer poddaje się odprawie, w pierwszej kolejności pokazuje swój bilet przy stanowisku odpraw. Bilet jest sprawdzany pod kątem ważności. Jeśli jest on ważny, pasażer odprawia swój bagaż. Jeśli bagaż ma zbyt dużą wagę, pasażer musi wnieść dodatkową opłatę. Następnie bagaż jest przekazywany do działu transportu bagaży. Pasażer otrzymuje swoją kartę pokładową.

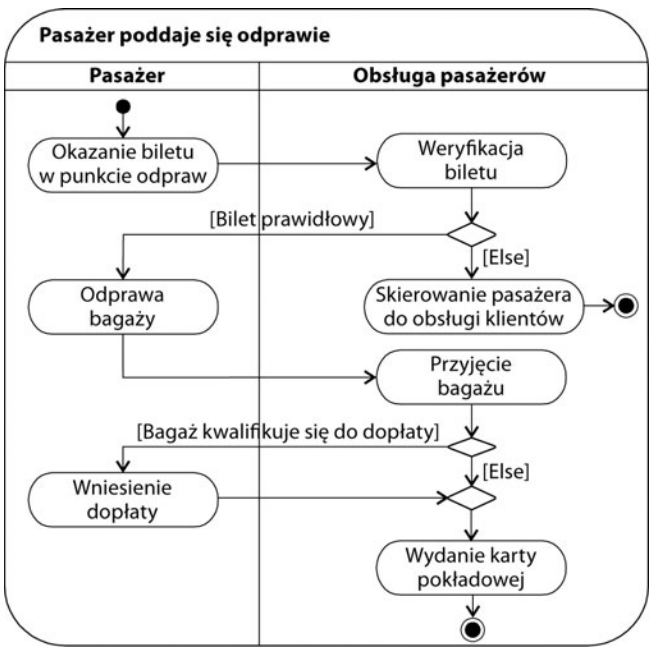
Poziom szczegółowości diagramów aktywności należy dobierać w sposób bardzo świadomy. Należy sprawdzić, jaki jej poziom jest do przyjęcia dla odbiorców diagramu, a z drugiej strony, jaka jest minimalna dopuszczalna szczegółowość. Nie ma możliwości znalezienia złotego środka odpowiedniego we wszystkich możliwych przypadkach; zagadnienie to jest ściśle związane z grupą odbiorców i ze specyfiką modelu.

Po uzupełnieniu szczegółów otrzymamy dodatkowe akcje, co prezentuje rysunek 3.20.

Rysunek 3.20. Akcje wchodzące w skład aktywności na wyższym poziomie abstrakcji

Przejęcie aktorów z biznesowych przypadków użycia — kto jest odpowiedzialny za jakie akcje?

W procesie biznesowym informacja o tym, kto jest odpowiedzialny za każdą akcję i kto ją wykonuje, jest bardzo ważna (rysunek 3.21).



Rysunek 3.21. Diagram aktywności przypadku użycia „odprawa"

W diagramach aktywności perspektywy zewnętrznej wykorzystywani są ci sami aktorzy, co w diagramach przypadków użycia. Każdy aktor jest odpowiedzialny za określone działanie; jest on umieszczany w określonej partycji jako jednostka odpowiedzialna.

Aktywności są przypisywane odpowiedzialnym za nie jednostkom. Podział diagramu aktywności na partycje pozwala dokonać czytelnego przeglądu zakresów odpowiedzialności. Podziału tego można również dokonać na bazie innych kryteriów.

Diagram aktywności może na przykład zostać podzielony w taki sposób, aby wyodrębnić operacje wykonywane ręcznie, automatycznie i w sposób półautomatyczny. Dzięki takiemu podziałowi na partycje uzyskujemy doskonałą podbudowę pod diagram przepływów na potrzeby systemu informatycznego.

Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Podobnie jak diagramy przypadków użycia, diagramy aktywności również muszą być zweryfikowane pod kątem poprawności. Weryfikacji należy dokonać we współpracy z dostawcami wiedzy.

Lista kontrolna 3.4. Weryfikacja diagramów aktywności w perspektywie zewnętrznej

- W trakcie konstruowania diagramów aktywności w perspektywie zewnętrznej należy pamiętać, że procedury wewnętrzne i procesy biznesowe nie mają znaczenia. Rozważania należy ograniczyć do opisu tych funkcji systemu biznesowego, które mają związek z uczestnikami zewnętrznymi.
- Warunki różnych wyjść nie powinny nakładać się na siebie — w przeciwnym wypadku przepływ sterowania staje się niejednoznaczny, co powoduje, że nie wiadomo dokładnie, jaką drogą będzie postępował po przejściu przez węzeł decyzyjny.
- Warunki muszą zawierać wszystkie możliwości — w przeciwnym wypadku przepływ sterowania może ugrzęznąć. W przypadku wątpliwości w węźle kontrolnym należy uwzględnić wyjście z warunkiem **else**.
- Rozgałęzienia i złączenia należy odpowiednio zrównoważyć. Liczba przepływów wychodzących z rozgałęzienia powinna zgadzać się z liczbą przepływów wchodzących do złączenia.

3.3.7. Diagramy sekwencji

UML oferuje dwa typy diagramów reprezentujących interakcje: sekwencji i komunikacji. Obydwa typy służą wizualizacji wymiany informacji. Każdy z nich ma jednak nieco inne przeznaczenie: w **diagramach komunikacji** kładzie się nacisk na powiązania między poszczególnymi obiektami oraz ich topologię, w **diagramach sekwencji** zaś skupia się na chronologii wymiany informacji. W perspektywie zewnętrznej cenniejsze wydają się diagramy sekwencji; sugerujemy więc rezygnację z diagramów komunikacji. Mamy po temu dwa powody:

- Diagramy sekwencji są łatwiejsze do zrozumienia dla ich twórców i czytelników. W konkretnych sytuacjach mieliśmy okazję zaobserwować, że są one chętniej przyjmowane z powodu ich prostoty.
- Unikamy w ten sposób nadmiaru diagramów prezentujących te same treści. W tym przypadku im mniej, tym lepiej!

Jeśli klient lub partner biznesowy wykorzystuje oferowaną usługę, między dostawcą a odbiorcą odbywa się komunikacja. Ten proces można opisać jako serię interakcji. Interakcje można w prosty i czytelny sposób umieścić na diagramie sekwencji. Aktywności stron oraz warunki występowania interakcji w tym diagramie nie mają znaczenia. Można jednak je nanieść na diagram w ramach opisu uzupełniającego.

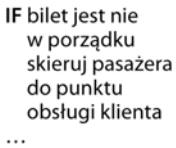
Podobnie jak diagramy aktywności, diagramy sekwencji mogą być modelowane w oparciu o kilka osobnych przypadków użycia (mogą je łączyć je lub występować w charakterze uzupełnienia). Diagram sekwencji ilustruje różne scenariusze biznesowego przypadku użycia.

Diagramy sekwencji mogą być używane jako podstawa przy wymianie komunikatów między systemem biznesowym a jednostkami zewnętrznymi (rysunek 3.22). Więcej informacji na ten temat przekażemy w rozdziale 5.

Rysunek 3.22. Elementy diagramu sekwencji

W diagramie sekwencji mamy do dyspozycji następujące elementy:

Komentarz — używany jest do wprowadzania dodatkowych informacji w diagramie sekwencji (UML zezwala na wprowadzanie komentarzy we wszystkich typach diagramów).

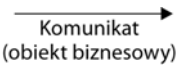


W komentarzach mogą być zawarte na przykład aktywności partnerów lub warunki wystąpienia sekwencji.

Obiekt — obiekty biorące udział w interakcjach są umieszczane na osi *x*. Obiekty są nadawcami i odbiorcami komunikatów w diagramie sekwencji.

W modelu systemu biznesowego (w perspektywie zewnętrznej) obiekty reprezentują aktorów tego systemu i sam system.

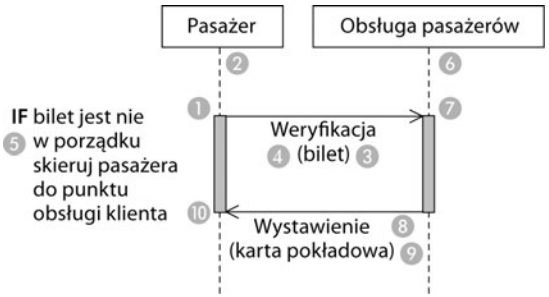
Komunikat i obiekt biznesowy — komunikaty wysyłane przez obiekty są prezentowane na osi *y*. Komunikaty są wprowadzane w kolejności chronologicznej od góry do dołu diagramu. Kierunek strzałki informuje o kierunku przekazu komunikatu.



Obiekt biznesowy jest napisany w nawiasach. Obiekty biznesowe są łączone w jedną całość z komunikatem. Przykładem takiego obiektu może być bilet, karta pokładowa oraz bagaż. Więcej szczegółów na temat tego typu elementów zawiera punkt 3.4.2, „Diagramy pakietów".

Czytanie diagramów sekwencji

Rysunek 3.23 przedstawia diagram sekwencji z obiektami **pasażer** i **obsługa pasażerów**. Cały diagram dokumentuje proces biznesowego przypadku użycia **odprawa pasażera**.



Rysunek 3.23. Diagram sekwencji „odprawa pasażera"

Diagram sekwencji czyta się od góry (1). Punkt początkowy w lewym górnym narożniku diagramu (1) jest umieszczony na linii pionowej reprezentującej obiekt **pasażer** (2) jako nadawcę i odbiorcę komunikatów. Przepływ rozpoczyna się w momencie wręczenia **biletu** (3) pracownikowi **obsługi pasażerów** w celu **weryfikacji** (4). Wywołanie akcji **weryfikacja** (4) stanowi komunikat: wręczany **bilet** (3) jest obiektem biznesowym. Grot strzałki informuje o tym, że nadawcą komunikatu jest **pasażer**, a **odbiorcą** (6) jest obiekt **obsługa pasażerów**.

Otrzymanie komunikatu przez obiekt obsługi pasażerów inicjalizuje aktywności, które są sygnalizowane szarą pionową linią (7). Diagram nie informuje o sposobie obsługi procesu przez obiekt obsługi pasażerów, co oznacza, że nie są na nim odzwierciedlone aktywności.

Jedyna wskazówka dotycząca aktywności może znajdować się w komentarzu (5). Komentarze mogą być wprowadzane na lewym marginesie diagramu sekwencji. Dokładny opis przetwarzania można znaleźć na diagramie aktywności „odprawa pasażera" (zobacz rysunek 3.21).

Na końcowym etapie sekwencji obsługa pasażerów **wydaje** (8) **kartę pokładową** (9). Tutaj interakcje ilustrowane diagramem kończą się dla obydwu stron. Sytuacja ta jest sygnalizowana końcem szerokiej, szarej, pionowej linii (10).

W modelu biznesowym nie interesują nas wszystkie możliwości diagramów sekwencji. UML oferuje o wiele więcej narzędzi przydatnych w diagramach tego typu, lecz z naszego doświadczenia wynika, że zaprezentowany podzbiór wystarczy do uchwycenia kluczowych aspektów.

3.3.8. Konstruowanie diagramów sekwencji

Poniższa lista kontrolna zawiera etapy niezbędne do stworzenia diagramu sekwencji. Poniżej omawiamy każdy z tych etapów.

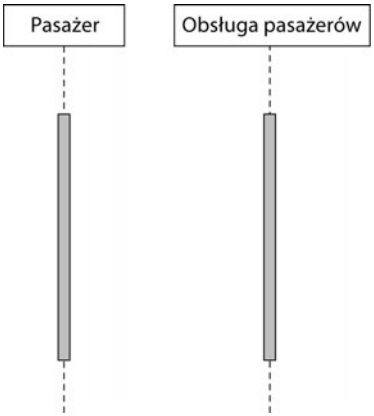
Lista kontrolna 3.5. Konstruowanie diagramów sekwencji w perspektywie zewnętrznej

- Identyfikacja aktorów i systemu biznesowego — kto bierze udział w sekwencji?
- Identyfikacja inicjatorów — kto rozpoczyna sekwencję?
- Opis wymiany komunikatów między aktorami a systemem biznesowym — które komunikaty są wymieniane?
- Identyfikacja przebiegu interakcji — jaka jest ich kolejność?
- Wprowadzanie dodatkowych informacji — co jeszcze jest istotne?
- Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Identyfikacja aktorów i systemu biznesowego — kto bierze udział w sekwencji?

Diagramy sekwencji ilustrują interakcje między aktorami a systemem biznesowym. Strony interakcji odpowiadają aktorom z diagramów przypadków użycia. W zależności od przepływu ilustrowanego na diagramie sekwencji należy dobrać odpowiednich aktorów i systemy biznesowe z diagramów przypadków użycia.

W naszym studium przypadku (rysunek 3.24) znajdujemy następujące strony interakcji: **pasażera** i **obsługę pasażerów**. Wykorzystamy je w diagramie sekwencji (rysunek 3.23).



Rysunek 3.24. Konstruowanie diagramów sekwencji

Identyfikacja inicjatorów — kto rozpoczyna sekwencję?

W każdej sekwencji interakcji należy zidentyfikować aktora, który ją inicjuje. Ten aktor nosi nazwę **inicjatora**. W perspektywie zewnętrznej modelu biznesowego każdy biznesowy przypadek użycia jest inicjowany przez aktora, zatem aktora tego wybiera się spośród wszystkich wykorzystanych w diagramach przypadków użycia.

W naszym diagramie sekwencji ilustrującym odprawę pasażera interakcja jest inicjowana przez pasażera — jest to ilustracja **odprawy** wykonywanej przez obsługę pasażerów.

Opis wymiany komunikatów między aktorami a systemem biznesowym — które komunikaty są wymieniane?

Po zidentyfikowaniu inicjatora należy zidentyfikować kolejne etapy sekwencji interakcji. Na każdym etapie komunikacji należy zidentyfikować przekazywaną informację. W ten sposób definiuje się komunikat. Komunikaty to żądania skierowane do określonego partnera komunikacji. Należy zidentyfikować również obiekty biznesowe przekazywane wspólnie z komunikatami.

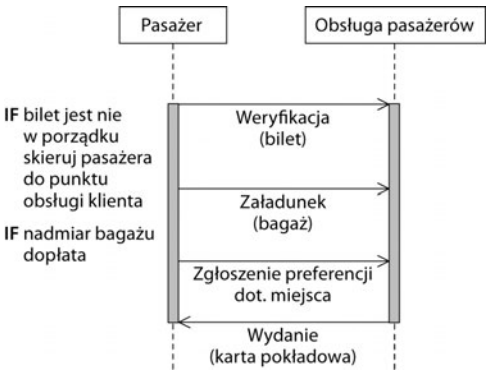
Identyfikacja przebiegu interakcji — jaka jest kolejność?

Wszystkie komunikaty są przekazywane w określone kolejności chronologicznej, którą również należy określić. Są one wypisywane na osi *y* w rosnącej kolejności chronologicznej (od góry do dołu — zobacz rysunek 3.25).

Wprowadzanie dodatkowych informacji — co jeszcze jest istotne?

Ważne aktywności aktorów i systemów biznesowych zaangażowanych w sekwencję można ująć na diagramie w postaci komentarzy. Komentarze umieszcza się na poziomie odpowiedniego komunikatu. Należy ograniczyć się do naprawdę kluczowych informacji, aby diagram nie został przeciążony (rysunek 3.26).

Rysunek 3.25. Konstruowanie diagramów sekwencji



Rysunek 3.26. Konstruowanie diagramów sekwencji

Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Kompletne diagramy sekwencji można zweryfikować za pomocą poniższej listy kontrolnej.

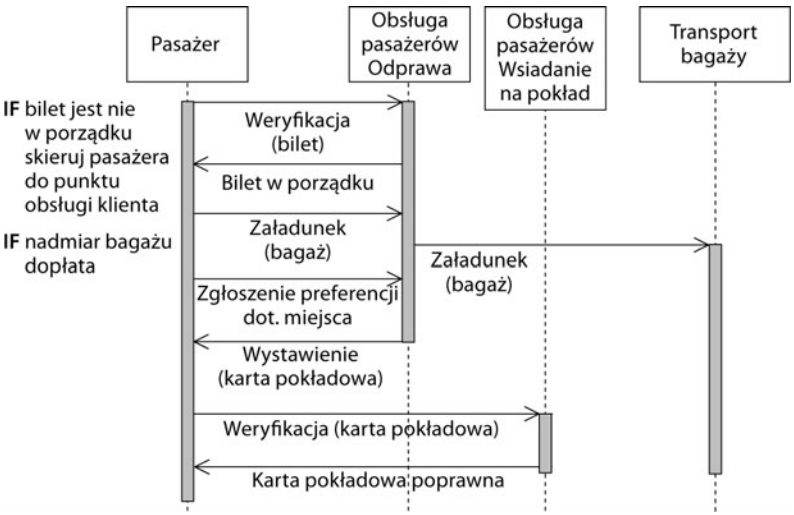
Lista kontrolna 3.6. Weryfikacja diagramu sekwencji w perspektywie zewnętrznej

- Czy wszystkie wymagane diagramy sekwencji są kompletne i dostępne? Każdemu biznesowemu przypadkowi użycia powinien odpowiadać jeden diagram.
- Czy diagramy sekwencji są poprawne? Każdy diagram sekwencji zawiera tylko jeden obiekt reprezentujący system biznesowy i co najwyżej taką liczbę obiektów, ilu aktorów jest zaangażowanych w danym biznesowym przypadku użycia.
- Czy każdy aktor jest wymieniony na diagramie przypadku użycia opisanym przynajmniej w jednym diagramie sekwencji?
- Czy na początku każdego z diagramów sekwencji jest wymieniony aktor inicjalizujący dany biznesowy przypadek użycia?
- Czy wszystkie ważne komentarze są umieszczone na diagramie? Czy nie można zmniejszyć ich liczby i dzięki temu zwiększyć czytelności diagramów?

3.3.9. Diagramy sekwencji o wysokim poziomie abstrakcji

Do zobrazowania procesów biznesowych w bardziej ogólnym ujęciu można użyć diagramów o wysokim poziomie abstrakcji, obejmujących kilka biznesowych przypadków użycia. Diagramy tego typu pozwalają lepiej zrozumieć interakcje między użytkownikami, partnerami i systemem biznesowym. Służą jako podstawa elektronicznej wymiany danych między systemem biznesowym, partnerami biznesowymi a dostawcami (zobacz rozdział 5.).

Rysunek 3.27 ilustruje obsługę pasażerów. Cały proces obejmuje dwa biznesowe przypadki użycia: **odprawę** i **wsiadanie na pokład**.

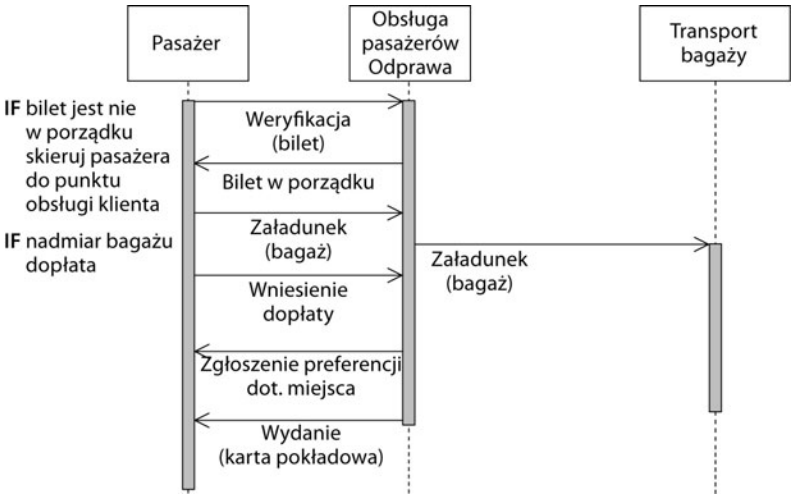


Rysunek 3.27. Diagram sekwencji „obsługa pasażerów"

3.3.10. Diagramy sekwencji do tworzenia scenariuszy zastosowania przypadków użycia

Diagramy sekwencji pomagają w uszczegóławianiu i definiowaniu biznesowych przypadków użycia poprzez położenie nacisku na wymianę komunikatów. Na diagramie sekwencji można przedstawić różne scenariusze biznesowych przypadków użycia. Reprezentacja jest ograniczona do wymiany komunikatów w ramach każdego takiego przypadku użycia. Poziom szczegółowości diagramu sekwencji o wysokim poziomie abstrakcji jest wyższy niż w przypadku diagramu sekwencji opisującego jeden biznesowy przypadek użycia.

Rysunek 3.28 przedstawia diagram sekwencji dla biznesowego przypadku użycia o nazwie **odprawa**. Ten diagram przedstawia scenariusz działań związanych z **odprawą pasażera** z punktu widzenia komunikacji między partnerami biorącymi udział w sekwencji; nie pojawia się tu



Rysunek 3.28. Diagram sekwencji biznesowego przypadku użycia „odprawa pasażera"

natomiast przedstawiciel pasażera. Diagramy sekwencji — podobnie jak diagramy aktywności — pokazują, czy poszczególne etapy biznesowych przypadków użycia są realizowane przez aktorów wspólnie, czy też niezależnie (w takim wypadku tego typu interakcje oczywiście nie pojawią się na diagramie sekwencji).

3.4. Perspektywa wewnętrzna

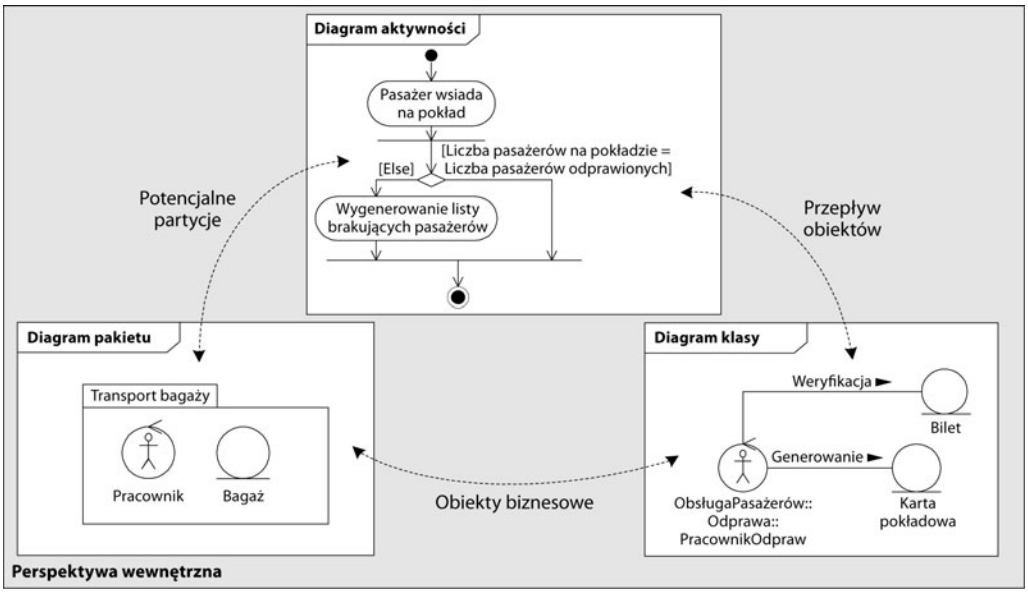
Perspektywa wewnętrzna opisuje wewnętrzny **proces** i **aktywności**, **powiązania** i **struktury** systemu biznesowego. Systemy informatyczne i osoby zaangażowane są odpowiedzialne za dostarczanie towarów i usług oferowanych przez system biznesowy. W tym momencie mamy już opisane środowisko systemu biznesowego, wchodzimy więc do czarnej skrzynki, jaką do tej pory był dla nas system. Od tej pory ma już znaczenie, czy proces w nim wykonywany jest realizowany ręcznie, czy wspomagany narzędziami informatycznymi; czy pracownik organizacji musi wypełniać jeden, dwa, czy też dwadzieścia formularzy; czy do realizacji zadań niezbędne są dostawy z zewnątrz.

3.4.1. Elementy perspektywy

Perspektywa zewnętrzna systemu biznesowego jest opisywana za pomocą następujących diagramów:

- **Diagramy pakietów** — opisują jednostki organizacyjne w postaci pakietów.
- **Diagramy klas** — opisują powiązania i związki między współpracownikami i obiektami biznesowymi.

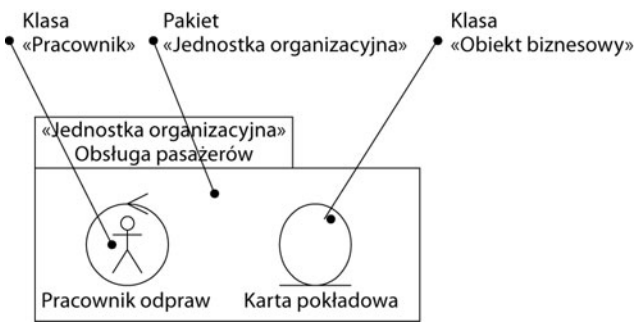
- **Diagramy aktywności** — opisują procesy biznesowe w ramach systemu biznesowego. Podmiotami są tu towary i usługi oferowane przez system biznesowy (rysunek 3.29).



Rysunek 3.29. Typy diagramów w wewnętrznej perspektywie systemu biznesowego

3.4.2. Diagramy pakietów

W wewnętrznej perspektywie systemu biznesowego bardzo duże znaczenie odgrywa struktura jednostki organizacyjnej. W UML-u jednostki organizacyjne są opisywane w postaci pakietów, które mogą zawierać pracowników, obiekty biznesowe i inne jednostki organizacyjne. W naszym studium przypadku opiszemy jednostkę organizacyjną obsługa pasażerów (rysunek 3.30).



Rysunek 3.30. Diagram pakietów

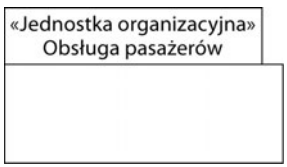
Jednostki organizacyjne mogą być odpowiedzialne za wykonywanie aktywności wchodzących w skład procesów biznesowych. Jednostkami organizacyjnymi są abstrakcje określonych stanowisk pracy w ramach organizacji.

W UML-u jednostki takie obejmują pracowników, obiekty biznesowe, inne jednostki organizacyjne oraz powiązania między nimi. Podstawowa zasada jest następująca: jednostki te są w całości częścią modelowanego systemu biznesowego, zaś odpowiadające im funkcjonalnie struktury spoza tego systemu są aktorami.

W diagramach pakietów mamy do czynienia z następującymi elementami:

Pakiet «Jednostka organizacyjna»

Jednostki organizacyjne są przedstawiane w postaci pakietów. W mniejszym prostokącie u góry diagramu, poniżej zapisu **«Jednostka organizacyjna»**, wpisuje się jej nazwę.



Strukturę jednostki organizacyjnej wpisuje się w głównym prostokącie diagramu. W większości przypadków wystarczy wymienić najważniejsze elementy (pracowników, obiekty biznesowe itp.).

Klasa «Pracownik»

Element klasy **«Pracownik»** jest wykorzystywany do opisywania ról osób zaangażowanych w wykonanie procesów biznesowych.

Nie interesuje nas status pracowników, czyli to, czy są oni zatrudnieni na stałe, najemni lub pracujący ochotniczo. Interesują nas natomiast realizowane przez nich role, czyli zadania. Pracownicy są odpowiedzialni za dostarczanie towarów i usług, są częścią systemu biznesowego. Oto najważniejsze cechy pracowników:

- są to osoby;
- są oni częścią systemu biznesowego;
- mogą się komunikować z innymi pracownikami oraz z aktorami spoza systemu biznesowego.

Pracowników opisuje się symbolem, zaś poniżej niego wpisywana jest rola pracownika. Symbol ma postać aktora otoczonego kółkiem, które ma symbolizować fakt umieszczenia tego aktora **wewnątrz** (systemu). Symbol pracownika można również pominąć. Wówczas stosuje się symbol klasy, a dla zasygnalizowania faktu, że mamy do czynienia z pracownikiem, dodaje się etykietę z napisem **«Pracownik»**.

«Obiekt biznesowy»

Obiekty biznesowe są **pasywne**, co oznacza, że nie inicjalizują interakcji. Obiekty biznesowe mogą być zaangażowane w różne biznesowe przypadki użycia i w więcej niż jedną interakcję. Dzięki temu mogą służyć jako element łączący między biznesowymi przypadkami użycia lub pracownikami zaangażowanymi w różne biznesowe przypadki użycia.

Pracownicy obsługują (wykorzystują, kontrolują, produkują) obiekty biznesowe. W naszym przypadku użycia obiektem jest na przykład bilet, sztuka bagażu czy karta pokładowa. Obiekty biznesowe są również reprezentowane za pomocą symboli — nazwę obiektu wpisuje się poniżej jego symbolu.

Również w przypadku obiektu biznesowego można pominąć symbol. Wówczas powyżej nazwy wpisuje się etykietę **«Obiekt biznesowy»**.

Czytanie diagramów pakietów

Przykładowy diagram pakietu znajduje się na rysunku 3.31. Poniżej etykiety **«Jednostka organizacyjna»** (1) znajduje się pakiet (2) reprezentujący jednostkę organizacyjną. Nazwą tej jednostki jest **obsługa pasażerów** (3). Znajdziemy tu **pracownika odpraw** (4) oraz obiekt biznesowy **karta pokładowa** (5). Symbol (4) po lewej stronie reprezentuje **pracownika**, etykieta (6) poniżej symbolu graficznego opisuje **rolę pracownika** w organizacji. Symbol (5) po prawej reprezentuje **obiekt biznesowy**, etykieta (7) poniżej symbolu wskazuje **typ** tego obiektu biznesowego.

Rysunek 3.31. Diagram pakietu

Na diagramie znajduje się tylko **jeden** symbol pracownika odpraw. Nie oznacza to jednak, że w organizacji jest tylko jeden pracownik zatrudniony na takim stanowisku. Symbol pracownika opisuje rolę spełnianą przez dowolną liczbę pracowników departamentu odpraw. Oczywiście, istnieją też inne role pracowników w ramach obsługi pasażerów (menedżer, asystent itp.). Nie mają one jednak znaczenia w odwzorowaniu naszych procesów i dlatego nie należy umieszczać ich na diagramie pakietu.

3.4.3. Konstruowanie diagramów pakietów

Poniższa lista kontrolna zawiera etapy niezbędne do skonstruowania diagramów pakietów. Każdy z etapów opisujemy poniżej.

Lista kontrolna 3.7. Konstruowanie diagramów pakietów w perspektywie wewnętrznej

- Utworzenie szkicu diagramu pakietu w systemie biznesowym — którzy pracownicy i które obiekty biznesowe tworzą system biznesowy?
- Identyfikacja dodatkowych jednostek organizacyjnych — kto jeszcze bierze udział w działaniu systemu?
- Przypisanie pracowników i obiektów biznesowych odpowiednim jednostkom organizacyjnym — kto należy do jakiej komórki?
- Identyfikacja dodatkowych jednostek organizacyjnych, pracowników i obiektów biznesowych — kto jeszcze bierze udział w działaniu systemu?
- Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Utworzenie szkicu diagramu pakietów w systemie biznesowym — którzy pracownicy i które obiekty biznesowe tworzą system biznesowy?

Pierwszą analizowaną jednostką organizacyjną jest cały modelowany system biznesowy. W naszym przypadku będzie to system obsługi pasażerów (rysunek 3.32). W pierwszym kroku wyszukujemy odpowiednie role pracowników (stanowiska) oraz obiekty biznesowe. Pomocne mogą okazać się istniejące opisy stanowisk i diagramy organizacyjne.

Identyfikacja dodatkowych jednostek organizacyjnych — kto jeszcze bierze udział w działaniu systemu?

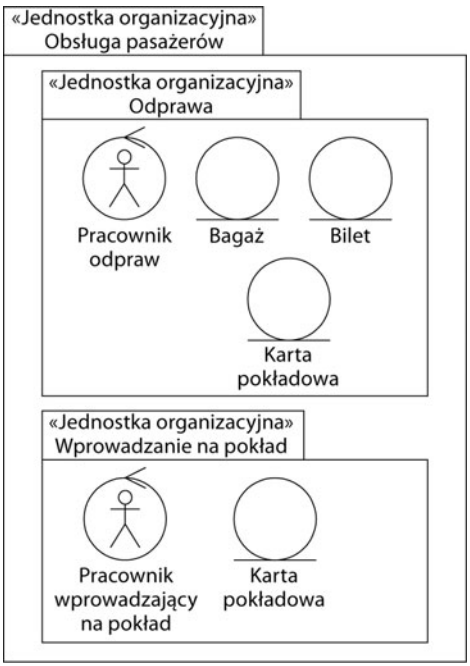
Niektóre jednostki organizacyjne można podzielić na inne jednostki organizacyjne (departamenty, zespoły, grupy). Do tego celu można posłużyć się diagramami organizacyjnymi i opisami stanowisk, z których wybierzemy jednostki odpowiednie dla modelu. Odpowiednie jednostki organizacyjne i stanowiska to takie, które są bezpośrednio zintegrowane z przetwarzaniem towarów i usług.

W naszym studium przypadku obsługę pasażerów dzielimy na składowe jednostki organizacyjne takie, jak odprawa i załadunek. Dalszy podział jest możliwy, ale ma sens jedynie wówczas, gdy pozwoli na lepsze odzwierciedlenie procesów biznesowych. Na przykład zadania realizowane przez sekretarkę nie mają kluczowego znaczenia w rozważanych procesach biznesowych.

Rysunek 3.32. Konstruowanie diagramu pakietowego

Przypisanie pracowników i obiektów biznesowych odpowiednim jednostkom organizacyjnym — kto należy do jakiej komórki?

Do utworzonych jednostek organizacyjnych muszą być przydzieleni pracownicy i obiekty biznesowe. Jak widać na rysunku 3.33, poszczególne obiekty biznesowe rozdzieliliśmy pomiędzy jednostki organizacyjne. Dzięki temu struktura i przyporządkowania są czytelne i zrozumiałe.



Rysunek 3.33. Jednostka organizacyjna „obsługa pasażerów"

Identyfikacja dodatkowych jednostek organizacyjnych, pracowników i obiektów biznesowych — kto jeszcze bierze udział w działaniu systemu?

Diagramy pakietów w UML-u reprezentujące jednostki organizacyjne nie powinny być mylone z diagramami organizacyjnymi. W rzeczywistości diagramy organizacyjne mają związek z diagramami pakietów, jednak te ostatnie, oprócz pracowników, uwzględniają także obiekty biznesowe. Z diagramów organizacyjnych możemy odczytać strukturę i role poszczególnych pracowników, by użyć ich jako podstawy do tworzenia diagramów pakietów.

Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Kompletny diagram pakietów można sprawdzić za pomocą poniższej listy kontrolnej.

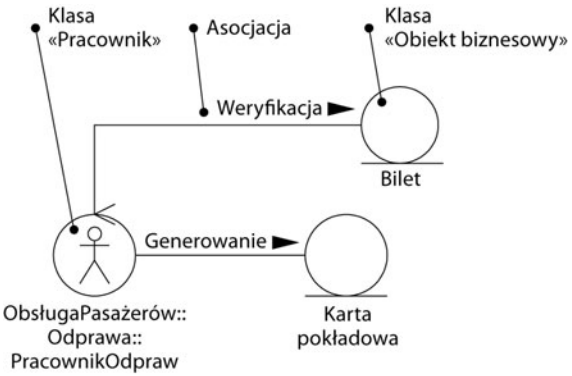
Lista kontrolna 3.8. Weryfikacja diagramów pakietów w perspektywie wewnętrznej

- Czy w diagramie pakietów zostali ujęci wszyscy pracownicy i jednostki organizacyjne zaangażowane w obsługę towarów i usług oferowanych przez system?
- Czy w diagramie pakietów nie znaleźli się pracownicy i jednostki organizacyjne niezwiązane z obsługą towarów i usług oferowanych przez system?
- Czy w diagramie pakietów znalazły się wszystkie obiekty biznesowe niezbędne do obsługi towarów i usług oferowanych przez system?
- Czy w diagramie pakietów nie znalazły się obiekty biznesowe niezwiązane z obsługą towarów i usług oferowanych przez system?

3.4.4. Diagramy klas

Diagramu klasy można użyć do zilustrowania strukturalnych elementów systemu biznesowego, to znaczy związków między pracownikami, obiektami biznesowymi i jednostkami zewnętrznymi. Diagramy klas można znacznie uprościć na poziomie modelu biznesowego i wykorzystać tylko wybrane elementy. W tym przypadku panuje zasada „im mniej, tym lepiej". Gdyby użyć niezliczonych opcji diagramów klas, modele stałyby się bardzo mało czytelne. Na poziomie systemu biznesowego musimy działać z takim założeniem, że nie wszyscy zaangażowani w tworzenie systemu posiadają wiedzę informatyczną i znają terminologię diagramów klas. Zaleta UML-a, czyli możliwość wykorzystania go do komunikacji między stronami o niejednolitym poziomie zaawansowania technicznego, w takim wypadku przestałaby działać. Szerszy opis diagramów klas można znaleźć w rozdziale 4.

W diagramach klas (rysunek 3.34) będziemy pracować z zaledwie kilkoma elementami:



Rysunek 3.34. Diagram klasy

Klasa «Pracownik»

Tę klasę opisaliśmy już w punkcie 3.4.2, „Diagramy pakietów". Jest ona dokładnie taka sama, jak w diagramach pakietów — i tak samo jak w tamtym przypadku może być oznaczona ikonką pracownika lub symbolem klasy.

Jak widać na rysunku 3.34, w nazwie klasy można wpisać całą ścieżkę, co będzie sygnalizować przynależność do pakietu. W naszym przykładzie ścieżka informuje, że **pracownik odpraw** należy do pakietu **odprawa**, a pakiet **odprawa** należy do pakietu **obsługa pasażerów**. Poszczególne elementy ścieżki są oddzielane podwójnym dwukropkiem. Klasa **pracownik** jest użyta w diagramie klasy do zilustrowania związków z innymi pracownikami, aktorami i obiektami biznesowymi.

Klasa «Obiekt biznesowy»

Klasę **obiekt biznesowy** omówiliśmy już w punkcie 3.4.2, „Diagramy pakietów". W diagramach klas stosuje się dokładnie te same klasy.

Podobnie jak w diagramie klasy, obiekty biznesowe można oznaczyć ikonką lub symbolem klasy.

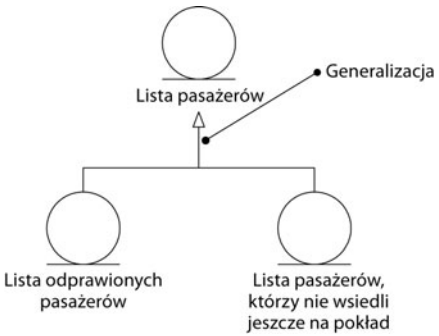
Powiązanie

Powiązanie reprezentuje związek między obiektami o dokładnie określonym znaczeniu. Można je opisać etykietą zawierającą jego nazwę. Aby wskazać powiązanie, którego dotyczy etykieta, można obok niej umieścić trójkąt wskazujący odpowiedni kierunek.

Oprócz wymienionych wyżej elementów warto wspomnieć także o generalizacji. Nie uważamy jednak, żeby stosowanie tego elementu było obowiązkowe.

Generalizacja

Generalizacja stanowi określone powiązanie między uogólnieniem a określonym elementem. Generalizacja i specjalizacja pozwalają tworzyć struktury hierarchiczne. Jeśli istnieje kilka obiektów biznesowych, które powinny być połączone w jeden element, generalizacja nadaje się do tego znakomicie (rysunek 3.35).

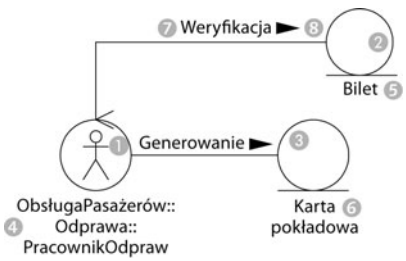


Rysunek 3.35. Diagram klas zawierający generalizację

W przypadku pracowników lepiej jednak stosować strukturalizację w ramach diagramu pakietu.

Czytanie diagramów klas

Rysunek 3.36 zawiera przykład diagramu klasy z naszego studium przypadku. Mamy tu klasy **pracownik odpraw** (1), **bilet** (2) i **karta pokładowa** (3) wraz ze scenariuszami ich zastosowań.



Rysunek 3.36. Diagram klas

Z etykiety (4) symbolu pracownika (1) wynika, że pracownik odpraw należy do jednostki organizacyjnej „odprawa", która z kolei należy do „obsługi pasażerów".

Etykiety zapisane przed etykietą pracownika, oddzielone podwójnymi dwukropkami, informują o jednostkach organizacyjnych, do których należy dany pracownik. Jak można zauważyć, obsługa pasażerów i odprawa są jednostkami organizacyjnymi zdefiniowanymi w ramach diagramu pakietu.

Na przykładowym diagramie mamy do czynienia z dwoma obiektami biznesowymi: **bilet** (5) i **karta pokładowa** (6).

Asocjację między klasami należy odczytywać w następujący sposób:

- pracownik odpraw (4) weryfikuje (7) bilet (5).

Mały trójkąt (8) obok nazwy asocjacji (7) wskazuje **kierunek** odczytu nazwy tej asocjacji. Wszystkie asocjacje w ramach diagramu klas można odczytywać właśnie w taki sposób.

W diagramach klas modelu systemu biznesowego nie stosuje się powtarzania symboli, a więc z diagramów nie można odczytać liczby obiektów lub klas biorących udział w asocjacji.

Na tym etapie nie ma znaczenia, czy pracownik odpraw generuje jedną, czy większą liczbę kart pokładowych. Ważne informacje na temat liczby obiektów można umieścić w ramach komentarzy. Do liczby elementów powrócimy na dalszych etapach prac nad modelem, a dokładniej w modelu systemu informatycznego, który omówimy w rozdziale 4., noszącym tytuł „Modelowanie systemów informatycznych".

3.4.5. Konstruowanie diagramów klas

Poniżej opisaliśmy etapy tworzenia diagramów klas.

Lista kontrolna 3.9. Konstruowanie diagramów klas w perspektywie wewnętrznej

- Identyfikacja klas — które z nich należy ująć w diagramie klas?
- Utworzenie asocjacji między klasami — które z nich są wzajemnie powiązane?
- Określenie znaczenia asocjacji — co oznaczają poszczególne powiązania?
- Zdefiniowanie generalizacji — czy obiekty biznesowe można pogrupować?
- Weryfikacja perspektywy — czy wszystko zostało wykonane prawidłowo?

Identyfikacja klas — które z nich należy ująć w diagramie klas?

Podczas konstruowania diagramu klas można użyć klas zdefiniowanych w diagramach pakietów w perspektywie wewnętrznej, czyli takich, jak pracownicy i obiekty biznesowe. Aktorzy z diagramów przypadków użycia również są klasami, które warto rozważyć pod kątem wykorzystania w diagramie klas. W naszym przykładzie wyróżniliśmy klasy przedstawione na rysunku 3.37.

Rysunek 3.37. Klasy wewnętrznej perspektywy systemu biznesowego

Utworzenie asocjacji między klasami — które z nich są wzajemnie powiązane?

W diagramach klas powiązania między klasami oraz reguły biznesowe modeluje się w formie asocjacji.

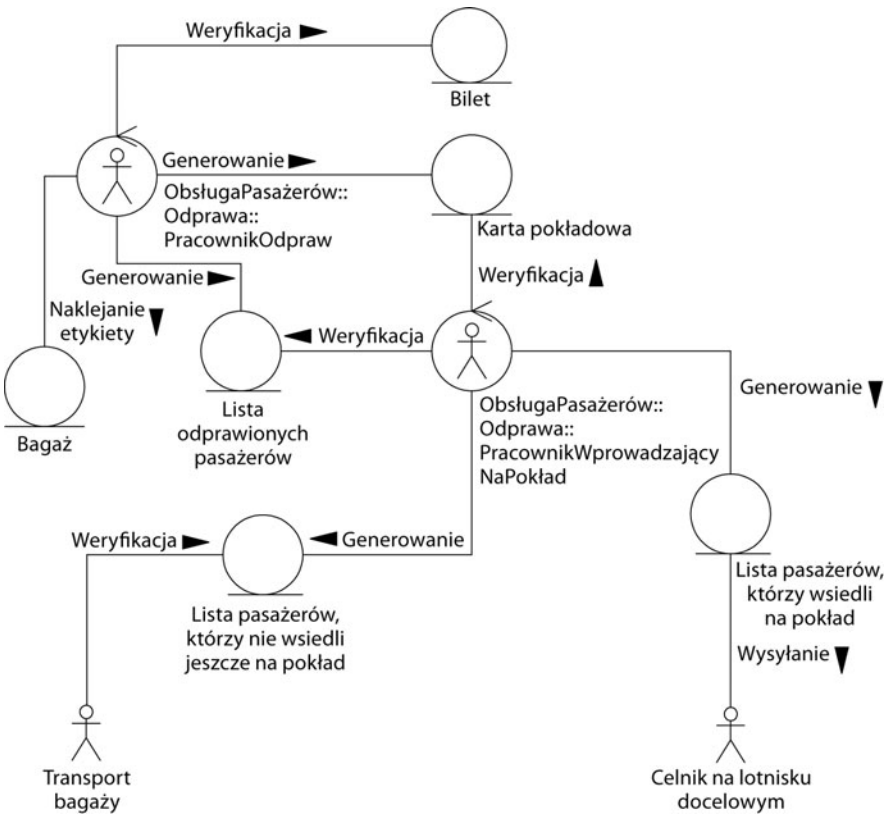
Przy modelowaniu asocjacji należy odpowiedzieć sobie na takie pytanie:

- Jakie powiązania występują między pracownikami, obiektami biznesowymi i innymi obiektami?

Mimo że na początku pracy z diagramem klas wykorzystujemy klasy zidentyfikowane wcześniej, z reguły w trakcie dyskusji na tym etapie prac identyfikowane są nowe klasy.

Określenie znaczenia asocjacji — co oznaczają poszczególne powiązania?

Asocjacje między poszczególnymi klasami muszą posiadać czytelne etykiety, aby diagram klas był zrozumiały i czytelny, możliwy do interpretacji w sposób intuicyjny. Na asocjacji z reguły umieszcza się strzałkę informującą o kierunku jej odczytu (zobacz rysunek 3.38).

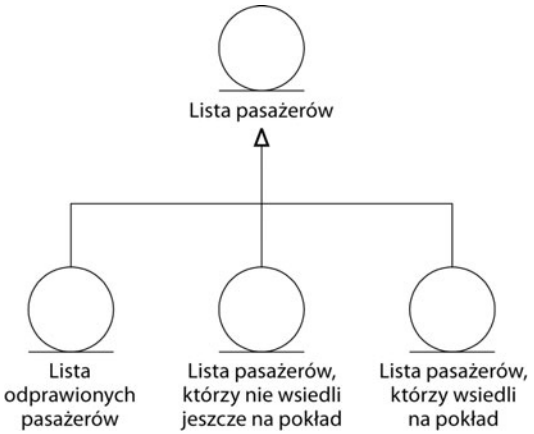


Rysunek 3.38. Diagram klas obsługi pasażerów

Zdefiniowanie generalizacji — czy obiekty biznesowe można pogrupować?

Obiekty biznesowe często warto pogrupować w obiekty o wyższym poziomie abstrakcji. W naszym studium przypadku warto na przykład zaznaczyć, że lista odprawionych pasażerów, lista pasażerów na pokładzie oraz lista pasażerów, którzy jeszcze nie znaleźli się na pokładzie, należą do klasy **«Lista pasażerów»** (rysunek 3.39).

Taka generalizacja informuje odbiorcę diagramu o tym, że wszystkie wymienione listy mają taką samą strukturę (zobacz również punkt 4.2.2, „Generalizacja, specjalizacja i dziedziczenie" w rozdziale 4.):



Rysunek 3.39. Generalizacja w diagramie klas

Weryfikacja perspektywy — czy wszystko zostało wykonane prawidłowo?

Gotowy diagram klasy można zweryfikować za pomocą poniższej listy kontrolnej.

Lista kontrolna 3.10. Weryfikacja diagramu klasy w perspektywie wewnętrznej

- Czy diagram klas jest kompletny? Czy wszystkie klasy występujące na diagramach pakietów są uwzględnione na diagramie klas?
- Czy wszystkie asocjacje są opisane w czytelny sposób? Czy zwroty strzałek są skierowane tak, jak powinny?
- Czy diagram klas jest prawidłowy? Intensywna analiza we współpracy z dostawcami wiedzy z reguły ujawnia większość popełnionych błędów.
- Czy poziom szczegółowości jest optymalny? Czy diagram zawiera wystarczająco dużo szczegółów, aby zobrazować niezbędne informacje, i jednocześnie jest wystarczająco oszczędny, aby nie zaciemniać obrazu?

3.4.6. Diagramy aktywności

Diagramy aktywności pozwalają zobrazować wewnętrzne procesy systemu biznesowego. W przeciwieństwie do tych skoncentrowanych na perspektywie zewnętrznej, diagramy aktywności w perspektywie wewnętrznej nie skupiają się już na związkach z aktorami.

Diagramy aktywności w perspektywie wewnętrznej stanowią ponadto doskonałą podbudowę pod tworzenie instrukcji.

Czytanie diagramów aktywności

Instrukcja czytania diagramów aktywności z punktu 3.3.5, „Diagramy aktywności" może być zastosowana również w stosunku do takich diagramów konstruowanych w perspektywie wewnętrznej.

3.4.7. Konstruowanie diagramów aktywności

Proces konstruowania diagramów aktywności w perspektywie wewnętrznej jest bardzo zbliżony do analogicznego procesu przeprowadzanego w perspektywie zewnętrznej.

Poniższa lista kontrolna została nieco zmodyfikowana na potrzeby perspektywy wewnętrznej.

Lista kontrolna 3.11. Konstruowanie diagramu aktywności w perspektywie wewnętrznej

- Gromadzenie źródeł informacji — skąd mam to wiedzieć?
- Wyszukiwanie aktywności i akcji — co należy zrobić, gdy aktorzy zainicjują dostarczenie towarów lub usług?
- Przejęcie aktorów z biznesowych przypadków użycia — kto jest odpowiedzialny za jakie akcje?
- Połączenie akcji — w jakiej kolejności są one przetwarzane?
- Udoskonalanie aktywności — czy powinny zostać dodane inne diagramy aktywności?
- Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Gromadzenie źródeł informacji — skąd mam to wiedzieć?

Na etapie konstruowania diagramu aktywności zastosowanie mają reguły opisane w punkcie 3.3.4, „Konstruowanie diagramów przypadków użycia".

Wyszukiwanie aktywności i akcji — co należy zrobić, gdy aktorzy zainicjują dostarczenie towarów lub usług?

W tym przypadku możemy dokonać zapożyczenia z diagramów aktywności w perspektywie zewnętrznej. W przypadku każdego biznesowego procesu opisanego w perspektywie zewnętrznej musimy odpowiedzieć na pytanie: „W jaki sposób odbywają się procesy wewnętrzne i jaką postać mają wewnętrzne procesy biznesowe?". W poszukiwaniu aktywności i akcji pomocne będą odpowiedzi na następujące pytania:

- Jakie działania muszą być podjęte przez pracowników systemu biznesowego, aby mógł on dostarczyć towary i usługi?
- Co robi każdy z pracowników?
- Jakie zdarzenia zewnętrzne inicjują każdą z aktywności i akcji?

Często można znaleźć istniejącą dokumentację przepływów, nieformalną lub ustrukturalizowaną, w której łatwo zidentyfikować aktywności.

Przejęcie aktorów z biznesowych przypadków użycia — kto jest odpowiedzialny za jakie akcje?

Za poszczególne akcje odpowiedzialni są pracownicy i jednostki organizacyjne z diagramu pakietów. Aktorzy z diagramów przypadków użycia również są tu potrzebni — pod warunkiem, że są zaangażowani w modelowane procesy biznesowe.

Każdy pracownik, każda jednostka organizacyjna i każdy aktor odpowiedzialni za określone aktywności muszą być umieszczeni w odpowiedniej partycji diagramu aktywności. Do ich odpowiedzialności przypisywane są poszczególne akcje.

Na etapie udoskonalania diagramu aktywności można dodawać dodatkowe obszary odpowiedzialności, na przykład indywidualne stanowiska lub zespoły.

Połączenie akcji — w jakiej kolejności są one przetwarzane?

Diagram aktywności powstaje przez łączenie poszczególnych akcji w przepływy, co stanowi opis wewnętrznego procesu biznesowego. W tworzeniu przepływu sterowania pomocne będą odpowiedzi na następujące pytania:

- W jakiej kolejności przetwarzane są akcje?
- Jakie warunki muszą być spełnione, aby akcja mogła być wykonana?
- W których miejscach konieczne są rozgałęzienia?
- Jakie akcje odbywają się równolegle?
- Czy zanim przepływ przejdzie do kolejnych akcji, musi koniecznie nastąpić zakończenie poprzedniej akcji?

Udoskonalanie aktywności — czy powinny zostać dodane inne diagramy aktywności?

Indywidualne akcje można podzielić na bardziej szczegółowe i udoskonalić je za pomocą innych diagramów aktywności. W różnych diagramach aktywności można również opisać różne scenariusze działań w procesie biznesowym.

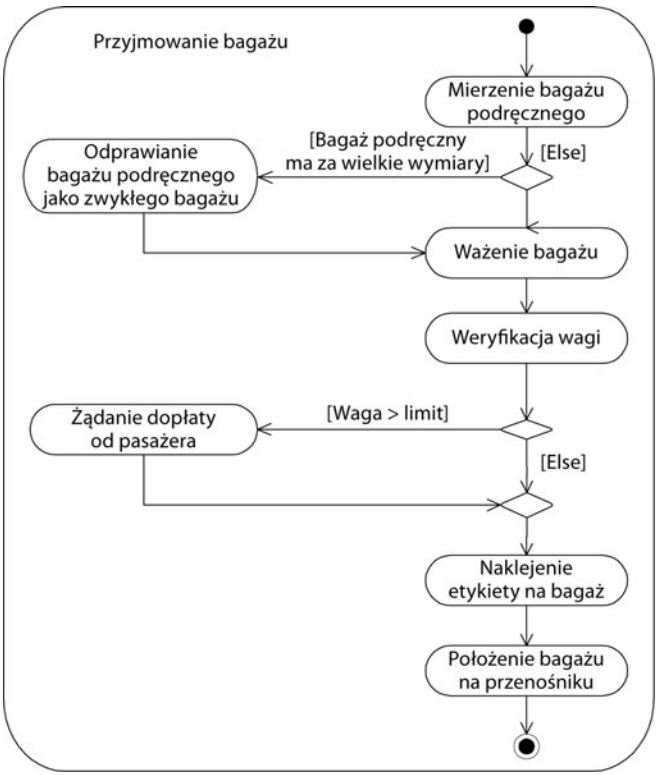
Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Diagramy aktywności w perspektywie wewnętrznej można weryfikować pod kątem poprawności i zawartości. Weryfikacja musi odbywać się przy udziale dostawców wiedzy.

Lista kontrolna 3.12. Weryfikacja diagramów aktywności w perspektywie wewnętrznej

- W trakcie konstruowania diagramów aktywności w perspektywie wewnętrznej należy pamiętać, że znaczenie mają tylko wewnętrzne procedury i procesy biznesowe.
- Warunki różnych wyjść nie powinny nakładać się na siebie; w przeciwnym wypadku przepływ sterowania staje się niejednoznaczny, co powoduje, że nie wiadomo dokładnie, jaką drogą będzie postępował po przejściu przez węzeł decyzyjny.
- Warunki muszą zawierać wszystkie możliwości; w przeciwnym wypadku przepływ sterowania może ugrzęznąć. W przypadku wątpliwości w węźle kontrolnym należy uwzględnić wyjście z warunkiem **else**.
- Rozgałęzienia i złączenia należy odpowiednio zrównoważyć. Liczba przepływów wychodzących z rozgałęzienia powinna zgadzać się z liczbą przepływów wchodzących do złączenia.

Rysunek 3.40 przedstawia diagram aktywności reprezentujący wewnętrzne procesy związane z aktywnościami przyjmowania bagaży podczas odprawy przez obsługę pasażerów.



Rysunek 3.40. Diagram aktywności „przyjęcie bagażu" w perspektywie wewnętrznej

Aktywność **przyjęcie bagażu** przedstawiona na rysunku 3.40 jest realizowana przez obsługę pasażerów. Dla pasażera nie ma znaczenia, w jaki sposób odbywa się transport bagażu. Pasażer jest zainteresowany wyłącznie tym, czy bagaż podręczny nie ma zbyt dużych wymiarów i czy nie musi płacić za nadbagaż. Pracownicy komórki transportu bagaży muszą nalepić etykiety na każdej sztuce bagażu. Wszystkie pozostałe szczegóły przedstawione na diagramie stanowią procesy wewnętrzne obsługi pasażerów i z tego powodu są oznaczone jako „perspektywa wewnętrzna".

ROZDZIAŁ 4

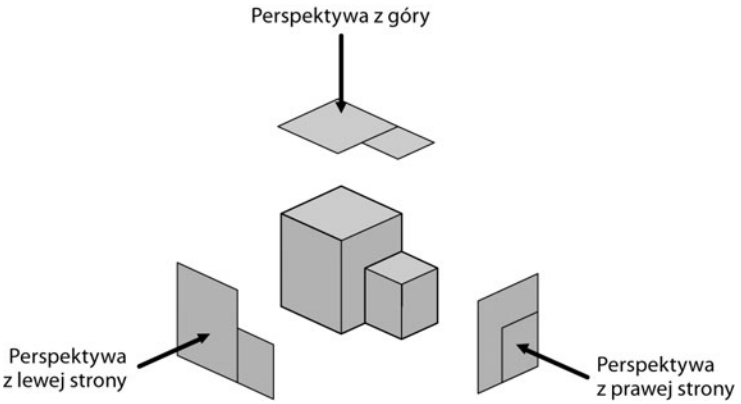
Modelowanie systemów informatycznych

Modelowanie jest fundamentem skutecznej implementacji nowych systemów informatycznych. Prawidłowy i kompletny model zapewnia, że na końcu pracy użytkownicy otrzymają dokładnie taki system informatyczny, jaki jest im potrzebny.

W tym rozdziale zaprezentujemy sposób tworzenia **koncepcyjnego modelu** systemu informatycznego za pomocą UML-a. Biorąc pod uwagę regułę 80:20, nie będziemy wykorzystywać wszystkich funkcji UML-a. Praktyka pokazuje, że nie ma możliwości modelowania za pomocą tego języka wszystkich aspektów systemu z wykorzystaniem wszystkich oferowanych opcji. Dzieje się tak, ponieważ w fazie implementacji zawsze pojawiają się nowe spostrzeżenia, których nie sposób przewidzieć na etapie koncepcyjnym. Ponadto modele powinny być tworzone z minimalnym nakładem kosztów.

Model systemu informatycznego składa się z czterech różnych perspektyw. Każda z nich kładzie nacisk na inny aspekt, ale są one ściśle wzajemnie powiązane. Taki sposób modelowania w różnych perspektywach przedstawia rysunek 4.1. Poszczególne perspektywy wykorzystywane w modelu systemu informatycznego w postaci diagramów UML przedstawia rysunek 4.2.

- Perspektywa zewnętrzna — diagramy przypadków użycia oraz diagramy sekwencji przypadków użycia.
- Perspektywa strukturalna — diagramy klas.
- Perspektywa interakcji — diagramy sekwencji oraz diagramy komunikacji.
- Perspektywa zachowań — diagramy stanów.



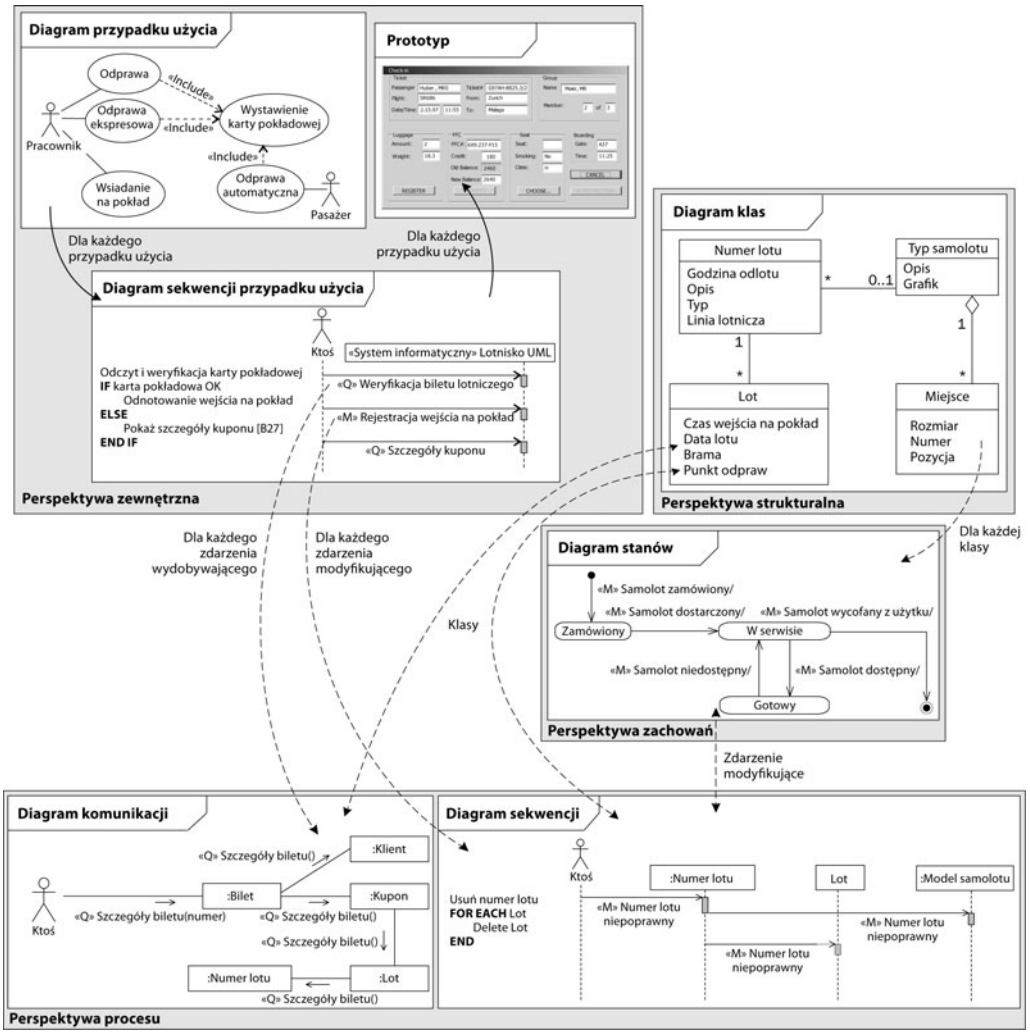
Rysunek 4.1. Różne perspektywy tego samego obiektu

Każda z tych perspektyw skupia się na pewnych aspektach i w związku z tym ignoruje inne. Wszystkie one w połączeniu tworzą stosunkowo kompletny model funkcjonalności systemu informatycznego.

- **Perspektywa zewnętrzna** — przedstawia przypadki użycia systemu informatycznego w postaci diagramów przypadków użycia oraz prototypu interfejsu. Dzięki temu jej odbiorca wie, jaką funkcjonalność będzie oferował system.
- **Perspektywa strukturalna** — przedstawia klasy systemu informatycznego w postaci diagramów klas. Dzięki niej wiadomo, jakie struktury informacji będą przetwarzane w systemie informatycznym.
- **Perspektywa zachowań** — przedstawia zachowanie poszczególnych obiektów w postaci diagramów stanów. Dzięki niej wiadomo, co może stać się z obiektami przetwarzanymi przez system informatyczny.
- **Perspektywa interakcji** — w postaci diagramów sekwencji i komunikacji przedstawia przepływy odbywające się w trakcie przetwarzania danych oraz zapytań w ramach systemu informatycznego. Dzięki niej widać, jakie operacje zachodzą w systemie informatycznym w wyniku interakcji z użytkownikiem.

Rzeczywistymi więzami łączącymi te wszystkie perspektywy w całość są zdarzenia. Występują one w trzech z czterech perspektyw:

- W **perspektywie zewnętrznej** indywidualne przypadki użycia są opisane jako sekwencje zdarzeń wysyłane do systemu informatycznego.
- **Perspektywa zachowań** przedstawia sposób reakcji każdej z klas na zdarzenia.
- **Perspektywa interakcji** przedstawia sposób, w jaki indywidualne zdarzenia w systemie informatycznym są przekazywane do odpowiednich obiektów.



Rysunek 4.2. Różne perspektywy systemu informatycznego

Jedynie diagram klas w perspektywie strukturalnej nie zawiera informacji o zdarzeniach. Diagram klas przedstawia klasy oraz ich powiązania, nie ma w nim jednak żadnych informacji o dynamice między nimi.

W naszym studium przypadku nie będziemy wykorzystywać wszystkich typów diagramów dostępnych w ramach UML-a. W praktyce proponowana przez nas kombinacja diagramów okazuje się wystarczającym wyborem z punktu widzenia modelowania systemów informatycznych. Za jej pomocą można zbudować spójny i kompletny model systemu informatycznego. W przypadku innych modeli optymalna kombinacja diagramów byłaby oczywiście inna. Na przykład w przypadku modeli biznesowych bardzo użyteczny okazuje się diagram aktywności.

W kolejnych podrozdziałach omówimy każdą z czterech perspektyw. W praktyce ich tworzenie nie zawsze odbywa się zgodnie z kolejnością, jaka została przyjęta w tej książce. Okazuje się bowiem, że praca nad jedną z perspektyw ujawnia nowe szczegóły przydatne w innej. Strzałki narysowane przerywaną linią na rysunku 4.2 sygnalizują najważniejsze z powiązań między perspektywami.

4.1. Perspektywa zewnętrzna

4.1.1. Perspektywa użytkownika, czyli „ważne, że działa, nieważne, jak"

W dzisiejszych czasach użytkownik praktycznie dowolnego nowoczesnego urządzenia, takiego jak kamera wideo, bankomat czy telefon komórkowy, rzadko jest zainteresowany szczegółami jego działania. Nie zaprząta sobie głowy elektroniką, z której składa się urządzenie, nie przejmuje się też oprogramowaniem, które steruje jego działaniem. Z drugiej strony ważne są dla niego możliwości urządzenia, czyli jego funkcjonalność. Na przykład nabywca telefonu komórkowego chce wiedzieć, czy potrafi on łączyć się za pomocą technologii WAP oraz ile jest w stanie zapisać numerów telefonów. Z reguły jest też zainteresowany możliwościami wykorzystania tego urządzenia, nie ma natomiast ochoty zapoznawać się ze szczegółami jego wewnętrznej budowy — pod warunkiem, że jest w stanie skorzystać z potrzebnych mu funkcji.

Tego typu spojrzenie na system określa się mianem **czarnej skrzynki**. System lub urządzenie są postrzegane jak czarna skrzynka — użytkownik nie może zajrzeć do środka. Nie wiadomo, jak coś działa, wiadomo jednak, co potrafi zrobić. W idealnym przypadku użytkownik powinien mieć dokładnie takie wyobrażenie systemu informatycznego. Powinien on móc wykorzystać system informatyczny do swoich celów tak samo, jak ekspres do kawy czy kserokopiarkę. Jeśli wie, co może zrobić z systemem, z reguły będzie też wiedział, w jaki sposób ma to robić.

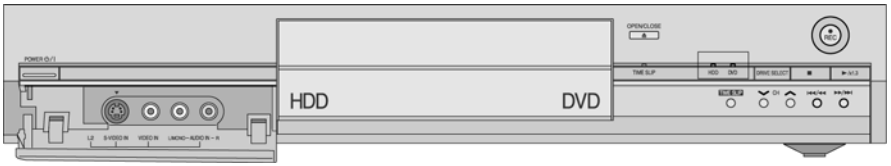
Perspektywa zewnętrzna jest ważnym elementem modelu systemu informatycznego. Za jej pomocą określa się oczekiwania przyszłych użytkowników w stosunku do systemu. Funkcjonalność zdefiniowana w tej perspektywie powinna być wykorzystana do weryfikacji systemu informatycznego pod kątem spełniania założonych wymagań.

Perspektywa zewnętrzna składa się z diagramów przypadków użycia, diagramów sekwencji przypadków użycia oraz prototypów interfejsów. Na pierwszy rzut oka to połączenie elementów może wydać się dość przypadkowe. Niektórzy analitycy mogą na przykład stwierdzić, że wymagania użytkowników wystarczy ująć w postaci opisu słownego. Doświadczenie wynikające z praktyki dowodzi jednak, że to nie jest prawda. Opis może być niespójny, nieprecyzyjny i niekompletny, a czytelnik może mieć problem z jego interpretacją zgodną z założeniami twórcy. System informatyczny jest tworzony tak, jak programista zinterpretuje podany mu

opis ze swojego punktu widzenia. Diagramy zaimplementowane w UML-u, opisane i objaśnione w tym rozdziale, mają na celu zapobieżenie nieporozumieniom i błędnym interpretacjom. Diagramy są narzędziami służącymi do opisywania wymagań w stosunku do systemów informatycznych.

Kluczowym elementem systemu informatycznego jest jego interfejs użytkownika. Interfejs użytkownika bankomatu składa się z małego monitora, klawiatury i otworów na kartę, banknoty i pokwitowanie. Ważnym elementem interfejsu bankomatu jest również sygnał dźwiękowy.

Interfejs użytkownika jest jedynym punktem kontaktowym użytkownika z systemem. Jeśli na przykład zabraknie przycisku nagrywania w magnetowidzie, użytkownik nie będzie miał możliwości nagrania audycji nawet wówczas, gdy urządzenie będzie wewnętrznie wyposażone w niezbędne podzespoły i oprogramowanie.



Rysunek 4.3. Zewnętrzna perspektywa systemu postrzeganego jako czarna skrzynka

Interfejs użytkownika pokazuje jedną z możliwości spojrzenia na funkcjonalność urządzenia (jedną perspektywę). Wszystko, czego zabraknie w tej perspektywie, będzie dla użytkownika niedostępne.

Interfejs użytkownika jest jednak wyrazem statycznego spojrzenia na system. Perspektywa tego typu nie pokaże tego, w jaki sposób system może być używany oraz w jaki sposób należy obsługiwać jego poszczególne elementy, aby osiągnąć określone cele.

Z tego powodu interfejsy użytkownika wymagają instrukcji obsługi. Oznacza to, że potrzebny jest opis identyfikujący dostępne akcje oraz sekwencje działań niezbędne, by system generował oczekiwane wyniki. W UML-u tego typu połączenia działań określa się terminem przypadków użycia. **Przypadki użycia** są podręcznikami użytkowania interfejsu użytkownika. Tyko te funkcje, które zostaną opisane w takim podręczniku, stanowią istotne z punktu widzenia użytkownika sekwencje działań.

W przypadku telefonu znaczącą sekwencją działań (przepływem) jest: podniesienie słuchawki, poczekanie na sygnał, wybranie prawidłowego numeru, poczekanie na odebranie połączenia przez odbiorcę, przeprowadzenie rozmowy, zakończenie połączenia.

Przepływów niezdefiniowanych w instrukcji obsługi nie uważa się za istotne dla systemu, a w konsekwencji nie są one uznane za obsługiwane przez niego. W źle zaimplementowanym systemie może zdarzyć się, że niezdefiniowane przepływy spowodują nieoczekiwane konsekwencje lub wręcz doprowadzą system do awarii.

Czasem w nowoczesnych systemach informatycznych istnieje możliwość dość swobodnego rozwiązywania problemów. W szczególności dotyczy to analizy danych. Z natury tego typu problemów wynika, że nie ma możliwości opisania ich w dokumentacji. W takich przypadkach istnieje konieczność znalezienia alternatywnych sposobów opisu, na przykład w postaci specyfikacji środowiska służącego do rozwiązywania problemów. Przypadek użycia zamiast opisu rozwiązania zawiera odwołanie do narzędzia raportującego lub specyfikacji języka zapytań.

Dla użytkowników system składa się z interfejsu użytkownika i przypadków użycia, choć z reguły te zagadnienia stanowią jedynie wierzchołek góry lodowej. Wszystko poniżej poziomu „wody" nie jest już dla użytkownika interesujące. Czy ktokolwiek zastanawiał się, jakie mechanizmy wewnętrzne są ukryte pod klawiaturą telefonu komórkowego?

> Ćwiczenie: zapisz przypadki użycia systemu, którego używasz na co dzień. To może być na przykład bankomat, ekspres do kawy albo portal WWW.

Najlepszy sposób modelowania przypadków użycia w systemie informatycznym polega na wyobrażeniu sobie użytkownika siadającego przed klawiaturą i po prostu pracującego z systemem (rysunek 4.4). Ten użytkownik staje się aktorem, a przypadek użycia to po prostu abstrakcyjny opis jego działań.

Rysunek 4.4. Aktor wykorzystujący przypadek użycia

Aktor:

- wykonuje działania bezpośrednio w systemie;
- zawsze znajduje się na zewnątrz systemu, z którym pracuje.

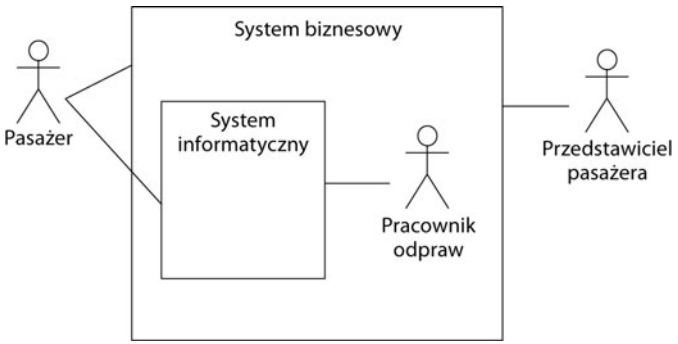
W przypadku systemu informatycznego oznacza to, że aktor jest zawsze operatorem systemu.

Nawet w przypadku systemów biznesowych (zobacz rozdział 3.) aktorzy są zewnętrznymi uczestnikami procesów. Mogą nimi być na przykład klienci lub partnerzy biznesowi.

Pracownik obsługujący system informatyczny jest jednak częścią systemu. Z tego właśnie powodu nie może być aktorem w systemie biznesowym.

Ta sytuacja została przedstawiona na rysunku 4.5.

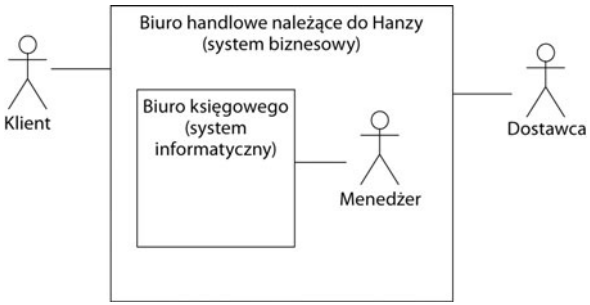
- **Pasażer** jest aktorem systemu biznesowego i ma kontakt przede wszystkim z pracownikiem odpraw. Może również być aktorem w systemie informatycznym, na przykład wówczas, gdy obsługuje komputer zawiadujący odprawami automatycznymi.
- **Pracownik odpraw** jest częścią systemu biznesowego. Z tego powodu nie może być aktorem w systemie biznesowym. Z drugiej strony, może być aktorem w systemie informatycznym.
- **Przedstawiciel pasażera**, który w jego imieniu dokonuje odprawy, jest aktorem w systemie biznesowym, lecz nie może być aktorem w systemie informatycznym, ponieważ zgodnie z przyjętymi procedurami nie może dokonywać odprawy automatycznej — nie ma zatem bezpośredniego kontaktu z systemem informatycznym.



Rysunek 4.5. Aktorzy systemu biznesowego i systemu informatycznego

Można zatem uznać, że aktorzy są tak samo jak przypadki użycia doskonale dostosowani do komunikacji z użytkownikami lub ekspertami w danej dziedzinie. Za ich pomocą można ustalać zewnętrzną funkcjonalność systemów informatycznych.

W przypadku biura handlowego naszego kupca stowarzyszonego w Hanzie oddzielenie systemu biznesowego od informatycznego jest dość kłopotliwym zadaniem. Można jednak przyjąć, że samo biuro oraz pracujący w nim urzędnicy wraz z sekretarzem Hindelbrandem tworzą funkcjonalność systemu informacyjnego (oczywiście nie informatycznego). W tej sytuacji przypadek użycia będzie odpowiadał zadaniom realizowanym przez Hindelbranda, takim jak aktualizacja potwierdzeń wpłaty czy podsumowywanie kosztów i dochodów z poprzedniego miesiąca. Aktorem jest tu sam właściciel, pan Hafenstein (rysunek 4.6).



Rysunek 4.6. Aktorzy w biurze handlowym kupca zrzeszonego w Hanzie

4.1.2. Elementy perspektywy

Perspektywa zewnętrzna systemu informatycznego (rysunek 4.7) składa się z następujących elementów:

- Diagramy przypadków użycia przedstawiające użytkowników systemu informatycznego (**aktorów**) oraz wszystkie zadania wykonywane przez tych użytkowników (**przypadki użycia**). W dalszej części rozdziału będziemy używać pojęcia diagram przypadków użycia (w liczbie pojedynczej). W praktyce z reguły nie ma sensu ograniczać się do jednego diagramu, ponieważ w większości przypadków byłby on po prostu przeciążony szczegółami.
- Diagramy sekwencji przypadków użycia przedstawiające przepływy interakcji między użytkownikiem a systemem informatycznym w ramach poszczególnych przypadków użycia.
- Prototypy interfejsów przedstawiające propozycje wyglądu interfejsów użytkownika.

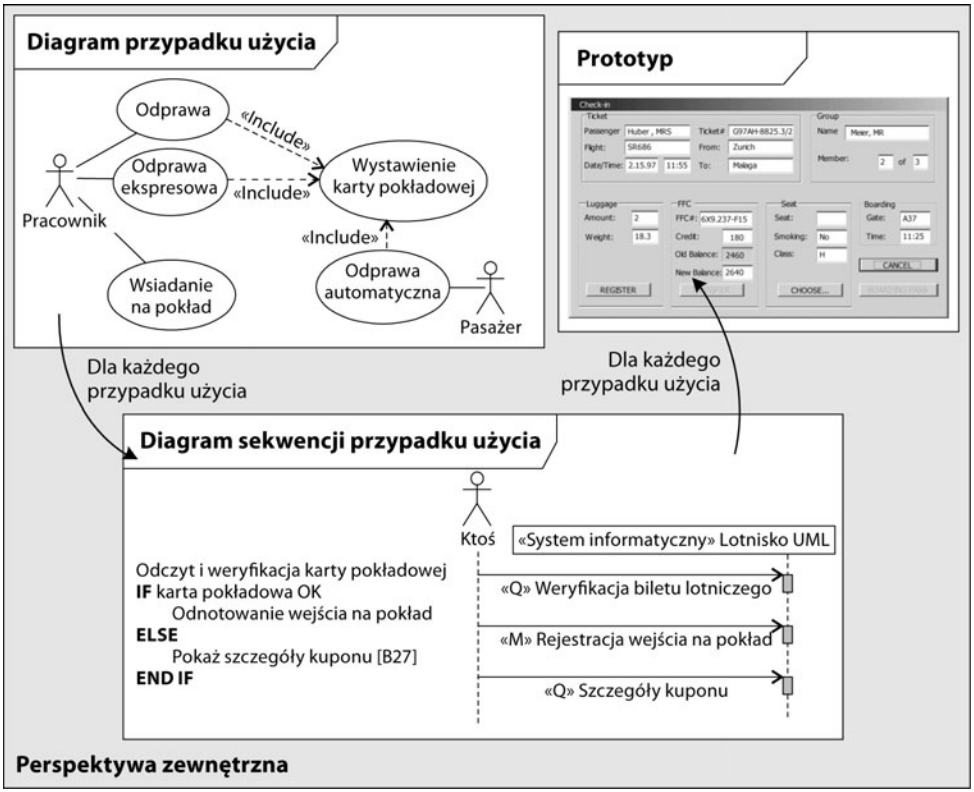
Wszystkie te elementy połączone ze sobą dają dobry ogląd systemu informatycznego z perspektywy użytkownika. W kolejnych punktach podrozdziału omówimy każdy z wymienionych wyżej elementów.

4.1.3. Diagram przypadków użycia

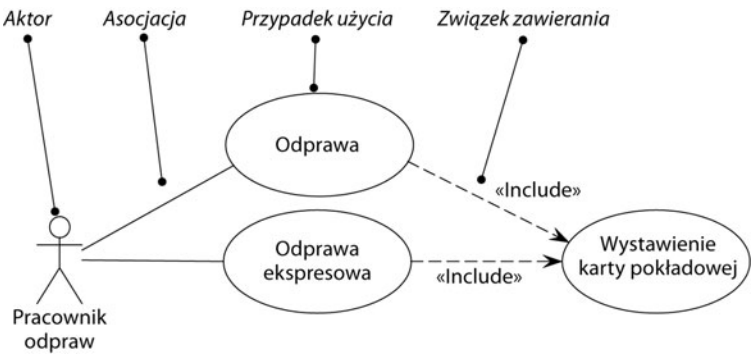
W diagramach przypadków użycia, których przykład przedstawia rysunek 4.8, pracujemy z następującymi elementami:

Aktor

Aktora można określić mianem użytkownika systemu informatycznego. Może to być na przykład pani Basia i pani Krysia z punktu odpraw. Ponieważ indywidualne osoby nie mają znaczenia dla modelu, wykorzystuje się ich abstrakcyjne definicje. Z tego powodu aktorzy otrzymują nazwy typu „pracownik odpraw" czy też „pasażer".



Rysunek 4.7. Perspektywa zewnętrzna



Rysunek 4.8. Elementy diagramu przypadków użycia

Aktorzy reprezentują role realizowane przez użytkowników systemu informatycznego — takie jak na przykład rola pracownika odpraw. Jedna osoba może pełnić więcej niż jedną rolę w systemie IT. Z punktu widzenia systemu informatycznego większe znaczenie ma to, jaką rolę realizuje osoba, niż to, kim jest ona w rzeczywistości. Z tego powodu ważne jest logowanie się

w systemie informatycznym „do określonej roli". Może to być na przykład rola zwykłego użytkownika lub administratora. W każdej z tych ról można wykorzystywać określone, specyficzne dla niej funkcje systemu.

Aktorzy nie są częścią systemu informatycznego. Jako pracownicy mogą jednak wchodzić w skład systemu biznesowego (rysunek 4.5).

Przypadek użycia

Przypadki użycia opisują interakcje występujące między aktorami a systemem informatycznym podczas wykonywania procesów biznesowych.

Przypadek użycia reprezentuje funkcję systemu informatycznego dostępną dla użytkownika (w postaci aktora).

Każda funkcja dostępna dla użytkownika w ramach systemu informatycznego powinna być zamodelowana jako przypadek użycia (lub część przypadku użycia). Funkcje istniejące w systemie informatycznym, a niedostępne przy użyciu normalnych metod, nie są dostępne dla użytkownika i nie należy tworzyć dla nich przypadków użycia.

Mimo że koncepcja przypadków użycia polega na opisywaniu interakcji, przepływy przetwarzania automatycznego, które z reguły nie wymagają interakcji z użytkownikiem, również bywają opisywane w diagramach przypadków użycia. Aktorem takiego przetwarzania automatycznego jest osoba inicjująca przetwarzanie. Przykładem takiego przypadku użycia może być **wygenerowanie statystyk odpraw**.

Powiązanie

Powiązanie informuje o zaistnieniu kontaktu aktora z przypadkiem użycia. Oznacza to, że aktor ma możliwość wykorzystania przypadku użycia. Diagram, na którym kilku aktorów jest powiązanych z jednym przypadkiem użycia, wskazuje na możliwość wykorzystania tego przypadku użycia przez każdego z nich, nie oznacza natomiast, że muszą (lub mogą) oni wykorzystywać ten przypadek użycia jednocześnie.

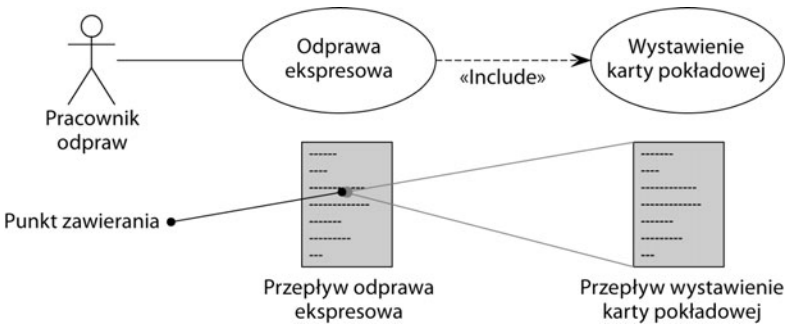
Zgodnie z definicją UML-a powiązanie oznacza jedynie tyle, że aktor **bierze udział** w przypadku użycia. My proponujemy wykorzystywać powiązania w nieco ściślej zdefiniowanym kontekście.

Związek zawierania

Związek zawierania występuje między dwoma przypadkami użycia.

Oznacza on, że przypadek użycia wskazywany grotem strzałki jest zawarty w przypadku użycia, z którego strzałka wychodzi. Dzięki tego typu związkowi mamy możliwość wykorzystania raz zdefiniowanego przypadku użycia w innych przypadkach użycia.

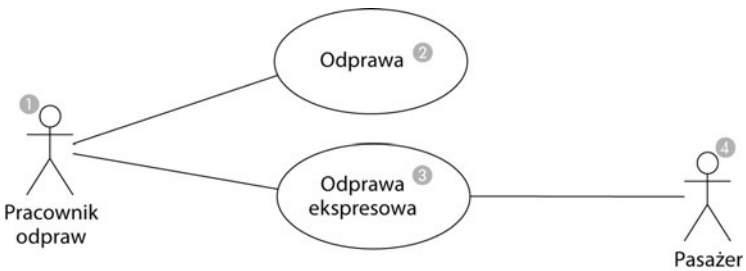
Rysunek 4.9 przedstawia przykład związku tego typu. W przepływie przypadku użycia **odprawa ekspresowa** jest zawarty przypadek użycia **wystawienie karty pokładowej**. Oznacza to, że w tym miejscu wykonywany jest przypadek użycia związany z wystawieniem karty pokładowej. Związki zawierania można postrzegać jako formę podprogramu (procedury).



Rysunek 4.9. Związki zawierania między przypadkami użycia

Czytanie diagramów przypadków użycia

Rysunek 4.10 przedstawia diagram przypadków użycia zawierający zarówno aktorów (**pracownik odpraw** i **pasażer**), jak i przypadki użycia **odprawa** i **odprawa ekspresowa**.



Rysunek 4.10. Prosty diagram przypadków użycia

W zależności od celu analizy diagram przypadków użycia można czytać, począwszy od aktora lub przypadku użycia.

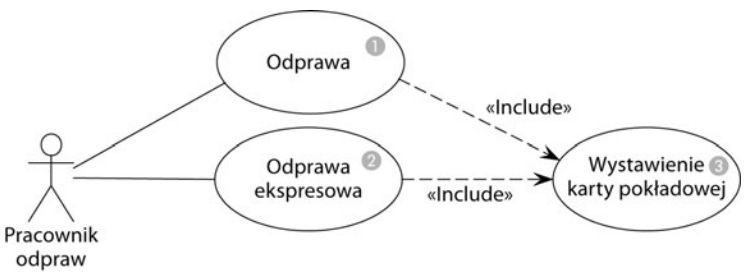
Jeśli rozpoczniemy od aktora **pracownik odpraw** (1), znajdziemy powiązania z dwoma przypadkami użycia: **odprawa** (2) i **odprawa ekspresowa** (3). Oznacza to, że osoba wykorzystująca system informatyczny w roli pracownika odpraw ma dostęp do funkcji związanych z przypadkami użycia **odprawa** i **odprawa ekspresowa**.

Dla czytelności diagramu przypadki użycia można umieszczać jeden nad drugim. Nie ma to żadnego znaczenia merytorycznego. Z diagramu przypadku użycia nie należy wyciągać wniosków na temat kolejności realizacji funkcji przez użytkownika.

Przypadki użycia **odprawa** (2) i **odprawa ekspresowa** (3) na tym etapie wiedzy o systemie są jedynymi, z którymi ma kontakt pracownik odpraw. Aktor **pasażer** (5) ma powiązanie z przypadkiem użycia **odprawa ekspresowa** (3), co oznacza, że osoby kontaktujące się z systemem informatycznym w roli pasażerów mogą bezpośrednio obsługiwać funkcje związane z odprawą ekspresową. Aktor **pracownik odpraw** (1) również ma powiązanie z przypadkiem użycia **odprawa ekspresowa** (3), co oznacza, że zarówno **pasażer**, jak i **pracownik odpraw** mogą wykorzystywać funkcje systemu informatycznego związane z taką odprawą. Nie oznacza to jednak, że w odprawie ekspresowej muszą brać udział obaj ci aktorzy.

Oczywiście, w przypadku użycia **odprawa** (2) pasażer zgłasza się do pracownika odpraw, lecz jako aktor systemu informatycznego nie ma bezpośredniej możliwości obsługi funkcji z tym związanych. W przypadku użycia **odprawa ekspresowa** (3) zarówno **pasażer**, jak i **pracownik odpraw** mają dostęp do odpowiednich funkcji systemu. Można założyć, że w przypadku pracownika jest to sytuacja wyręczania pasażera w obsłudze systemu. Z punktu widzenia systemu biznesowego pasażer zawsze jest aktorem, ponieważ nie wchodzi on w skład systemu biznesowego.

Rysunek 4.11 przedstawia diagram przypadków użycia uwzględniający związki zawierania między przypadkami użycia **odprawa** (1) oraz **odprawa ekspresowa** (2) a przypadkiem użycia **wystawienie karty pokładowej** (3). Oznacza to, że zarówno podczas odprawy, jak i podczas odprawy ekspresowej dochodzi do wystawienia karty pokładowej. Z naszego doświadczenia wynika, że taka forma to najprostszy sposób wielokrotnego wykorzystania wspólnych elementów przypadków użycia.



Rysunek 4.11. Diagram przypadku użycia ze związkami zawierania

4.1.4. Zdarzenia wydobywające i modyfikujące dane

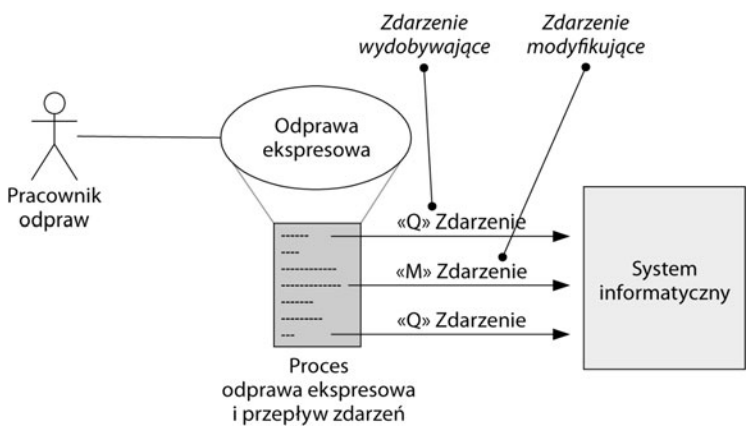
Zdarzeniem w UML-u jest specyfikacja wystąpienia, czyli opis czegoś, co się może wydarzyć. W kontekście przypadków użycia zdarzeniem można nazwać sytuację występującą w systemie informatycznym. Zdarzenia są inicjowane przez użytkowników za pośrednictwem interfejsu użytkownika, na przykład wskutek kliknięcia przycisku *Wyszukaj* lub naciśnięcia klawisza *Enter*. Zdarzenie powoduje wykonanie określonych działań w systemie informatycznym.

- **Zdarzenia związane z wydobywaniem informacji** (**zapytania**, ang. *queries*) są zdarzeniami, których celem jest wyświetlenie informacji. Z reguły nie powodują one zmian w systemie informatycznym. Zdarzenia wydobywające dane skutkują zaprezentowaniem użytkownikowi wyników.
- **Zdarzenia modyfikujące dane** (ang. *mutation events*) mają na celu zapis, modyfikację lub usunięcie informacji w systemie informatycznym. Wynik zdarzenia modyfikującego jest zależny od wyniku samej modyfikacji: w przypadku powodzenia informacja zostaje poprawnie zapisana, zmodyfikowana lub usunięta, a użytkownikowi jest przedstawiana stosowna informacja. W przypadku niepowodzenia w systemie informatycznym nie zachodzi żadna zmiana, a informacja o tym stanie również musi zostać przekazana użytkownikowi.

UML nie rozróżnia między zdarzeniami modyfikującymi a zdarzeniami wydobywającymi. Do naszych celów posługujemy się stereotypami rozszerzeń dostępnymi w specyfikacji UML-a. Są to mechanizmy pozwalające na tworzenie rozszerzeń i własnych elementów UML-a. Język ten rozszerzyliśmy zatem na nasze potrzeby, tworząc dwa specjalne przypadki zdarzeń:

- Stereotyp **«Q»** informuje, że mamy do czynienia ze zdarzeniem specjalnym typu „wydobycie danych" (**Q**uery);
- Stereotyp **«M»** informuje, że mamy do czynienia ze zdarzeniem specjalnym typu „modyfikacja danych" (**M**utation).

Dzięki temu możemy opisywać zdarzenia w różnych typach diagramów, zachowując informacje na temat charakteru zdarzeń. Rysunek 4.12 przedstawia zdarzenia wydobywające dane oraz zdarzenia modyfikujące w ramach przypadku użycia. Reakcja systemu informatycznego na zdarzenia nie jest opisana w postaci osobnego zdarzenia. Każde zdarzenie wydobywające lub modyfikujące wywołuje odpowiednią informację zwrotną: otrzymanie informacji lub komunikat o błędzie.



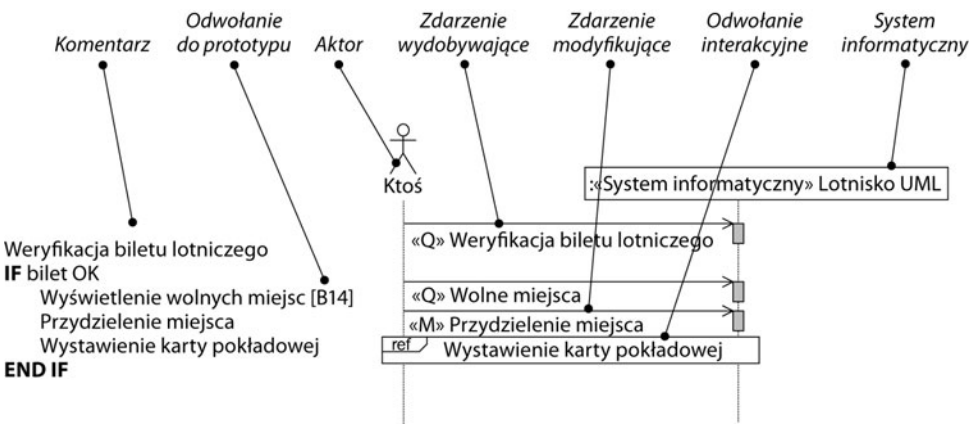
Rysunek 4.12. Zdarzenia wydobywające dane «Q» oraz zdarzenia modyfikujące «M»

W biurze handlowym kupca zrzeszonego w Hanzie zdarzenia możemy opisać w następujący sposób:

- Sekretarz Hildebrand sprawdza sumę pieniędzy przesłaną przez agenta z Rygi w ciągu ostatnich sześciu miesięcy. To jest **zdarzenie wydobywające «Q»**.
- Sekretarz Hildebrand podlicza rodzaje futer i kwoty zapłacone za nie do tej pory przez klientów z Rosji. To jest **zdarzenie wydobywające «Q»**.
- Sekretarz Hildebrand wprowadza do ksiąg informacje o tym, że należy zaprzestać współpracy z warsztatem farbiarskim mistrza Schildknechta. To jest **zdarzenie modyfikujące «M»**.

4.1.5. Diagram sekwencji przypadku użycia

Diagram sekwencji przypadku użycia jest zalecanym przez nas specjalnym sposobem stosowania diagramów sekwencji UML-a (rysunek 4.13). Diagramy sekwencji omówimy szczegółowo w podrozdziale 4.4, „Perspektywa interakcji".



Rysunek 4.13. Elementy diagramu sekwencji przypadku użycia

W diagramach sekwencji przypadku użycia mamy do czynienia z następującymi elementami:

Komentarz

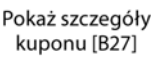
Przepływ przypadku użycia jest opisany za pomocą kombinacji tekstu i diagramu sekwencji.

Weryfikacja karty pokładowej
IF karta pokładowa OK
...

Komentarze mogą opisywać przepływ przypadku użycia w sposób uproszczony. UML pozwala umieszczać je na diagramach w dowolny sposób.

Odwołanie do prototypu

W komentarzach można umieszczać odwołania do formularzy ekranowych, wydruków i innych elementów interfejsu użytkownika.



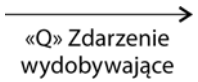
W ten sposób utworzyliśmy połączenie między diagramem sekwencji przypadku użycia a prototypem.

Aktor „ktoś"

Aktor „ktoś" reprezentuje dowolnego aktora diagramu przypadku użycia. „Ktoś" jest źródłem zdarzeń trafiających do systemu informatycznego w ramach przypadku użycia.

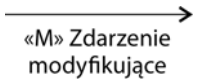
Przypadek użycia może być wykorzystywany przez różnych aktorów. Aktorem o nazwie „ktoś" posługujemy się w celu uogólnienia. Dzięki temu nie musimy wskazywać rzeczywistego aktora.

Zdarzenie wydobywające



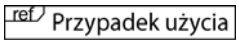
Zdarzenie wydobywające dane jest wysyłane do systemu informatycznego w celu uzyskania informacji (zobacz punkt 4.1.4, „Zdarzenia wydobywające i modyfikujące dane").

Zdarzenie modyfikujące



Zdarzenie modyfikujące jest zdarzeniem wysyłanym do systemu informatycznego w celu wprowadzenia modyfikacji w danych (zobacz punkt 4.1.4, „Zdarzenia wydobywające i modyfikujące dane").

Odwołanie interakcyjne



Odwołanie interakcyjne informuje, że w tym miejscu diagramu sekwencji przypadku użycia wywoływany jest inny diagram (zobacz punkt 4.1.3, „Diagram przypadków użycia").

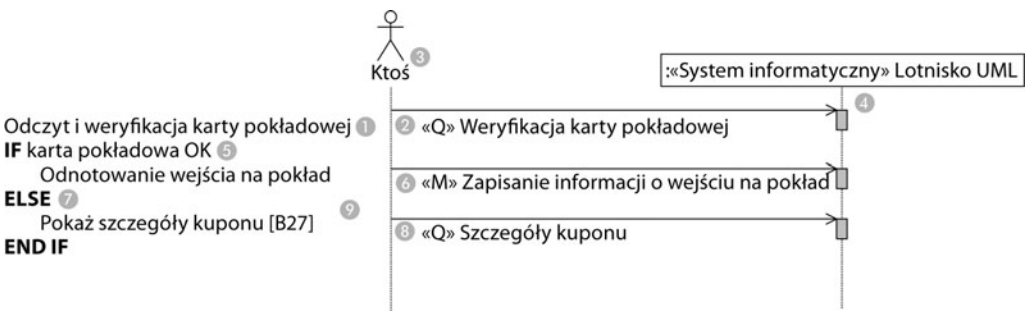
System informatyczny

System informatyczny jest czarną skrzynką zawierającą wszystkie obiekty i funkcjonalność.

Wszystkie zdarzenia wywołane przez przypadek użycia trafiają do systemu informatycznego. W perspektywie zewnętrznej nie interesuje nas to, które obiekty w systemie informatycznym biorą udział w obsłudze zdarzeń.

Czytanie diagramów przypadków użycia

Rysunek 4.14 przedstawia diagram sekwencji przypadku użycia **wsiadanie na pokład**. Taki diagram zawsze należy do przypadku użycia, ponieważ opisuje interakcje w ramach przepływów przypadku użycia. Informacje dotyczące przepływu są wpisane w komentarzu po lewej stronie. Po pierwsze, sprawdzana (1) jest ważność karty pokładowej. Do tego służy zdarzenie wydobywające **«Q» weryfikacja karty pokładowej** (2) zainicjowane przez aktora przypadku użycia (3) w systemie informatycznym (4). W jaki sposób zdarzenie jest obsługiwane wewnętrznie przez system — tego diagram nie przedstawia, ponieważ system jest w tej perspektywie postrzegany jako czarna skrzynka. Sprawdzana jest ważność karty pokładowej, co oznacza prawdopodobnie, że informacje na jej temat są wprowadzane do systemu i porównywane z zapisanymi w nim informacjami. Komunikat zwracany przez system w wyniku zdarzenia zawiera informację o wyniku tej weryfikacji. Wynik nie ma strzałki, jest po prostu odsyłany zwrotnie „wzdłuż strzałki zdarzenia" (2), tylko w przeciwnym kierunku.



Rysunek 4.14. Diagram sekwencji przypadku użycia

Jeśli karta pokładowa zostanie zweryfikowana pomyślnie (5), w systemie informatycznym można zapisać informację o tym, że pasażer wsiadł na pokład. Do tego służy zdarzenie modyfikujące **«M» zapisanie informacji o wejściu na pokład** (6), które jest wysyłane do systemu informatycznego (4). Ponownie nie widzimy, co dzieje się w systemie w wyniku tego zdarzenia. Jeśli karta pokładowa nie zostanie zweryfikowana pomyślnie (7), na ekranie komputera powinna pojawić się informacja o bilecie zawierającym błędną kartę pokładową. W tym celu do systemu wysyłane jest zdarzenie **«Q» wyświetlenie informacji o bilecie** (8). To zapytanie zwraca odpowiednie informacje, o ile znajdują się one w systemie informatycznym. Ekran

z informacjami o bilecie można zaprojektować w postaci prototypu interfejsu. Odwołanie [B27] (9) informuje o numerze prototypu interfejsu formularza ekranowego. Jak widać, cały przypadek użycia **wsiadanie na pokład** został opisany w postaci sekwencji zdarzeń wydobywających i modyfikujących dane.

4.1.6. Konstruowanie perspektywy zewnętrznej

Poniższa lista kontrolna zawiera etapy niezbędne do skonstruowania perspektywy zewnętrznej. W kolejnych punktach omówimy poszczególne etapy z tej listy.

Lista kontrolna 4.1. Konstruowanie diagramów w perspektywie zewnętrznej

- Gromadzenie źródeł informacji — skąd mam to wiedzieć?
- Identyfikacja potencjalnych aktorów — kto pracuje z systemem informatycznym?
- Identyfikacja potencjalnych przypadków użycia — co można zrobić z systemem informatycznym?
- Połączenie aktorów i przypadków użycia — jakie czynności może wykonać każdy z użytkowników systemu informatycznego?
- Opisanie aktorów — kogo lub co reprezentują?
- Poszukiwanie dalszych przypadków użycia — jakie inne funkcje ma realizować system informatyczny?
- Modyfikacja przypadków użycia — co należy uwzględnić, a co pominąć w przypadku użycia?
- Udokumentowanie przypadków użycia — jakie zdarzenia występują w przypadku użycia?
- Modelowanie powiązań między przypadkami użycia — co można wykorzystać ponownie?
- Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Proponowana kolejność jest zweryfikowana w praktyce pod kątem sensowności. Nie jest ona kolejnością jedynie słuszną, ponieważ w praktyce poszczególne etapy nakładają się wzajemnie.

Wymienione wyżej etapy są z reguły realizowane przez analityka, który musi mieć ogólne rozeznanie zarówno w technologiach informatycznych, jak i w specyfice systemów biznesowych (w zakresie omówionym w rozdziale 3.). Jednakże konieczna jest konsultacja analityka z dostawcami wiedzy, na przykład z użytkownikami systemu. W wyniku wymienionych wyżej etapów powstaje zewnętrzna perspektywa systemu informatycznego, która może być odczytana i zrozumiana przez ekspertów w dziedzinie.

Gromadzenie źródeł informacji — skąd mam to wiedzieć?

Istnieje kilka źródeł informacji, które można wykorzystać do zdefiniowania perspektywy zewnętrznej:

- Oczywiście, pierwszym źródłem informacji powinien być model biznesowy. Aktorzy systemu biznesowego, pracownicy i biznesowe przypadki użycia są doskonałym punktem początkowym i z nich możemy pozyskać pierwszych aktorów i przypadki użycia w modelu systemu informatycznego (zobacz rozdział 3.).

- Specyfikacje techniczne, projektowe i dokumenty tego typu.
- Przyszli użytkownicy są bardzo ważnym źródłem informacji w perspektywie zewnętrznej, charakterystycznej wszak dla punktu widzenia użytkownika.
- Eksperci techniczni z dziedziny zgodnej z tworzoną aplikacją.
- Diagramy organizacyjne, struktura organizacyjna i opisy stanowisk.

W tym zadaniu bardzo przydaje się przyjęcie punktu widzenia użytkownika. Warto rozmawiać z użytkownikami i obserwować ich przy pracy.

Identyfikacja potencjalnych aktorów — kto pracuje z systemem informatycznym?

Ten etap wiąże się z identyfikacją i wstępnym doborem aktorów. Dobór nie musi jednak być kompletny ani w pełni bezbłędny. W tym przypadku obowiązuje reguła „im więcej, tym lepiej". Aktorzy zidentyfikowani na tym etapie będą wykorzystywani na kolejnych.

W identyfikacji potencjalnych aktorów pomocne mogą okazać się odpowiedzi na następujące pytania, które warto zadać na przykład przyszłym użytkownikom systemu. W odpowiedziach należy unikać wskazywania konkretnych osób. Warto natomiast skupić się na grupach osób lub innych aktorach.

- Którzy aktorzy i pracownicy mają mieć bezpośredni dostęp do systemu informatycznego?
- Które grupy osób wymagają pomocy ze strony systemu informatycznego w celu wykonywania codziennej pracy?
- Które grupy osób wykorzystują najważniejsze funkcje systemu informatycznego?
- Które grupy osób wykorzystują drugorzędne funkcje systemu, takie jak utrzymanie i administracja?
- Które grupy osób zdefiniowane w modelu organizacyjnym (zobacz diagram organizacyjny) firmy lub departamentu pracują z systemem informatycznym?

Kluczowe jest zobrazowanie konkretnych i bezpośrednich interakcji z modelowanym systemem informatycznym.

W punkcie odpraw to pracownik odpraw jest aktorem pracującym z systemem informatycznym. Przy komputerze do odpraw automatycznych użytkownikiem systemu jest pasażer pozbawiony bagażu — wykorzystuje system dzięki biletowi, który może być wczytany i przeanalizowany w sposób automatyczny. Pasażer wkłada bilet do czytnika i wskazuje na ekranie jedno z niezajętych miejsc na pokładzie samolotu.

Pasażer jest również aktorem systemu biznesowego, natomiast pracownicy działu odpraw oraz inni pracownicy lotniska nie są takimi aktorami. Są pracownikami, więc wchodzą w skład systemu biznesowego, co zostało przedstawione na rysunku 4.15.

Rysunek 4.15. Potencjalni aktorzy

Identyfikacja potencjalnych przypadków użycia — co można zrobić z systemem informatycznym?

Na tym etapie mamy za zadanie dokonać pierwszej selekcji przypadków użycia. W tym miejscu również obowiązuje reguła „im więcej, tym lepiej". W identyfikacji potencjalnych przypadków użycia przydatne będą odpowiedzi na następujące pytania:

- Które biznesowe przypadki użycia są obsługiwane przez system informatyczny?
- Które biznesowe aktywności systemu biznesowego powinny być obsługiwane przez system informatyczny?
- W zależności od poziomu szczegółowości aktywności biznesowych na tym etapie można skonstruować przypadki użycia dla każdej z aktywności biznesowych.
- Jakie są cele i założenia systemu informatycznego?
- Jakie są podstawowe funkcje systemu informatycznego?
- Do czego aktorzy potrzebują systemu informatycznego?
- Jakich drugorzędnych funkcji, takich jak utrzymanie i administracja, wymaga system informatyczny?
- Jakie funkcje przewiduje prototyp interfejsu?

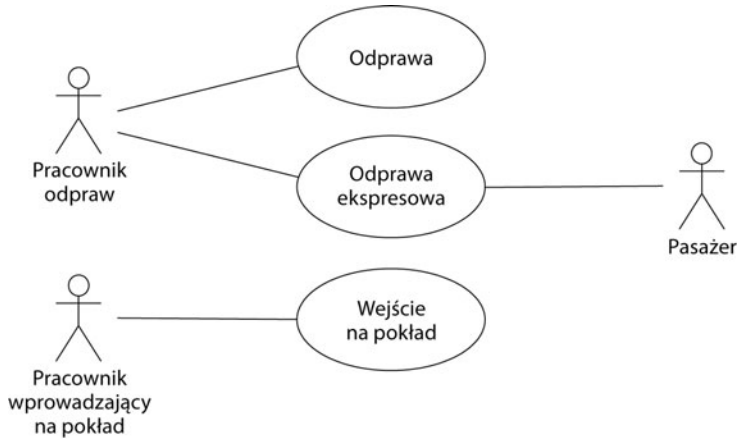
Połączenie aktorów i przypadków użycia — jakie czynności może wykonać każdy z użytkowników systemu informatycznego?

Przyporządkowanie aktorom biznesowych przypadków użycia pozwala nakreślić pierwszy szkic diagramu przypadku użycia. Przedstawiamy go na rysunku 4.16. Na tym etapie należy odpowiedzieć na pytanie:

- Którzy aktorzy mają dostęp do każdego z przypadków użycia?

Rysunek 4.16. Potencjalne przypadki użycia

Ten pierwszy szkic stanowi podstawę, w oparciu o którą można tworzyć i udoskonalać diagram przypadków użycia. Wersję udoskonaloną przedstawia rysunek 4.17.



Rysunek 4.17. Pierwszy szkic diagramu przypadków użycia

Opisanie aktorów — kogo lub co reprezentują?

Aktor na diagramie musi mieć nazwę w jasny sposób wskazującą jego rolę. Należy odpowiedzieć na pytanie:

- W jaki sposób aktor może być precyzyjnie opisany?

W tym miejscu bardzo ważne jest, aby wykorzystywana terminologia odpowiadała jak najlepiej środowisku, w którym będzie wykorzystywany model. Użytkownicy systemu informatycznego muszą utożsamić się z aktorami w przypadkach użycia, bowiem w przeciwnym wypadku nie zrozumieją znaczenia diagramów! Jeśli to konieczne, rola aktora może być dodatkowo opisana w komentarzu. Można w nim zawrzeć zakres obowiązków aktora lub wymagania w stosunku do systemu informatycznego z punktu widzenia aktora. Można też wykorzystać formalną definicję roli realizowanej przez aktora.

Poszukiwanie dalszych przypadków użycia — jakie inne funkcje ma realizować system informatyczny?

Po zidentyfikowaniu pierwszej grupy przypadków użycia można je wykorzystać jako punkt wyjścia do stworzenia kompletnego diagramu przypadków użycia. Przypadki przeoczone w pierwszym podejściu można zidentyfikować, zadając następujące pytania (dla każdego zidentyfikowanego przypadku użycia z osobna):

- Czy istnieją funkcje wykonywane za pomocą systemu informatycznego, które muszą być zrealizowane przed analizowanym przypadkiem użycia?
- Czy istnieją funkcje wykonywane za pomocą systemu informatycznego, które muszą być zrealizowane po analizowanym przypadku użycia?
- Czy istnieją funkcje wykonywane za pomocą systemu informatycznego, które muszą być zrealizowane, jeśli analizowany przypadek użycia nie jest wykonany?

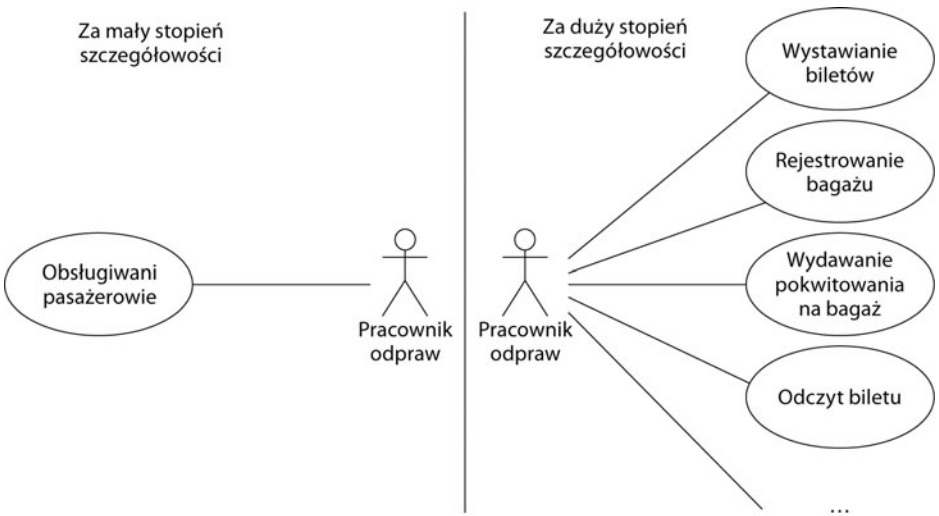
Bardzo ważne, aby nie stracić z oczu rzeczywistego systemu informatycznego. Istnieje ryzyko, że modelowane przypadki użycia zaczną wykraczać poza analizowany system informatyczny. Na przykład zakupienie biletu na samolot, akcja, która z pewnością musi nastąpić przed odprawą, nie należy do analizowanego systemu informatycznego obsługi pasażerów.

W naszym studium przypadku odpowiedzi na zadane powyżej pytania mogą być następujące:

- Przed odprawą należy uzyskać informacje na temat biletu i lotu.
- Po odprawie odbywa się wsiadanie do samolotu.
- Jeśli pasażer nie podda się w odprawie, jego bilet musi zostać unieważniony.

Modyfikacja przypadków użycia — co należy w nich uwzględnić, a co pominąć?

Najtrudniejszą częścią modelowania przypadków użycia jest dobranie właściwego poziomu szczegółowości diagramów. Mamy możliwość doboru poziomu pośredniego — dwie skrajności zostały przedstawione na rysunku 4.18.



Rysunek 4.18. Skrajne poziomy szczegółowości przypadków użycia w systemie informatycznym „obsługa pasażerów"

Żadne z tych skrajnych podejść nie ma praktycznego sensu. Jeśli cały system informatyczny zostanie wciśnięty w pojedynczy przypadek użycia, powstanie zupełnie nic nieznaczący diagram. Nie można z niego wyciągnąć żadnych użytecznych informacji na temat systemu. Z drugiej strony przy zbyt szczegółowym podejściu przypadki użycia mogą zostać podzielone na zbyt małe elementy. To powoduje, że diagram staje się za bardzo skomplikowany i zawiera dużą liczbę przypadków użycia z trudnymi do rozpoznania zależnościami.

Na szczęście istnieją kryteria pomagające znaleźć optymalny poziom szczegółowości przypadku użycia. Aby zapobiec zbytniemu ich rozrostowi, należy dla każdego z nich zadać następującą serię pytań:

- Czy przypadek użycia składa się z wzajemnie powiązanych, spójnych sekwencji interakcji?

  Wszystkie elementy ujęte na diagramie przypadku użycia muszą być wzajemnie powiązane. Wystawienie karty pokładowej i poszukiwanie zagubionego bagażu to operacje zupełnie ze sobą niezwiązane. Przypadki użycia niezgodne z tym kryterium należy rozbić na składowe.

- Czy pojedynczy aktor może wykonać wszystkie przedstawione na diagramie przypadki użycia?

  Mimo że UML pozwala na udział większej liczby aktorów w wykonaniu przypadku użycia, w większości sytuacji lepiej unikać takiego założenia. Jeśli przypadek użycia opisuje interakcję osoby z komputerem, należy przyjąć regułę, że nie powinny brać w niej udziału żadne dodatkowe osoby. Przypadki użycia niezgodne z tą regułą należy podzielić na składowe.

Aby uniknąć nadmiernego rozdrobnienia przypadków użycia, możemy zadać następujące pytania dla każdego zidentyfikowanego przypadku użycia:

- Czy przypadek użycia daje jednoznaczny i adekwatny wynik?

  Przypadek użycia nie może opisywać niekompletnego etapu pośredniego, na przykład „wybierz klienta". Powinien on generować wynik, który ma sens z ogólnego punktu widzenia. Przypadki użycia niezgodne z tą regułą należy połączyć z innymi.

- Czy przypadek użycia nie jest nigdy wykonywany samodzielnie, lecz zawsze w sekwencji z innymi przypadkami użycia?

  Przypadki użycia nie powinny opisywać etapów pośrednich wykonywanych tylko w połączeniu z innymi etapami pośrednimi. Jeśli ta reguła nie jest zachowana, należy te przypadki połączyć z innymi.

Weryfikacja istniejących przypadków użycia za pomocą powyższych pytań pozwala na określenie rozsądnego i jednolitego poziomu szczegółowości przez ich rozbijanie i łączenie.

Udokumentowanie przypadków użycia — jakie zdarzenia w nich występują?

Informacje uzyskane z diagramów przypadków użycia nie są wystarczające do zrozumienia samych przypadków użycia. Należy dodatkowo opisać przepływ interakcji reprezentowany przypadkami użycia. Oprócz czysto słownego opisu bardzo cenne okazują się diagramy sekwencji przypadków użycia. Dla każdego przypadku użycia możemy zadać poniższe pytania. Pozwolą one utworzyć diagramy sekwencji przypadków użycia.

- Jakie etapy pracy z systemem informatycznym można wyróżnić?

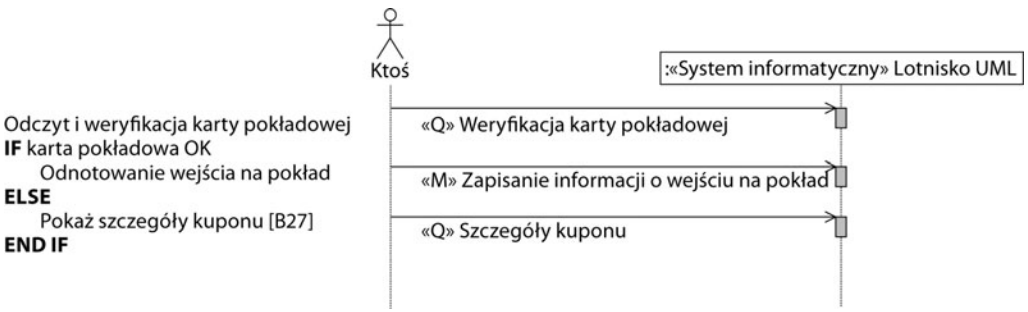
  Aby odpowiedzieć na to pytanie, należy przyjrzeć się aktorowi podczas pracy z systemem informatycznym. Jakie czynności wykonuje on w systemie? Jakie dane wprowadza? Jakie informacje wyświetla system? Jak wygląda interakcja? Trzeba znaleźć odpowiedni poziom szczegółowości.
- Jakich informacji przypadek użycia powinien dostarczać aktorowi?

  Jeśli informacja ma być pobrana z systemu, powinno do niego być wysłane zdarzenie wydobywające dane.
- Jakie informacje powinny być zapisywane, modyfikowane lub usuwane z systemu?

  Jeśli informacja powinna być zmieniona, do systemu musi być wysłane zdarzenie modyfikujące.

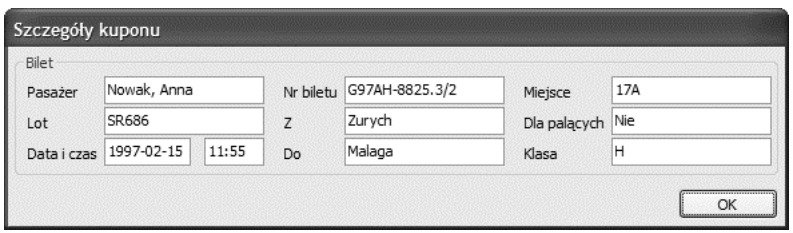
Opis przepływów jest zatem sekwencją etapów wprowadzania danych do systemu lub ich wydobywania. Innymi słowy, chodzi o interakcję użytkownika z systemem. Na etapie opisywania przepływów system informatyczny zawsze jest analizowany jako czarna skrzynka.

W rzeczywistości przypadek użycia nie zawsze zachodzi w taki sam sposób. Warto zatem stosować proste struktury kontrolne służące do reprezentowania rozgałęzień i alternatyw, co przedstawia rysunek 4.19.



Rysunek 4.19. Diagram sekwencji przypadku użycia „wsiadanie na pokład"

Dokumentacja przypadków użycia powinna zawierać również opisy analizowanego interfejsu użytkownika. Przykład takiego okna dialogowego (o etykiecie [B27]) przedstawia rysunek 4.20.



Rysunek 4.20. Okno dialogowe [B27] z przypadku użycia „wsiadanie na pokład"

Modelowanie powiązań między przypadkami użycia — co można wykorzystać ponownie?

Jeśli okaże się, że fragmenty interakcji są w wielu przypadkach takie same, ich części wspólne można ująć w osobnym przypadku użycia. Za pomocą związku zawierania można go włączyć do innych przypadków użycia, które go wykorzystują. Należy odpowiedzieć na pytanie:

- Czy istnieją części diagramów sekwencji przypadków użycia takie same, jak w innych diagramach (i czy zawsze pozostają takie same)?

Weryfikacja perspektywy — czy wszystko jest wykonane prawidłowo?

Zarówno diagramy przypadków użycia, jak i sekwencji przypadków użycia muszą być zweryfikowane z pomocą dostawców wiedzy. W idealnym przypadku dostawcy wiedzy mogą czytać i analizować diagramy samodzielnie (co nie jest zbyt trudne — nasza książka powinna służyć pomocą w zakresie każdej z perspektyw). Następnie osoby weryfikujące powinny stwierdzić, czy diagramy są kompletne i prawidłowe. Jeśli dostawcy wiedzy nie są w stanie samodzielnie odpowiedzieć na to pytanie, diagramy muszą zostać im odczytane. Następnie analityk wspólnie z dostawcami wiedzy musi ocenić ich kompletność i prawidłowość. Dopiero po zatwierdzeniu diagramów proces ich tworzenia można uznać za zakończony i stwierdzić, że zawierają one całą potrzebną wiedzę na temat tworzonego systemu informatycznego.

Kompletny diagram przypadków użycia można zweryfikować za pomocą listy kontrolnej 4.2.

Lista kontrolna 4.2. Weryfikacja diagramu przypadku użycia w perspektywie zewnętrznej

- Czy diagram jest kompletny?

  Diagram można uznać za kompletny wówczas, gdy nie istnieją inne, nieuwzględnione na nim przypadki użycia. Każda funkcja, którą użytkownik może wykorzystać w systemie, musi być ujęta w postaci przypadku użycia (jeśli to konieczne, przypadki użycia mogą być rozrysowane na wielu diagramach, jednak uznajemy je wszystkie za pojedynczy diagram przypadków użycia).

- Czy wszystkie przypadki użycia są prawidłowe?

  Przypadki użycia są prawidłowe, jeśli opisują funkcje systemu informatycznego i są zgodne z definicją przypadku użycia.

- Czy poziom szczegółowości diagramu jest odpowiedni?

  Poziom szczegółowości diagramu przypadków użycia powinien być zgodny z założonymi wymaganiami:

  - Przypadek użycia reprezentuje funkcjonalnie spójną sekwencję interakcji. Jest on wykonywany przez pojedynczego aktora.
  - Przypadek użycia opisuje zrozumiałe funkcje systemu, które służą do generowania konkretnych rezultatów.
  - Przypadek użycia powinien być wykonywany w całości.

- Czy przypadki użycia są nazwane prawidłowo?

  Nazwa każdego z przypadków użycia powinna opisywać odpowiadającą mu aktywność systemu informatycznego.

- Czy aktorzy są dobrani prawidłowo?

  Aktorzy w przypadku użycia reprezentują role wykonywane przez kogoś (osobę) lub coś (na przykład inny system) w interakcji z systemem informatycznym.

Gotowe diagramy sekwencji przypadków użycia można weryfikować w oparciu o listę kontrolną 4.3.

Lista kontrolna 4.3. Weryfikacja diagramów sekwencji przypadków użycia w perspektywie zewnętrznej

- Czy zostały zdefiniowane wszystkie diagramy sekwencji przypadków użycia?
  Każdy przypadek użycia powinien mieć zdefiniowany diagram sekwencji opisujący możliwe przepływy.
- Czy wszystkie diagramy sekwencji przypadków użycia są prawidłowe?
  Każdy diagram sekwencji przypadku użycia powinien zawierać tylko jeden obiekt reprezentujący system informatyczny i składać się wyłącznie ze zdarzeń wydobywających lub modyfikujących dane.

4.2. Perspektywa strukturalna

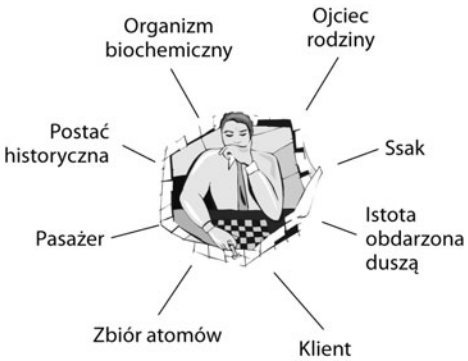
4.2.1. Obiekty i klasy

Fundamentem podejścia obiektowego jest jak najdoskonalsza reprezentacja zjawiska występującego w rzeczywistości — najpierw w postaci modelu, a następnie w postaci systemu informatycznego. Model nie może jednak być stuprocentową reprezentacją rzeczywistości. Każde rzeczywiste zjawisko, czy to istota żywa, czy obiekt, czy też koncepcja, jest tak skomplikowane i posiada tak wiele różnorodnych aspektów swojego istnienia, że niemożliwe jest uwzględnienie ich wszystkich w modelu.

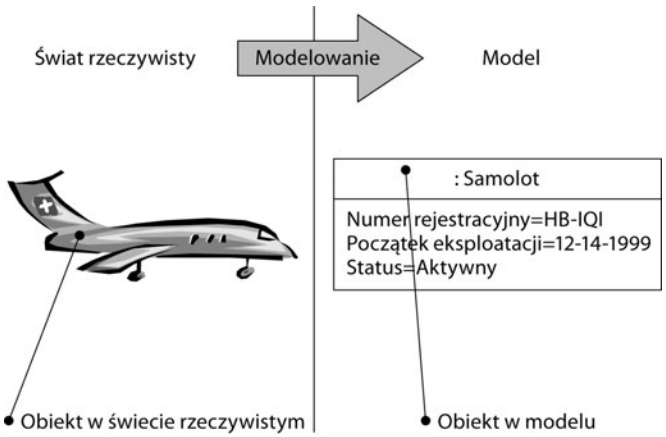
Aby stworzyć prawidłową reprezentację zjawiska w postaci modelu, należy skupić się na kilku ważnych zagadnieniach i zignorować pozostałe. Najważniejsze — to znaczy najbardziej interesujące — zagadnienia obiektu są podkreślane, pozostałe są pomijane. Na tym właśnie polega sztuka modelowania obiektów.

Jeśli chcemy skonstruować prawidłowy model, musimy wiedzieć, jakie cechy modelowanych zjawisk są potrzebne systemowi informatycznemu. Model „Jan Kowalski" będzie miał inną konstrukcję w przypadku systemu obsługi klientów, a inną w systemie informacji medycznej lub podatkowej (rysunek 4.21). Tylko wtedy, gdy wiadomo (choćby w przybliżeniu) jaki jest cel systemu informatycznego, można zbudować jego obiekty funkcjonalne.

Przy tworzeniu modeli zawsze dokonuje się uproszczeń pod kątem założonego celu. Nasze rozważania ograniczamy do najważniejszych aspektów założonego celu i pomijamy wszystko inne. Rysunek 4.22 przedstawia ten poziom abstrakcji na przykładzie samolotu.



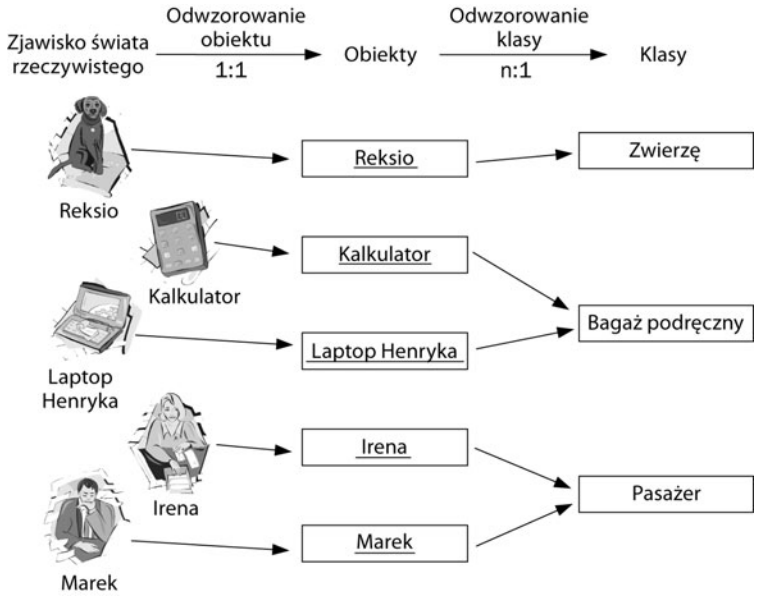
Rysunek 4.21. Różne perspektywy postrzegania Jana Kowalskiego



Rysunek 4.22. Modelowanie

Gdy w modelach abstrakcyjnych opisywane są zjawiska świata rzeczywistego, rozróżnia się dwa etapy ich tworzenia. Pierwszy polega na wygenerowaniu abstrakcji na podstawie konkretnych osób lub rzeczy i stworzeniu obiektów. Drugi etap to łączenie podobnych obiektów w klasy. Rysunek 4.23 przedstawia kilka sposobów modelowania zjawisk rzeczywistych do postaci obiektów, a potem na klas.

Bezpośrednią reprezentacją konkretnego istniejącego zjawiska jest obiekt. Pomiędzy tym modelowanym zjawiskiem a jego modelem istnieje relacja 1:1. Obiekt reprezentuje dokładnie jeden egzemplarz ze świata rzeczywistego. W bazach danych obiekt odpowiada na przykład jednemu wierszowi tabeli. Definicja obiektu to pierwszy krok w stronę stworzenia abstrakcji, ponieważ w obiekcie zamodelowane są jedynie wybrane cechy zjawiska. Na przykład w modelu **Marek** przedstawiającym osobę o imieniu Marek jej cechy redukuje się do najważniejszych z jakiegoś punktu widzenia. Tu na przykład sprowadza się ją do roli pasażera — interesuje nas imię, nazwisko i data urodzenia.



Rysunek 4.23. Generowanie obiektów i klas

Ćwiczenie: zapisz definicje kilkunastu obiektów ze swojego otoczenia.

Drugi etap abstrakcji polega na łączeniu podobnych obiektów w klasy. Przez **podobne** rozumiemy takie obiekty, w których:

- cel abstrakcji jest zbliżony;
- grupuje się je na podstawie zbliżonych charakterystyk;
- prezentują one zbliżone zachowanie.

W większości przypadków te dwa etapy tworzenia abstrakcji występują jednocześnie, co w praktyce oznacza, że klasy definiuje się bezpośrednio, bez jawnie wyodrębnionego etapu pośredniego w postaci identyfikacji obiektów.

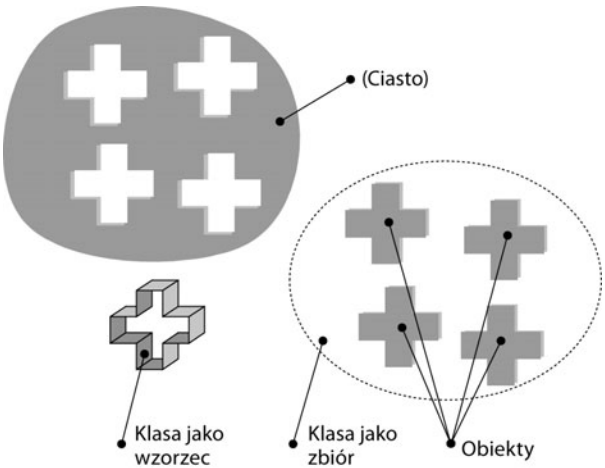
Modelowanie jest często bardziej skomplikowanym zadaniem, szczególnie w sytuacjach, gdy modelowane zjawisko istnieje jedynie jako koncepcja nieposiadająca fizycznej reprezentacji w rzeczywistości. Dawniej istniały fizyczne (papierowe) listy magazynowe czy książeczki oszczędnościowe w banku; obecnie tego typu dokumenty istnieją jedynie wirtualnie, w postaci informacji.

Ćwiczenie: spróbuj pogrupować w klasy obiekty zgromadzone w poprzednim ćwiczeniu.

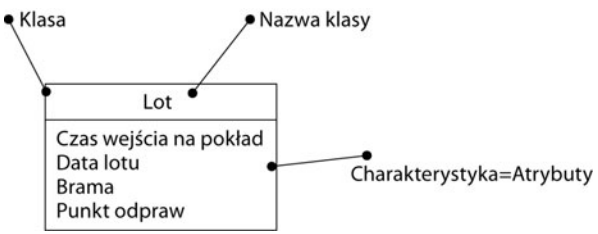
Praca z klasami staje się prostsza, gdy weźmie się pod uwagę, że termin **klasa** ma dwa nieco odmienne znaczenia:

- Klasa to **wzorzec**, w oparciu o który tworzy się obiekty.
- Klasa to **zbiór obiektów** stworzonych w oparciu o pewną definicję (wzorzec).

Klasa jako wzorzec dyktuje cechy i zachowania obiektom stworzonym na jej podstawie. Na rysunku 4.24 przedstawiono ją w postaci foremki, za pomocą której z ciasta wycina się ciasteczka (obiekty tej klasy).



Rysunek 4.24. Ciasteczka, klasy i obiekty



Rysunek 4.25. Klasa jako wzorzec

Klasa jako zbiór obiektów może być przedstawiona w formie tabeli w bazie danych, w której zapisane są wiersze, czyli obiekty tej klasy (o wspólnych cechach — kolumnach).

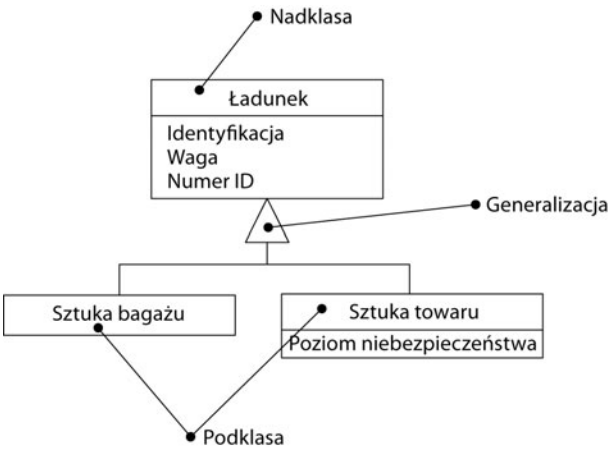
Klasy obok atrybutów z reguły zawierają metody, które determinują ich zachowanie. W naszym podejściu do modelowania systemów informatycznych z reguły unikamy wykorzystywania tej możliwości. Zachowanie obiektów jest uzależnione przede wszystkim od ich stanów. Metoda „odwołaj lot" klasy „lot" powinna wykonywać zupełnie inne działania, gdy jest wywoływana w stanie „w trakcie lotu", niż gdy jej stanem jest „w przygotowaniu".

Zgodnie z naszym doświadczeniem reguły tego typu mogą być modelowane o wiele prościej na diagramach stanów w perspektywie zachowań niż za pomocą metod klas. Dopiero na dalszych etapach projektowania i implementacji realizowane są szczegóły zachowania klas, co jest zachowywane w postaci ich metod.

W biurze handlowym kupca Hanzy klasą jest na przykład księga (indeks klientów) w połączeniu z opiekującym się nią księgowym. Obiektami tej klasy są poszczególne pozycje w księdze, czyli pojedynczy klienci. W księdze znajdują się również odniesienia do innych ksiąg — w postaci odwołań krzyżowych.

4.2.2. Generalizacja, specjalizacja i dziedziczenie

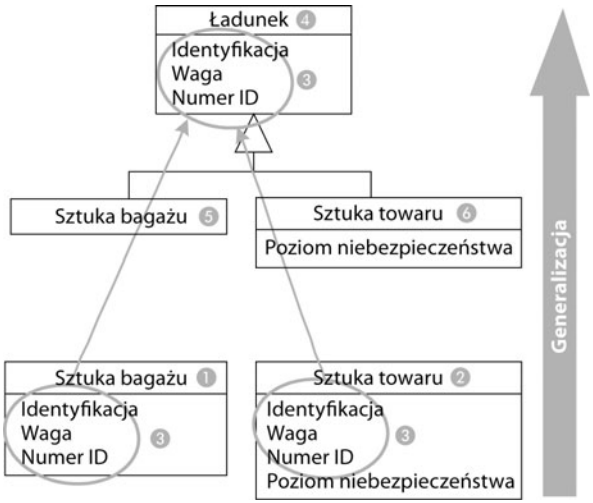
Przy rozważaniach na temat technik obiektowych do głowy przychodzą takie terminy, jak nadklasa, podklasa czy dziedziczenie. Te koncepcje są bardzo ważne podczas pracy z językami programowania zorientowanymi obiektowo, takimi jak Java, Smalltalk czy C++. W przypadku modelowania klas w celu zilustrowania koncepcji technicznych zagadnienia te mają drugorzędne znaczenie. Powodem takiego stanu rzeczy jest fakt, że modelowanie obiektów lub zjawisk świata rzeczywistego daje niewiele okazji do wykorzystania dziedziczenia (weźmy pod uwagę diagram klas w naszym studium przypadku). Niemniej jednak dla porządku należy omówić znaczenie tych terminów (rysunek 4.26).



Rysunek 4.26. Notacja generalizacji

Generalizacja jest procesem wydobywania współdzielonych cech z dwóch lub większej liczby klas i łączenia ich w uogólnioną nadklasę. Wspólnymi cechami mogą być atrybuty, asocjacje czy metody.

Na rysunku 4.27 widnieją klasy **Sztuka bagażu** (1) oraz **Sztuka towaru** (2). Mają one wspólne niektóre cechy. Z perspektywy dziedziny te dwie klasy są bardzo podobne. Gdyby wykonać ich generalizację, należałoby połączyć ich wspólne cechy (3) i stworzyć nową klasę **Ładunek** (4). Klasy **Sztuka bagażu** (5) i **Sztuka towaru** (6) stałyby się wówczas podklasami klasy **Ładunek**.



Rysunek 4.27. Przykład generalizacji

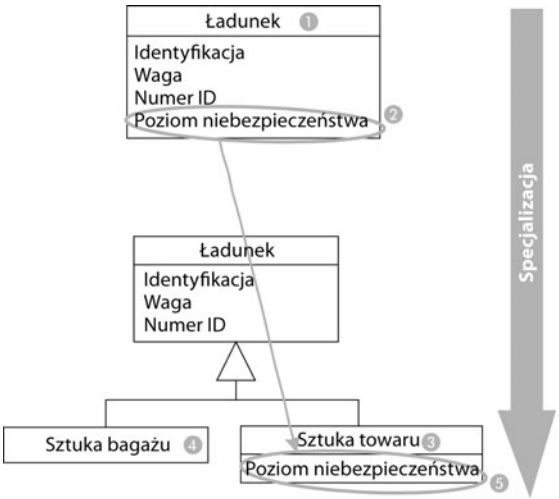
Współdzielone atrybuty (3) są zadeklarowane tylko w nadklasie, lecz występują również w podklasach, mimo że nie są w nich jawnie zadeklarowane.

> Ćwiczenie: zastanów się, czy któreś z klas zidentyfikowanych w poprzednich ćwiczeniach nadają się do generalizacji.

O odróżnieniu od generalizacji **specjalizacja** polega na tworzeniu nowych podklas z istniejących klas. Jeśli okaże się, że niektóre atrybuty, asocjacje i metody są wykorzystywane wyłącznie w pewnej grupie obiektów klasy, warto stworzyć odpowiednią podklasę. Najbardziej ogólna klasa w hierarchii generalizacji i specjalizacji nosi nazwę nadklasy i jest z reguły rysowana u góry diagramu hierarchii. Najbardziej specjalizowane klasy są podklasami i są rysowane u dołu diagramu, poniżej odpowiedniej nadklasy.

Na rysunku 4.28 klasa **Ładunek** (1) posiada atrybut **Poziom niebezpieczeństwa** (2), potrzebny tylko w przypadku towarów, a nie w przypadku bagaży pasażerów. Ponadto (nie widać tego na rysunku 4.28) tylko bagaż ma asocjację z biletem. Oczywiście, w przypadku istnienia jednej klasy te dwie zupełnie wzajemnie niezwiązane koncepcje byłyby połączone w jedną. Dzięki specjalizacji można stworzyć dwie podklasy do obsługi dwóch różnych przypadków specjalnych, czyli **Sztuki towaru** (3) oraz **Sztuki bagażu** (4). Atrybut **Poziom niebezpieczeństwa** (5) jest umieszczony tam, gdzie jego miejsce, w klasie **Sztuka towaru**. Atrybuty klasy **Ładunek** (1) znajdują się również w klasach **Sztuka towaru** (3) oraz **Sztuka bagażu** (4).

> Ćwiczenie: zastanów się, czy któreś z klas zidentyfikowanych w poprzednich ćwiczeniach nadają się do specjalizacji.



Rysunek 4.28. Przykład specjalizacji

Tak wygląda sam mechanizm. Jednak związek między nadklasą a podklasą ma o wiele szersze znaczenie, niż można się spodziewać na pierwszy rzut oka. Związek ten wymusza następujące reguły:

- Wszystkie założenia dotyczące nadklasy dotyczą również podklas. Podklasy **dziedziczą** atrybuty, asocjacje i operacje od swoich nadklas. Na przykład jeśli nadklasa **Ładunek** posiada atrybut **Waga**, podklasa **Sztuka bagażu** również posiada atrybut **Waga**, mimo że tego atrybutu nie wymienia się w specyfikacji klasy **Sztuka bagażu**.
- Wszystko, co można wykonać z obiektem nadklasy, można też wykonać z obiektem podklasy. Na przykład jeśli ładunek można załadować na pokład, można to samo zrobić ze sztuką bagażu.
- W terminologii modelowanego systemu podklasa musi być specjalną formą nadklasy. Sztuka bagażu jest na przykład specjalnym przypadkiem ładunku. Kontrprzykładem może być to, że lot nie jest specjalnym przypadkiem numeru lotu.

4.2.3. Statyczne i dynamiczne reguły biznesowe

Reguły biznesowe są regułami dziedziny opisanymi w ramach modelu systemu informatycznego. Reguły dziedziny można zdefiniować na podstawie strategii biznesowych, wymagań, zaleceń technicznych i ograniczeń. Reguły biznesowe nie są związane z technologią informatyczną, są definiowane jedynie na podstawie dziedziny. Przykłady reguł biznesowych:

- Podczas odprawy każdy pasażer musi mieć przyporządkowane miejsce na pokładzie samolotu.
- Dla każdego lotu każde miejsce na pokładzie może być przypisane tylko jednemu pasażerowi.
- Lot nie może zostać odwołany po rozpoczęciu.

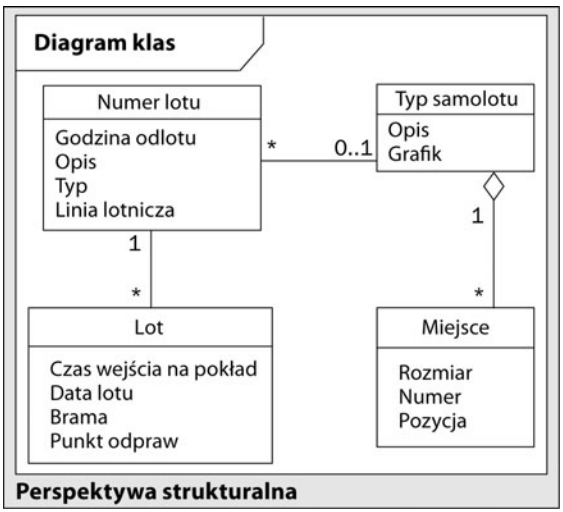
Wielu wymogów nie daje się modelować w postaci reguł biznesowych. Oprócz modelu informatycznego częścią specyfikacji systemu informatycznego powinien być katalog wymagań. W tej książce nie będziemy jednak zajmować się zagadnieniami tego typu.

Reguły biznesowe można podzielić na dwie kategorie:

- **Statyczne reguły biznesowe** — można je zweryfikować w dowolnym momencie. Te reguły dotyczą statycznych struktur klas. Można je przedstawić w diagramach klas perspektywy strukturalnej.
- **Dynamiczne reguły biznesowe** — reguły, które można zweryfikować jedynie w określonym momencie, to znaczy wtedy, gdy wystąpi określone zdarzenie. Tego typu reguły dotyczą dynamicznych zachowań obiektów klas. Można je dokumentować w ramach diagramów stanów perspektywy zachowań.

4.2.4. Elementy perspektywy

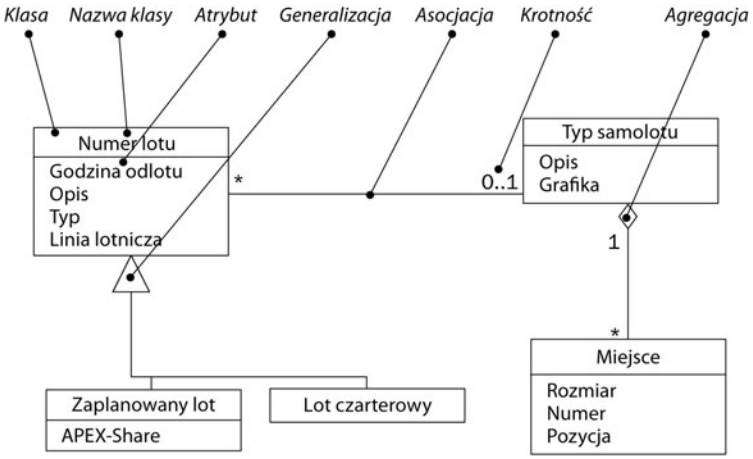
Perspektywa strukturalna systemu informatycznego — przedstawiona na rysunku 4.29 — składa się z jednego lub kilku diagramów klas. Diagramy klas opisują wszystkie klasy systemu informatycznego, ich wzajemne powiązania (asocjacje), charakterystyki (atrybuty) oraz, w uproszczeniu, ich zachowanie (metody). To jest najbardziej znana perspektywa metodologii zorientowanej obiektowo i — niestety — często jedyny diagram tworzony na etapie projektowym. Diagramy klas przedstawiają wewnętrzne statyczne struktury systemu informatycznego.



Rysunek 4.29. Perspektywa strukturalna

4.2.5. Diagram klas

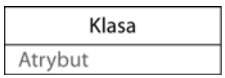
W diagramie klas przedstawionym na rysunku 4.30 mamy do dyspozycji następujące elementy:



Rysunek 4.30. Elementy diagramu klas

Klasa

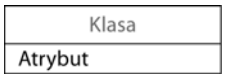
Klasa reprezentuje odpowiednią koncepcję dziedziny, taką jak zbiór osób, obiektów lub zjawisk przedstawianych w systemie informatycznym.



Przykładami klas są pasażerowie, samoloty i bilety.

Atrybut

Atrybut klasy reprezentuje taką jej cechę, która jest interesująca z punktu widzenia użytkownika systemu informatycznego.



Interesującą cechą charakterystyczną pasażera może być na przykład jego nazwisko i imię oraz wiek.

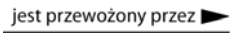
Generalizacja

Generalizacja jest związkiem między dwoma klasami — ogólną i specjalną:

Więcej informacji na ten temat można znaleźć w punkcie 4.2.2, „Generalizacja, specjalizacja i dziedziczenie".

Asocjacja

Asocjacja reprezentuje powiązanie między dwoma klasami.



Asocjacja informuje o tym, że obiekt jednej klasy posiada związek z obiektami innej klasy, przy czym to połączenie ma określone, zdefiniowane z góry znaczenie (na przykład „jest przewożony przez").

Krotność

Krotność służy przekazaniu informacji o liczbie obiektów zaangażowanych w asocjację.

Zobacz również rysunek 4.32.

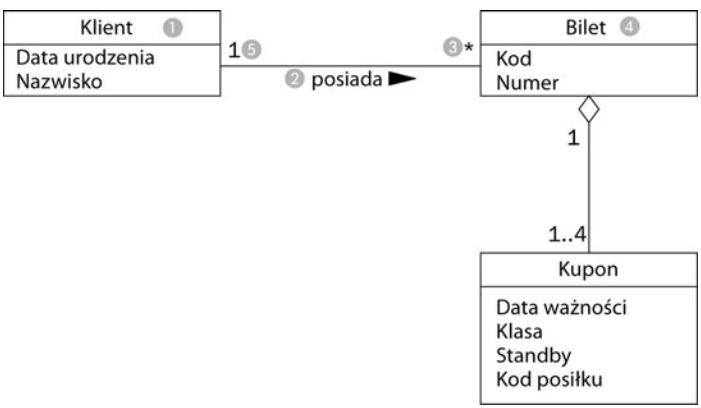
Agregacja

Agregacja jest specjalnym przypadkiem asocjacji (zobacz wyżej) o znaczeniu „składa się z":

Znaczenie agregacji jest sygnalizowane symbolem rombu, opis jest zbędny.

Czytanie diagramów klas

Rysunek 4.31 przedstawia diagram klas z naszego studium przypadku. Mamy tu klasy **Klient**, **Bilet** i **Kupon**, ich atrybuty i asocjacje:



Rysunek 4.31. Diagram klas z asocjacjami

Patrząc na diagram klas z rysunku 4.31, można zobaczyć asocjacje między klasami **Klient** i **Bilet**.

- Jeden obiekt pierwszej klasy posiada asocjację z większą liczbą obiektów drugiej klasy.

W powyższym zdaniu (stanowiącym wzorzec) należy wstawić odpowiednie nazwy z diagramu. Nazwą pierwszej klasy jest **Klient** (1), nazwą drugiej klasy jest **Bilet** (4). Nazwa asocjacji to **posiada** (2).

- Klient (1) posiada (2) * (3) biletów (4).

Jeśli symbol gwiazdki zastąpimy jej znaczeniem, otrzymamy prawidłowe zdanie:

- Klient (1) posiada (2) zero, jeden lub kilka (3) biletów (4).

Asocjacje z reguły nie są zwrotne, co znaczy, że obowiązują w obydwu kierunkach. Naszą asocjację można zatem odczytać również w następujący sposób:

- Bilet (4) jest w posiadaniu (2) dokładnie jednego (5) klienta (1).

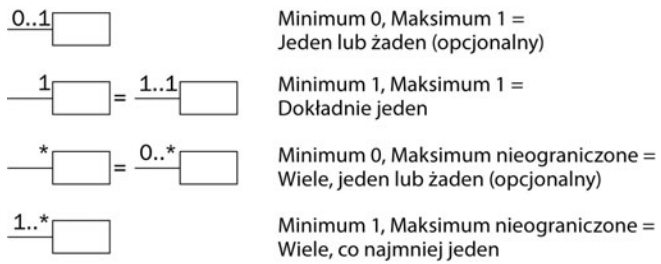
Przy nazwie asocjacji występuje trójkąt (2), który informuje o „kierunku działania" tej nazwy.

Wszystkie asocjacje z diagramu klas można odczytywać w opisany wyżej sposób.

Specyfikacja liczby obiektów drugiej klasy (asocjację zawsze odczytuje się, przyjmując jedną z klas za wyjściową) jest nazywana krotnością. Krotności zawsze zapisuje się zgodnie z poniżej podaną regułą.

Najpierw ograniczenie dolne, następnie dwukropek, potem ograniczenie górne.

Rysunek 4.32 przedstawia najczęściej stosowane kombinacje krotności.



Rysunek 4.32. Krotności

W UML-u wolno również stosować specjalne wartości dolnego i górnego ograniczenia, na przykład 2..4, ale też 6..*.

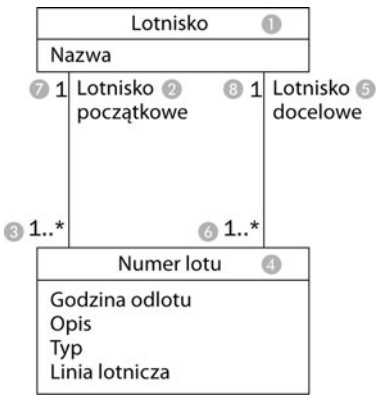
Nazwa asocjacji jest ważna, ponieważ określa jej znaczenie w ramach dziedziny. W przeciwieństwie do samej asocjacji, która działa w obydwu kierunkach, nazwa asocjacji działa tylko w jednym kierunku sygnalizowanym czarnym trójkątem. Jeśli asocjacja nie ma etykiety, jej

znaczenie trzeba wyłuskać z kontekstu lub przyjąć ogólne znaczenie, takie jak „posiada", lub „należy do". W przypadku wątpliwości lepiej nadać asocjacjom zbyt wiele etykiet, niż ryzykować problemy ze zrozumieniem diagramów. Z własnego doświadczenia wiemy, że wiele diagramów bywa niezrozumiałych wyłącznie z powodu braku odpowiednich, czytelnych etykiet.

Asocjacje można również postrzegać jako implementacje statycznych reguł biznesowych (zobacz punkt 4.2.3, „Statyczne i dynamiczne reguły biznesowe"). Określenia typu „bilet należy do dokładnie jednego klienta" są udokumentowane właśnie w formie asocjacji w ramach diagramu klas.

Innym możliwym sposobem powiązania klas z ich znaczeniem w dziedzinie są w UML-u **role**. Dzięki nim można określić, jaką rolę obiekt jednej klasy odgrywa dla obiektu innej klasy.

Patrząc na diagram klas z rysunku 4.33, możemy odczytać następującą asocjację ze zdefiniowaną rolą pomiędzy numerem lotu a lotniskiem:



Rysunek 4.33. Diagram klas z określeniem ról

- **Lotnisko** (1) jest **lokalizacją początkową** (2) dla **jednego** lub **większej liczby** (3) **numerów lotu** (4).

Na tym samym rysunku mamy jeszcze jedną asocjację między klasami **numer lotu** i **lotnisko**:

- **Lotnisko** (1) jest **lokalizacją docelową** (5) dla **więcej niż jednego** (3) **numeru lotu** (4).

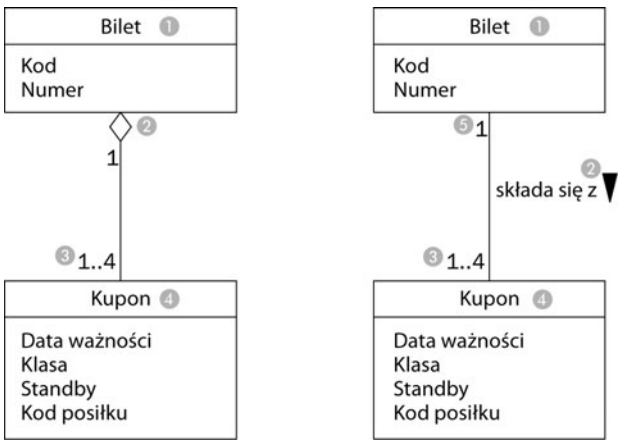
Obie te asocjacje można również odczytać w przeciwnych kierunkach, choć role określa się tylko w jednym:

- **Numer lotu** (4) ma **lokalizację startową** (2) na **dokładnie jednym** (7) **lotnisku** (1).
- **Numer lotu** (4) ma **lokalizację docelową** (5) **dokładnie na jednym** (7) **lotnisku** (1).

Ta asocjacja informuje, że dany numer lotu ma zdefiniowane lotniska startu i lądowania. Przykładem numeru lotu może być LX317, lot samolotu należącego do szwajcarskich linii lotniczych Crossair, kursującego na trasie z Londynu do Zurychu.

Wśród wielu znaczeń, jakie mogą mieć asocjacje w dziedzinie, jest jedno na tyle ważne, że w UML-u otrzymało własny symbol: jest to agregacja całość-część. Ten typ związku jest wykorzystywany zawsze, gdy obiekty jednej klasy wchodzą w skład (są częściami składowymi) innej klasy.

Na diagramie klasy z rysunku 4.34 agregacja występuje po lewej (biały romb). Można ją odczytać w następujący sposób:



Rysunek 4.34. Diagram klas z agregacją

- Bilet (1) składa się z (2) od 1 do 4 (4) kuponów.

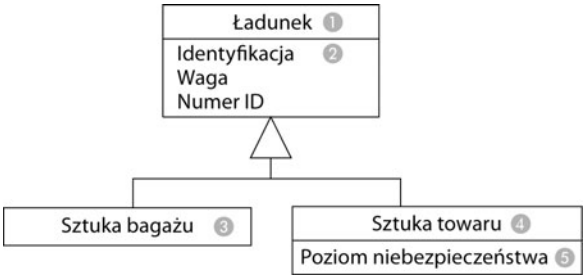
I w drugą stronę:

- Kupon (4) wchodzi w skład (2) dokładnie jednego (5) biletu (1).

Drugi przykład (bez białego rombu, lecz z nazwą asocjacji) ma dokładnie takie samo znaczenie.

Ostatnie elementy UML-a wykorzystywane przez nas w modelowaniu to generalizacja i specjalizacja, które służą zobrazowaniu związków między nadklasą a podklasą. Generalizacja i specjalizacja z rysunku 4.35 mogą być odczytywane od góry do dołu lub od dołu do góry. Jeśli zaczniemy od góry, znajdziemy klasę **Ładunek** (1) z atrybutami: **Identyfikacja**, **Waga**, **Numer ID** (2). Ta klasa ma dwie specjalizacje: **Sztuka bagażu** (3) i **Sztuka towaru** (4). Klasa **Sztuka towaru** posiada dodatkowy atrybut: **Stopień niebezpieczeństwa** (5).

Jeśli będziemy czytali od dołu, znajdziemy klasy **Sztuka bagażu** (3) i **Sztuka towaru** (4). Nadklasą dla obu jest **Ładunek** (1) — zawiera ona wspólne dla nich wszystkich atrybuty i metody.



Rysunek 4.35. Diagram klas z generalizacją i specjalizacją

4.2.6. Konstruowanie diagramów klas

Podstawowy problem przy konstruowaniu diagramów klas to zidentyfikowanie odpowiednich klas. Ten problem będziemy analizować z dwóch punktów widzenia i dlatego diagram klas będziemy budować w dwóch etapach.

W analizie zstępującej (od ogółu do szczegółu) klasy są identyfikowane na podstawie ogólnego zrozumienia modelowanych zagadnień. Analiza tego typu polega na znalezieniu podstawowych zarysów struktur, które następnie są rozbudowywane na etapie analizy wstępującej. Analiza zstępująca wymaga (w uproszczeniu) odpowiedzi na pytanie:

- Jakie informacje lub koncepcje domenowe mogą być wykorzystywane przez tworzony system informatyczny?

Ważnymi źródłami informacji na tym etapie są wiedza dotycząca danej dziedziny (na przykład słowne opisy obszaru działania aplikacji) oraz przedstawiciele grupy użytkowników. Dzięki tym źródłom informacji w większości systemów informatycznych można znaleźć podstawowe struktury klas.

W analizie wstępującej (od szczegółu do ogółu) klasy są identyfikowane głównie w oparciu o wejściowe i wyjściowe dane systemu informatycznego. Na tym etapie pomocne są odpowiedzi na pytanie:

- Jaka informacja jest potrzebna w charakterze wejścia i wyjścia systemu informatycznego?

Podstawą tej analizy są klasy zidentyfikowane na etapie analizy zstępującej. Ważnymi źródłami informacji są tu istniejące wejścia i wyjścia, takie jak formularze ekranowe i wydruki.

Zgodnie z naszym doświadczeniem te dwa etapy razem prowadzą do osiągania zadowalających wyników w modelowaniu systemów informatycznych obsługujących duże ilości informacji. Istnieją jednak systemy, które mają za zadanie zarządzanie lub kontrolę jakiegoś procesu. Cechują się one z reguły skomplikowanymi mechanizmami wewnętrznymi, lecz nie obsługują żadnych (albo bardzo niewiele) danych. W takich systemach zaleca się podejście mniej skupione na danych, na przykład projektowanie oparte na odpowiedzialnościach. Podejście to zostało opisane w książce Rebeki Wirfs-Brock, Briana Wilkersona i Lauren Wiener, *Designing Object-Oriented Software*[1].

[1] R. Wirfs-Brock, B. Wilkerson, L. Wiener, *Designing Object-Oriented Software*, Prentice Hall, 1998.

Lista kontrolna 4.4 przedstawia etapy niezbędne podczas konstruowania diagramów kas. Każdy z tych etapów omówimy w kolejnych punktach rozdziału.

Lista kontrolna 4.4. Konstruowanie diagramów klas w perspektywie strukturalnej

Analiza zstępująca:

- Identyfikacja i modelowanie klas — które z nich są potrzebne?
- Identyfikacja i modelowanie asocjacji — w jaki sposób klasy są powiązane wzajemnie?
- Definiowanie atrybutów — co chcemy wiedzieć o obiektach?

Analiza wstępująca:

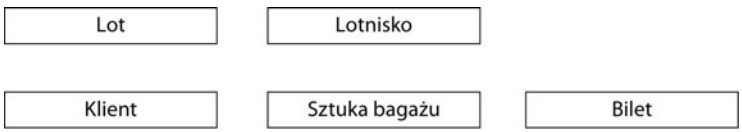
- Identyfikacja zapytań i wyjść — jakich danych powinien dostarczać i jakie dane przyjmować system informatyczny?
- Formułowanie zapytań i wejść — w jaki sposób ma wyglądać interfejs użytkownika?
- Przeprowadzanie analizy informacji — jakich potrzebujemy klas, asocjacji i atrybutów?
- Konsolidowanie diagramów klas — na ile pasują do siebie poszczególne części?
- Weryfikacja diagramów klas — czy wszystko zostało wykonane prawidłowo?

Identyfikacja i modelowanie klas — które z nich są potrzebne?

Analiza wzajemnych powiązań, potrzeb związanych z informacją oraz aktorów i prototypów jest przeprowadzana w oparciu o ogólną wiedzę z zakresu danej dziedziny, dyskusje z ekspertami i dokumentację. W tym miejscu należy zadać sobie następujące pytania:

- Jakie zagadnienia będą poddawane analizie w systemie informatycznym?
- Jakie klasy można zdefiniować w oparciu o odpowiedź na pierwsze pytanie?

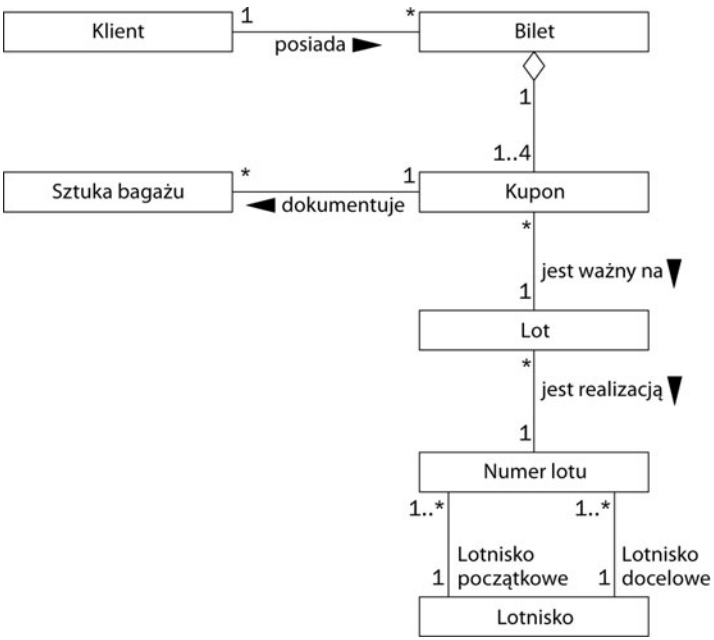
Odpowiedzi na te pytania dostarczają listę potencjalnych klas, które należy zamodelować w pierwszym szkicu diagramu klas. W praktyce wyniki uzyskane na tym etapie prac mogą być znacznie zróżnicowane. Jednakże nie zdarzyło nam się nigdy, żeby nie udało się w tym miejscu zidentyfikować zupełnie żadnych klas. Jeśli analityk nie ma większego doświadczenia w identyfikowaniu klas, czasem nieuniknione okazuje się wielokrotne przeprowadzanie analizy zstępującej. Z czasem analitykowi uda się nabyć umiejętność wykrywania klas i unikania elementów, które nimi nie są (rysunek 4.36).



Rysunek 4.36. Potencjalne klasy

Identyfikacja i modelowanie asocjacji — w jaki sposób klasy są wzajemnie powiązane?

Powiązania między klasami uzyskanymi na poprzednim etapie oraz reguły biznesowe modeluje się w postaci asocjacji, którym należy nadać czytelne nazwy oraz zdefiniować ich krotności, co przedstawia rysunek 4.37. Pytania przydatne na tym etapie:



Rysunek 4.37. Klasa i jej asocjacje

- Jakie powiązania istnieją między obiektami?
- Jak wiele obiektów każdej z klas jest zaangażowanych w powiązanie?

Pierwsze z tych pytań należy zadać w odniesieniu do par obiektów każdej z klas, na przykład dla pary klas **Lot** i **Pasażer**. W tym miejscu duże znaczenie ma to, czy powiązanie jest bezpośrednie, czy też można określić je jedynie za pośrednictwem innych obiektów. W naszym przykładzie okazuje się, że klient jest właścicielem biletu, który z kolei składa się z kuponów obowiązujących jedynie na określony lot. Celem drugiego pytania jest określenie krotności związku, na przykład określenie liczby biletów, które może posiadać klient, oraz kwestia, do ilu klientów należy jeden bilet (rysunek 4.37).

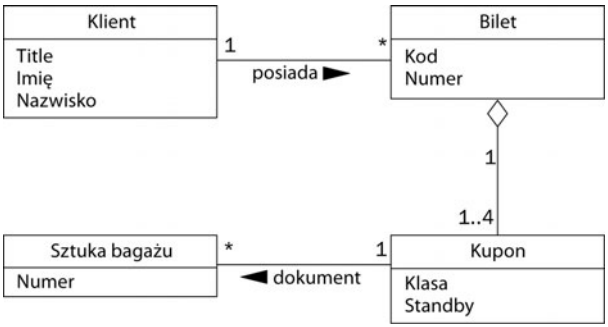
Pomimo że na początku tego etapu zaczęliśmy od klas znalezionych w poprzednim, dzięki dyskusjom może okazać się, że nagle klas zrobi się więcej.

Definiowanie atrybutów — co chcemy wiedzieć o obiektach?

Informacje na temat klas modeluje się w postaci atrybutów. Na tym etapie należy odpowiedzieć na pytanie:

- Jakie informacje chcemy zdobyć na temat klas?

Należy skupić się na rzeczywiście najbardziej kluczowych atrybutach każdej z klas (rysunek 4.38). Na to pytanie nie uda się odpowiedzieć wyczerpująco, jeśli nie zostaną precyzyjnie przeanalizowane dane wejściowe i zapytania, co dzieje się podczas analizy wstępującej. Z tego powodu nie należy poświęcać temu pytaniu zbyt dużej ilości czasu.



Rysunek 4.38. Klasy i atrybuty

Identyfikacja zapytań i wyjść — jakich danych powinien dostarczać i jakie dane przyjmować system informatyczny?

Na pierwszym etapie analizy wstępującej należy zidentyfikować indywidualne zapytania i wejścia systemu informatycznego. Zapytania są tutaj ważniejsze, ponieważ właśnie udzielanie odpowiedzi na nie jest w gruncie rzeczy główną funkcją systemu informatycznego. Pomocne pytania:

- Jakich informacji ma dostarczać system informatyczny?
- Jakie informacje ma przyjmować system informatyczny?

Odpowiadając na te pytania, można posiłkować się zidentyfikowanymi do tej pory przypadkami użycia. Zapytania i modyfikacje zostały już naszkicowane w diagramie sekwencji. Innym dobrym źródłem informacji są procesy biznesowe odzwierciedlone w ramach modelu systemu biznesowego (rozdział 3.). Rezultatem tego etapu jest lista wymagań w stosunku do informacji. Przykład takiej listy przedstawia rysunek 4.39.

Wymagania	Typ	Przypadek użycia
Karta pokładowa	Wyjście	Wystawienie karty pokładowej
Lista pasażerów	Wyjście	Wsiadanie na pokład
Szczegóły biletu	Wyjście	Odprawa, odprawa ekspresowa
Szczegóły biletu	Wyjście	Odprawa, odprawa ekspresowa
Kupon	Wejście	Odprawa, odprawa ekspresowa
...		

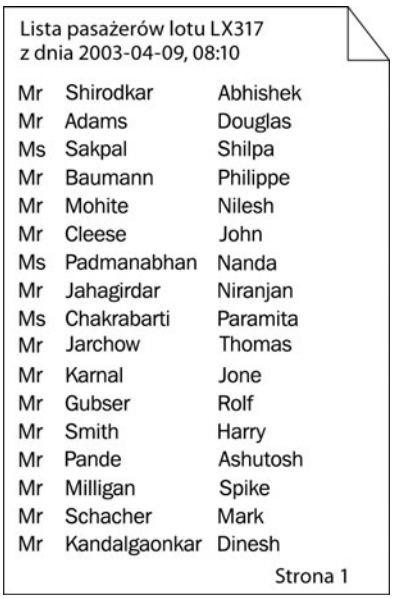
Rysunek 4.39. Lista wymagań w stosunku do informacji

Formułowanie zapytań i wejść — w jaki sposób ma wyglądać interfejs użytkownika?

Aby stworzyć diagramy klas poszczególnych zapytań i wejść, najpierw musimy określić ich wygląd. Na tym etapie gromadzi się projekty formularzy skomplikowanych zapytań lub danych wejściowych. Rysunek 4.40 przedstawia listę pasażerów, rysunki 4.66 (interfejs użytkownika) oraz 4.67 (karta pokładowa) przedstawiają inne przykłady elementów interfejsu. Pytanie pomocne na tym etapie:

- Jak dokładnie wygląda formularz zapytania lub dane wejściowe?

Dobrymi źródłami informacji są istniejące już formularze (na przykład lista pasażerów z rysunku 4.40) oraz prototypy interfejsu użytkownika:


Lista pasażerów lotu LX317
z dnia 2003-04-09, 08:10

Mr	Shirodkar	Abhishek
Mr	Adams	Douglas
Ms	Sakpal	Shilpa
Mr	Baumann	Philippe
Mr	Mohite	Nilesh
Mr	Cleese	John
Ms	Padmanabhan	Nanda
Mr	Jahagirdar	Niranjan
Ms	Chakrabarti	Paramita
Mr	Jarchow	Thomas
Mr	Karnal	Jone
Mr	Gubser	Rolf
Mr	Smith	Harry
Mr	Pande	Ashutosh
Mr	Milligan	Spike
Mr	Schacher	Mark
Mr	Kandalgaonkar	Dinesh

Strona 1

Rysunek 4.40. Lista pasażerów

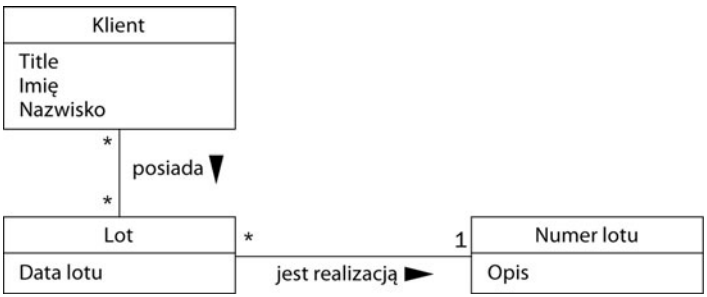
Przeprowadzanie analizy informacji — jakich potrzebujemy klas, asocjacji i atrybutów?

Na tym etapie prac większa część analizy wstępującej jest już gotowa. Dla każdego zapytania lub wejścia stworzony jest diagram klasy. Do tego zadania wykorzystane zostały wstępne projekty wejścia i wyjścia systemu informatycznego. Można to nazwać modelowaniem klas na małą skalę. Pytania przydatne na tym etapie:

- Jakie istnieją elementy danych na wejściu i wyjściu?
- Jakie obiekty skrywają się za tymi elementami danych?

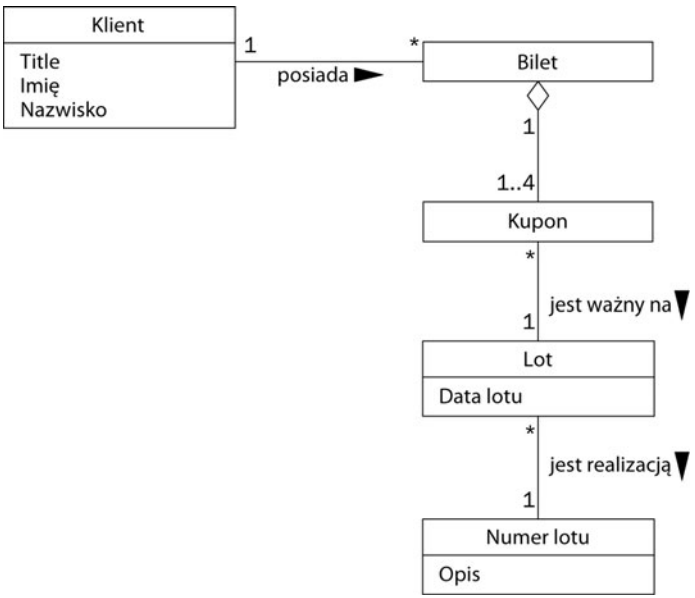
- Jakie związki istnieją między znalezionymi obiektami?
- Które z zamodelowanych dotychczas obiektów mogą być wykorzystane?

W przypadku listy pasażerów z rysunku 4.40 można skonstruować diagram klas przedstawiony na rysunku 4.41.



Rysunek 4.41. Diagram klas dla listy pasażerów

Jeśli weźmiemy dodatkowo pod uwagę klasy znalezione już w ramach analizy zstępującej, możemy stworzyć diagram przedstawiony na rysunku 4.42.



Rysunek 4.42. Zmodyfikowany diagram klas dla listy pasażerów

Konsolidowanie diagramów klas — na ile pasują do siebie poszczególne części?

Na tym ostatnim etapie należy skumulować w jednym diagramie wszystkie pojedyncze diagramy klas. Powinniśmy mieć już wykryte i poprawione wszystkie pozostałe niespójności. Przydatne pytania:

- Czy w poszczególnych diagramach można znaleźć klasy o różnych nazwach reprezentujące ten sam obiekt?
- Czy w poszczególnych diagramach klas istnieją różne powiązania, które mają to samo znaczenie?
- Czy w poszczególnych diagramach klas istnieją różne atrybuty, które mają to samo znaczenie?

Podczas konsolidowania diagramów klas w jeden, ogólny diagram powyższe pytania właściwie narzucają się same. Po zidentyfikowaniu niespójności z reguły poprawienie ich jest już łatwym zadaniem. Jeśli do modelowania klas w oparciu o zapytania i wejścia zostały wykorzystane klasy zidentyfikowane podczas analizy zstępującej, mogą pojawić się konflikty i powtórzenia w konfrontacji z diagramami klas znalezionymi na pozostałych etapach. Również te nadmiarowości należy usunąć.

Weryfikacja diagramów klas — czy wszystko zostało wykonane prawidłowo?

Gotowy diagram klas w perspektywie strukturalnej można zweryfikować za pomocą następującej listy kontrolnej:

Lista kontrolna 4.5. Weryfikacja diagramów klas w perspektywie strukturalnej

- Czy diagram klas jest kompletny?

  Na to pytanie będziemy w stanie odpowiedzieć dopiero w podrozdziale 4.4, „Perspektywa interakcji". Dopiero bowiem wówczas będzie widać, czy diagram klas odpowiada na wszystkie zapytania systemu informatycznego. Jeśli ten wymóg jest spełniony, diagram klas można uznać za kompletny.

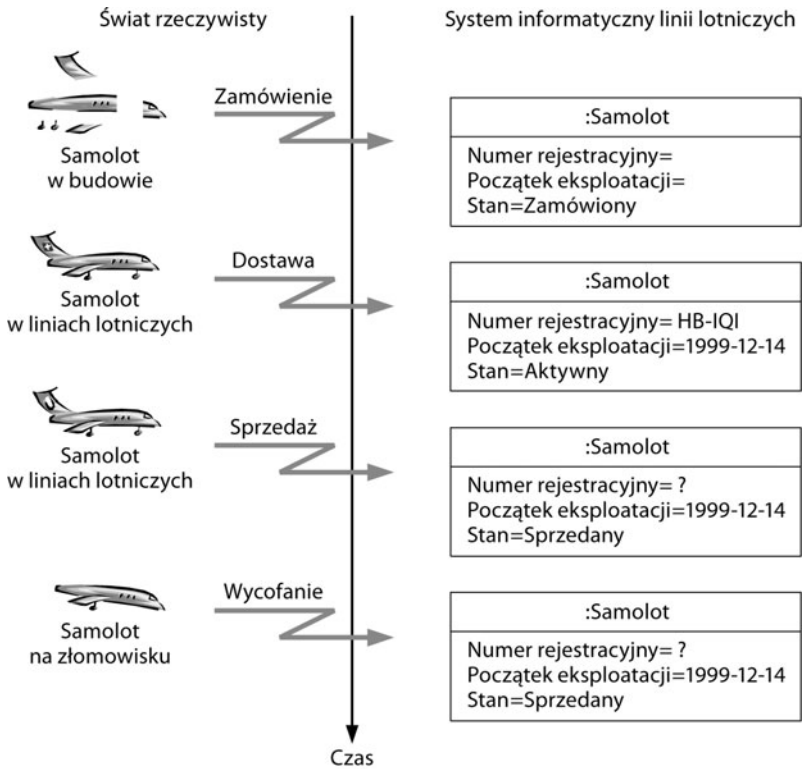
- Czy diagram klas jest poprawny?

  Na to drugie pytanie o wiele trudniej znaleźć odpowiedź. Z naszego doświadczenia wynika, że najlepszym sposobem jej odszukania jest wspólna analiza diagramu z dostawcami wiedzy. To pozwala odkryć większość błędów. Oprócz tego diagram klas można testować pod kątem podejrzanych wzorców strukturalnych. Najlepiej posłużyć się tu odpowiednim narzędziem. Wprowadzenie do analizy wzorców strukturalnych pozostaje jednak poza obszarem tematycznym tej książki. Wyjaśnienie tej koncepcji oraz narzędzie pomocne w analizie wzorców strukturalnych można znaleźć pod adresem *http://www.knowgravity.com/eng/value/cassandra.htm*.

4.3. Perspektywa zachowań

4.3.1. Cykl życia obiektu

Osoby, obiekty i zjawiska świata rzeczywistego, które są modelowane na potrzeby systemu informatycznego, cechują się określonym cyklem życia. W rzeczywistości można zaryzykować stwierdzenie, że mają one dwa życia: życie w świecie rzeczywistym oraz życie obiektu zobrazowane w modelu. Choć te dwa życia są powiązane, nie muszą koniecznie pokrywać się ze sobą. Z reguły życie rozpoczyna się od narodzin (czy też momentu utworzenia), a kończy się śmiercią lub zniszczeniem. Pomiędzy tymi dwoma punktami życie toczy się dość jednolitym nurtem, którego przykład przedstawia rysunek 4.43.



Rysunek 4.43. Cykl życia samolotu

Zastanówmy się chwilę nad cyklem życia samolotu. Weźmy pod uwagę samolot Airbus A330-223 linii lotniczych Swiss International Airlines, o numerze rejestracyjnym HB-1QI.

- Życie samolotu Airbus A330-223 (rzeczywiste) rozpoczyna się — w zależności od punktu widzenia — od początku procesu konstrukcyjnego tego samolotu;
- Życie obiektu Airbus A330-223 w systemie informatycznym rozpoczyna się wraz z wprowadzeniem do systemu informacji o tym samolocie. Może to wystąpić z chwilą złożenia zamówienia, ponieważ samolot mógł zostać zarejestrowany w systemie informatycznym w celu zaplanowania inwestycji. Początkiem życia tego obiektu może równie dobrze być moment dostarczenia samolotu. Inicjatorem narodzin obiektu w systemie informatycznym zawsze jest zdarzenie modyfikujące.

Ponieważ w rzeczywistości komercyjne samoloty są sprzedawane na długo przed rozpoczęciem ich konstruowania, bardzo możliwe jest, że narodziny obiektu następują znacznie wcześniej niż w momencie uznawanym za początek jego fizycznego istnienia.

- Rzeczywista śmierć następuje wraz z fizycznym zniszczeniem obiektu. W przypadku samolotu Airbus A330-233 wiąże się ona z wycofaniem go z użytku lub z ewentualną katastrofą.
- Śmierć obiektu w systemie informatycznym następuje wraz z usunięciem go z systemu, w tym przypadku z systemu linii lotniczych Swiss Airlines. Inicjatorem śmierci obiektu w systemie informatycznym zawsze jest zdarzenie modyfikujące.

Ponieważ w rzeczywistości często zdarza się, że samoloty przed całkowitym wycofaniem z użytkowania są odsprzedawane innym użytkownikom, możliwa jest logiczna śmierć obiektu w systemie informatycznym na długo przed fizyczną śmiercią w świecie rzeczywistym.

Pomiędzy narodzinami a śmiercią obiekty „żyją" w systemie informatycznym. Oznacza to, że obiekt taki może być odczytywany i modyfikowany. Jest on odczytywany w wyniku zdarzenia wydobywającego (zapytania), a modyfikowany w wyniku zdarzenia modyfikującego (mutatora). Więcej na ten temat można znaleźć w punkcie 4.1.4, „Zdarzenia wydobywające i modyfikujące dane".

Dopóki odczytywanie i modyfikowanie obiektów nie podlega ograniczeniom, nie są to operacje szczególnie interesujące z punktu widzenia modelowania. Można je wtedy opisać w postaci prostego diagramu stanów (rysunek 4.44). Jeśli jednak modyfikacje podlegają określonym regułom, istotne staje się jakieś udokumentowanie tych reguł. Mamy tu na myśli dynamiczne reguły biznesowe (zobacz punkt 4.2.3, „Statyczne i dynamiczne reguły biznesowe"). Dynamiczne reguły biznesowe to takie, które mają zastosowanie wyłącznie w określonym momencie — to znaczy wówczas, kiedy zachodzi zdarzenie wydobywające lub modyfikujące dane. Zachowanie obiektów jest zdeterminowane głównie przez tego typu reguły.

Oto przykłady dynamicznych reguł biznesowych:

- Samolot nie może mieć przypisanego lotu, jeśli jest w trakcie prac serwisowych.
- Samolot nie może być wycofany z użytku, dopóki funkcjonuje w grafiku lotów.

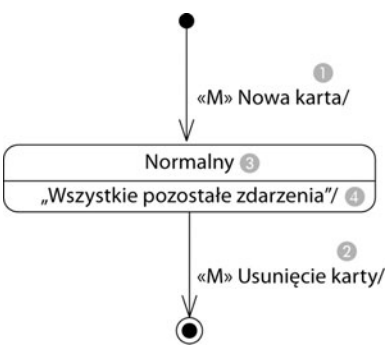
Jeśli przyjrzeć się bliżej regułom biznesowym, można zauważyć, że odnoszą się one do określonych zdarzeń oraz do stanów obiektów.

- Zdarzenie modyfikujące **przydzielenie do lotów** nie jest dozwolone w stanie **w serwisie** obiektu **samolot**.
- Zdarzenie modyfikujące **wycofanie z użytku** nie jest dozwolone w stanie **przydzielone loty** obiektu **samolot**.
- Zdarzenie modyfikujące **start** nie jest dozwolone w stanie **w trakcie przeniesienia** obiektu **samolot**.

Innymi słowy, w przypadku obiektów powinna być możliwość określenia stanów, w których nie mogą one przyjmować pewnych zdarzeń, oraz zaprogramowania sposobów, w jakie powinny one reagować na zdarzenia dozwolone w określonych stanach.

> Ćwiczenie: zastanów się, jakie dynamiczne reguły biznesowe obwiązują dla obiektu **bilet lotniczy** w systemie obsługi pasażerów.

W perspektywie zachowań przygotowuje się po jednym diagramie stanów dla każdej z klas. Te diagramy dokumentują reguły biznesowe, które muszą być przestrzegane, oraz zdarzenia dozwolone w każdym ze stanów obiektu. W najprostszym przypadku dozwolone są wszystkie zdarzenia. Rysunek 4.44 przedstawia prosty diagram stanów dla klasy **karta stałego klienta**.



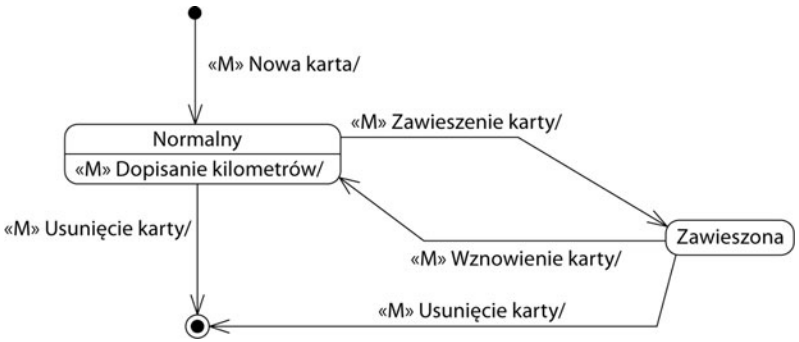
Rysunek 4.44. Prosty diagram stanów obiektu „karta stałego klienta"

W wyniku zdarzenia **«M» nowa karta** (1) generowany jest nowy obiekt. Obiekt jest usuwany w wyniku zdarzenia **«M» usunięcie karty** (2). Pomiędzy tymi zdarzeniami obiekt znajduje się w stanie **normalny** (3), w którym dozwolone są **wszystkie inne zdarzenia** (4) (w prawdziwym diagramie stanów zamiast określenia „wszystkie inne zdarzenia" należy umieścić listę dozwolonych zdarzeń).

Jeśli jednak dodamy regułę biznesową, diagram stanów stanie się bardziej skomplikowany. W celu zademonstrowania tego w praktyce uzupełnijmy nasz diagram stanów o następujące reguły:

- Musi istnieć możliwość wstrzymania ważności i wznowienia karty stałego klienta.
- Nie ma możliwości dopisania kilometrów do wstrzymanej karty stałego klienta.

Jeśli diagram stanów uzupełnimy o powyższe reguły biznesowe, otrzymamy wersję przedstawioną na rysunku 4.45.



Rysunek 4.45. Bardziej skomplikowana wersja diagramu stanów obiektu „karta stałego klienta"

Diagram stanów przedstawiony na rysunku 4.45 dokumentuje ścieżki lub granice życia obiektu **karta stałego klienta**. Z diagramu można wyczytać dopuszczalne i niedopuszczalne łańcuchy zdarzeń. Jednym z możliwych przepływów jest na przykład **«M» nowa karta**, **«M» dopisanie kilometrów**, **«M» dopisanie kilometrów**, **«M» dopisanie kilometrów**, **«M» wstrzymanie karty**, **«M» usunięcie karty**.

Przykładem niedopuszczalnej sekwencji może być **«M» nowa karta**, **«M» dopisanie kilometrów**, **«M» dopisanie kilometrów**, **«M» wstrzymanie karty**, **«M» dopisanie kilometrów**, **«M» usunięcie karty**. Przedostatnie zdarzenie **«M» dopisanie kilometrów** jest niedopuszczalne. Gdyby przed tym zdarzeniem wystąpiło zdarzenie **«M» wznowienie karty**, przepływ ponownie byłby dopuszczalny.

Zastanówmy się, czy następujące łańcuchy są dopuszczalne zgodnie z diagramem stanów przedstawionym na rysunku 4.45:

- **«M» dopisanie kilometrów, «M» dopisanie kilometrów, «M» dopisanie kilometrów, «M» usunięcie karty, «M» usunięcie karty**.
- **«M» nowa karta, «M» dopisanie kilometrów, «M» zawieszenie karty, «M» wznowienie karty, «M» dopisanie kilometrów, «M» usunięcie karty**.
- **«M» nowa karta, «M» dopisanie kilometrów, «M» zawieszenie karty, «M» zawieszenie karty, «M» usunięcie karty**.

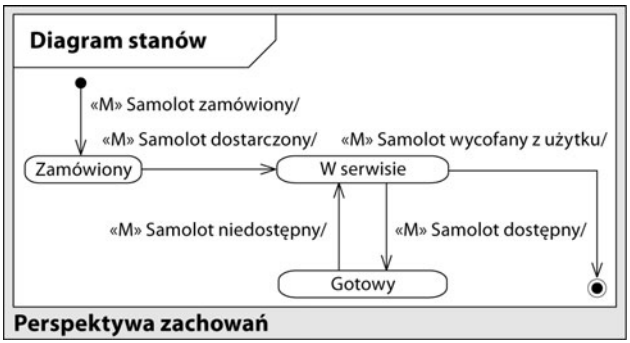
Życie obiektu toczy się właśnie tego typu z góry ustanowionych kursami, co oznacza, że obiekt musi działać zgodnie z określonymi regułami. Z tego powodu perspektywa zachowań ma szczególne znaczenie, ponieważ rolą systemu informatycznego jest zapewnienie przestrzegania tych reguł. Reguły powinny być udokumentowane w prawidłowy i kompletny sposób, aby uniknąć nieporozumień zarówno ze strony użytkownika, jak i twórcy systemu informatycznego. W gotowym systemie informatycznym nie powinno być możliwości usuwania lub modyfikowania obiektów przez użytkownika w przypadku, kiedy nie jest to dozwolone w regułach biznesowych.

W biurze handlowym kupca Hanzy obiektem jest księga, na przykład księga zamówień wraz z opiekującym się nią księgowym. Diagram stanów obiektu zawiera reguły, których musi przestrzegać urzędnik podczas „obsługi" tej księgi. Te reguły definiuje księga zasad. Znajduje się w niej na przykład informacja o tym, że nie można anulować zamówienia dostarczonego, lecz jeszcze nie zapłaconego. Jeśli ktoś poobserwowałby pracę księgowego przez odpowiednio długi czas, mógłby spisać wszystkie zdarzenia związane z księgą zamówień. Obsługuje ona na przykład następujące zdarzenia: zapisanie, modyfikacja, dostarczenie, anulowanie lub zapłata. Diagram stanów w perspektywie zachowań zawiera właśnie wynik tego typu obserwacji życia obiektu.

4.3.2. Elementy perspektywy

Perspektywa zachowań, której przykład przedstawia rysunek 4.46, składa się z wielu diagramów stanów, z których każdy przedstawia zachowanie indywidualnego obiektu. Z tego powodu wszystkie diagramy stanów połączone w całość prezentują zachowanie wszystkich obiektów w systemie informatycznym. W praktyce na tym diagramie nie przedstawia się wszystkich diagramów, lecz jedynie te, które spełniają następujące wymagania:

- zawierają wiele ważnych reguł biznesowych;
- opisują ważne obiekty.



Rysunek 4.46. Perspektywa zachowań

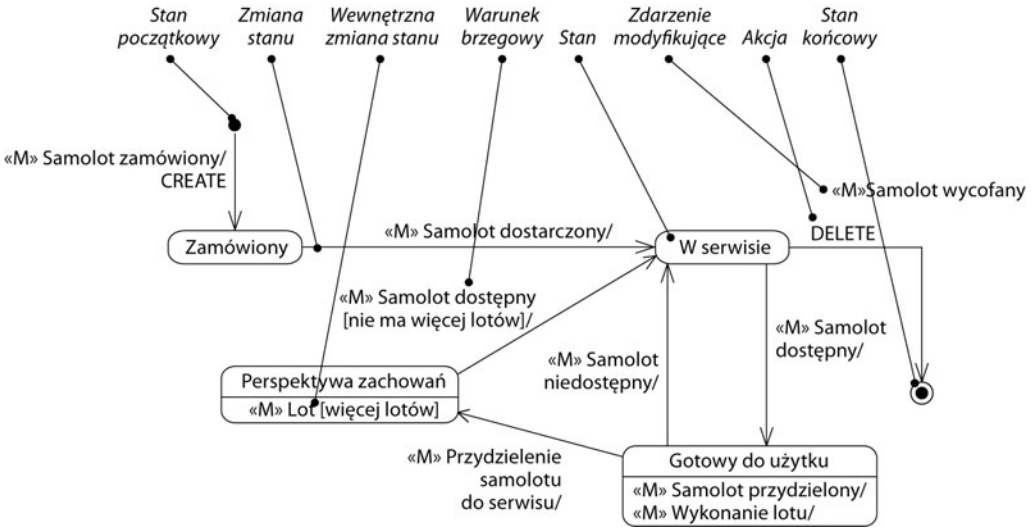
4.3.3. Diagram stanów

W przypadku diagramów stanów takich jak te na rysunku 4.47 pracujemy z następującymi elementami:

Stan początkowy

Stan początkowy reprezentuje źródło wszystkich obiektów.

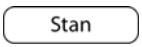
To nie jest zwykły stan, ponieważ obiekty jeszcze nie istnieją.



Rysunek 4.47. Elementy diagramu stanów

Stan

Stan obiektu jest zawsze zdeterminowany przez jego atrybuty i asocjacje. Stany w diagramie stanów reprezentują zbiór kombinacji wartości, w których obiekt w reakcji na zdarzenia zachowuje się zawsze w określony sposób.



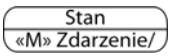
Z tego powodu nie każda modyfikacja atrybutu prowadzi do nowego stanu.

Zmiana

Zmiana reprezentuje przejście z jednego stanu do drugiego.

Zmiana wewnętrzna

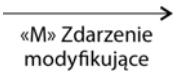
Zmiana wewnętrzna jest taką formą zmiany, gdzie stan nie zmienia się. Oznacza to w praktyce, że obiekt obsługuje zdarzenie bez zmiany stanu.



Zdarzenia inicjujące wewnętrzne zmiany są wypisane w dolnej części symbolu stanu. Na przykład **karta stałego klienta** w stanie normalnym pozostaje w tym stanie również po zajściu zdarzenia **«M» dopisanie kilometrów**.

Zdarzenie modyfikujące

Zdarzenie modyfikujące jest inicjatorem przejścia z jednego stanu w inny lub zmiany wewnętrznej, gdy stan pozostaje ten sam.



Akcja

Akcja jest aktywnością obiektu inicjującego zdarzenie.

«M» Zdarzenie/Akcja

Akcja opisuje reakcję obiektu na zdarzenie. Opis może być tekstowy lub sformalizowany.

Warunek brzegowy

Warunek brzegowy to taki, który musi być spełniony, aby odbyła się odpowiednia dla niego zmiana.

[Warunek brzegowy]

Warunki brzegowe mogą być używane do dokumentowania sytuacji, w których to samo zdarzenie może prowadzić do różnych zmian.

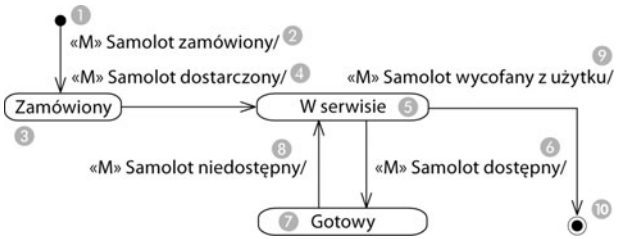
Stan końcowy

Stan końcowy reprezentuje koniec istnienia obiektu.

Stan końcowy nie jest w rzeczywistości stanem obiektów, ponieważ obiekty w tym stanie już nie istnieją.

Czytanie diagramów stanów

Diagram przedstawiony na rysunku 4.48 zawiera wszystkie stany, jakie może przyjąć obiekt w trakcie swojego istnienia. Dodatkowo przedstawia też zmiany między stanami i zdarzenia inicjujące te zmiany.



Rysunek 4.48. Diagram stanów ze zdarzeniami

Każdy obiekt klasy **samolot** pojawia się znikąd (1) (stan początkowy) i odchodzi donikąd (10) (stan końcowy). Taka sytuacja jest z reguły typowa dla większości klas — każda klasa posiada stan początkowy (1) i stan końcowy (10).

W trakcie swojego istnienia obiekt **samolot** (ciągle mamy na myśli obiekt w systemie informatycznym, nie prawdziwy samolot) może przyjąć jeden z trzech stanów: **zamówiony** (3), **w serwisie** (5) i **gotowy do użycia** (7).

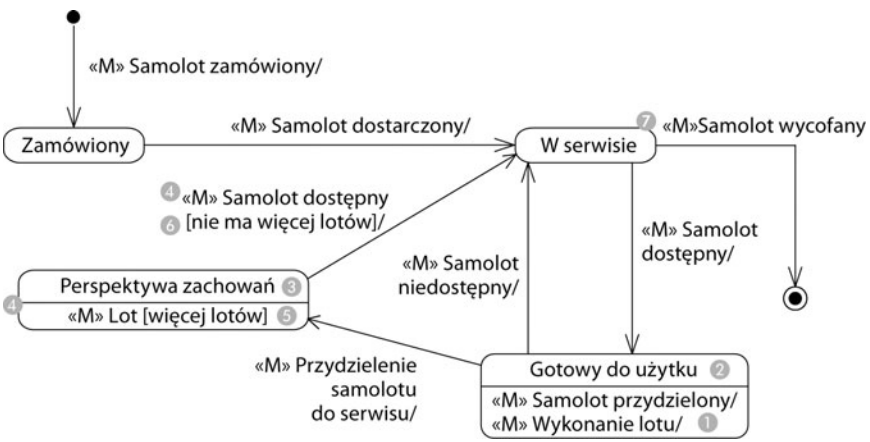
Zdarzenie **«M» samolot zamówiony** prowadzi do sytuacji, gdy znikąd (1) w systemie informatycznym pojawia się samolot (narodziny). Natychmiast po narodzinach obiekt przechodzi w stan **zamówiony** (3).

Jeśli wystąpi zdarzenie **«M» samolot przybył** (4), a samolot pozostaje w stanie **zamówiony** (3), zdarzenie to powoduje zmianę tego stanu na **w serwisie** (5). Jeśli samolot znajduje się w jakimkolwiek innym stanie (poza **zamówiony**), nie następuje zmiana stanu.

W wyniku zdarzeń **«M» samolot udostępniony** (6) i **«M» samolot wstrzymany** (8) obiekt **samolot** wielokrotnie w trakcie swojego istnienia zmienia stany między **w serwisie** (5) a **gotowy do użycia** (7).

Na końcu swojego cyklu życia obiekt **samolot** znika, co jest skutkiem zdarzenia **«M» samolot wycofany z użytku** (9). Samolot znika w nicości (10), co oznacza, że obiekt zostaje usunięty (śmierć).

Rysunek 4.49 przedstawia więcej elementów, które mogą znaleźć się na diagramie stanów.

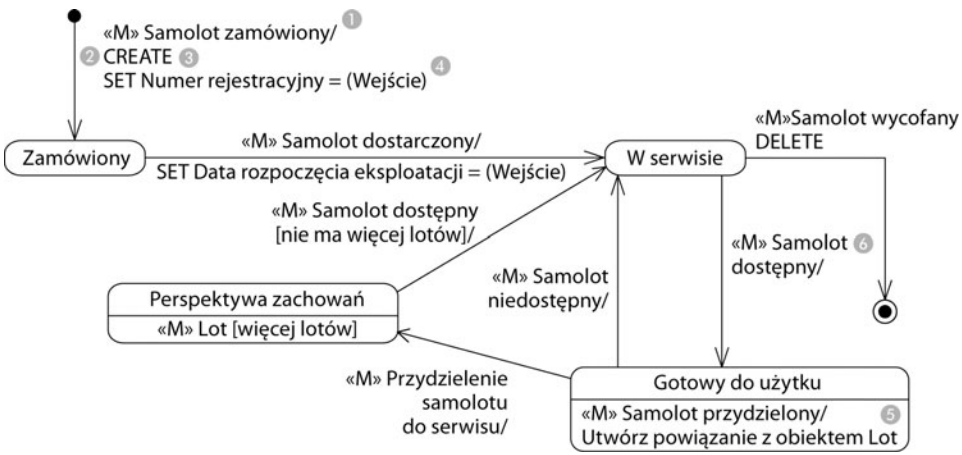


Rysunek 4.49. Diagram stanów ze zmianami wewnętrznymi i warunkami brzegowymi

Oprócz opisanych wyżej zmian istnieją również zmiany wewnętrzne. Zdarzenie **«M» przydział lotu** (1), występujące w momencie przypisania samolotu do lotu, nie powoduje zmiany stanu obiektu **samolot**. Samolot pozostaje w stanie **gotowy do użycia** (2) zarówno przed wystąpieniem tego zdarzenia, jak i po nim.

Warunek brzegowy pozwala na zaakceptowanie lub odrzucenie zdarzenia w zależności od spełnienia tego warunku. Jeśli w stanie **przeznaczony do serwisu** (3) wystąpi zdarzenie **«M» wykonanie lotu** (4), reakcja obiektu jest uzależniona od warunku brzegowego ujętego w nawiasy kwadratowe. Jeśli spełniony jest warunek **[więcej lotów]** (5) (co oznacza, że samolot w danej chwili ma przydzielone loty), odbywa się zmiana wewnętrzna. Samolot nadal pozostaje w stanie **przeznaczony do serwisu** (3). Jeśli jednak spełniony jest warunek **[nie ma więcej lotów]** (co oznacza, że samolot nie ma przydzielonych żadnych następnych lotów), następuje zmiana stanu na **w serwisie** (7).

Akcje opisują sposób reakcji obiektu na zdarzenia modyfikujące. Rysunek 4.50 przedstawia kilka typów akcji. Nazwę akcji wpisuje się po nazwie zdarzenia i ukośniku (1). Akcje CREATE (2) i SET numer rejestracyjny = (input) (4) występują w wyniku zdarzenia modyfikującego **«M» samolot zamówiony**. CREATE informuje o tym, że tworzony jest nowy samolot. SET numer rejestracyjny = (input) informuje o tym, że wartość wpisywana przez użytkownika w ramach przypadku użycia jest przypisywana atrybutowi **numer rejestracyjny**. Poszczególne akcje oddziela się średnikami (;) (3). Oprócz akcji CREATE i SET (zobacz punkt 4.3.4, „Konstruowanie diagramów stanów") mogą występować tu również akcje opisane zwykłym tekstem. W następstwie zdarzenia modyfikującego **«M» przydział lotu** wywoływana jest akcja **utworzenie związku z lotem** (5), która wskazuje na to, że jest tworzony związek między obiektem **samolot** a obiektem **lot**. Jeśli do zdarzenia nie zostanie przydzielona żadna akcja (6), może to oznaczać, że akcja nie została określona lub że w wyniku zdarzenia ulega zmianie stan obiektu.



Rysunek 4.50. Diagram stanów

Po głębszej analizie studium przypadku zauważymy, że diagram stanów z rysunku 4.50 należy uzupełnić o kolejne stany i zdarzenia.

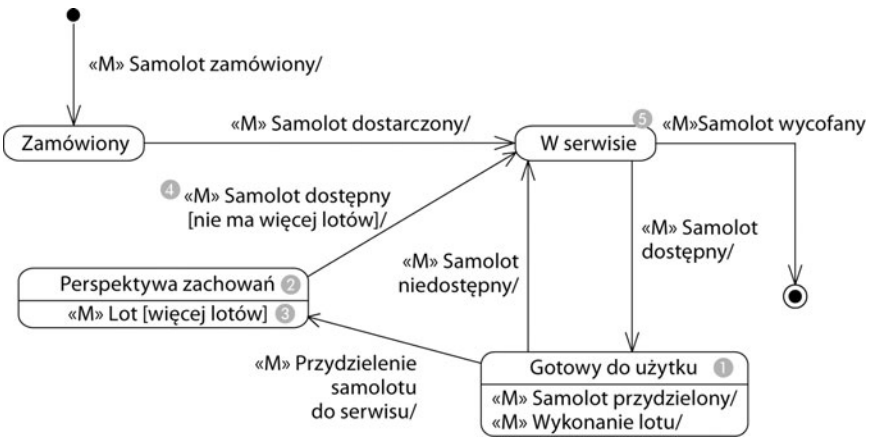
Diagram stanów dokumentujący wszystkie możliwe działania obiektu można czytać po prostu w sposób sekwencyjny. Jednak o wiele użyteczniejsza forma korzystania z diagramu stanów polega na tym, że odbiorca stawia sobie kilka pytań i szuka na nie odpowiedzi:

- Co dzieje się z obiektem w wyniku zdarzenia?

Odpowiedź na to pytanie jest w każdym wypadku uzależniona od bieżącego stanu obiektu, zatem pytanie to powinno w rzeczywistości brzmieć następująco:

- W jaki sposób obiekt w każdym ze swoich stanów reaguje na określone zdarzenie?
- Które zdarzenia mają znaczenie dla obiektu?
- W jaki sposób (w wyniku których zdarzeń) obiekt może opuścić określony stan?
- W jaki sposób (w wyniku których zdarzeń) obiekt może osiągnąć określony stan?

Spróbujmy odpowiedzieć na niektóre z tych pytań w oparciu o diagram stanów przedstawiony na rysunku 4.51.



Rysunek 4.51. Selektywne odczytywanie diagramu stanów

- W jaki sposób obiekt **samolot** w stanie **gotowy do użytku** (1) reaguje na zdarzenie **«M» przydział samolotu**?

  Aby odpowiedzieć na to pytanie, musimy sprawdzić, czy zdarzenie **«M» przydział samolotu** jest dopuszczalne w stanie **gotowy do użycia** (1). Zdarzenie jest dopuszczalne, jeśli istnieje zmiana stanu (strzałka) opisana nazwą zdarzenia lub jeśli istnieje wewnętrzna zmiana stanu (wpis w dolnej części symbolu stanu). W naszym przykładzie nie istnieje odpowiednia zmiana stanu, lecz istnieje zmiana wewnętrzna. Oznacza to, że obiekt w stanie **gotowy do użycia** (1) akceptuje zdarzenie **«M» przydział samolotu**, po którym pozostaje w stanie **gotowy do użycia** (1).

- W jaki sposób obiekt **samolot** w stanie **przeznaczony do serwisu** (2) reaguje na zdarzenie **«M» wykonanie lotu**?

  Aby odpowiedzieć na to pytanie, należy sprawdzić, czy zdarzenie **«M» wykonanie lotu** jest dopuszczalne w stanie **przeznaczony do serwisu** (2). W naszym przykładzie zachodzi wówczas zarówno zmiana stanu, jak i zmiana wewnętrzna. W reakcji na każde zdarzenie możliwa jest tylko jedna zmiana stanu (obiekt **samolot** w każdym momencie może być tylko w jednym stanie), potrzebne są zatem kryteria określające zmianę, która może zajść w wyniku zdarzenia. W tym przypadku zadanie to jest

dość proste dzięki warunkom brzegowym **[więcej lotów]** (3) i **[nie ma więcej lotów]** (4). Musimy sprawdzić, czy istnieje więcej lotów przydzielonych do tego samolotu. Czytamy: obiekt **samolot** w stanie **przeznaczony do serwisu** (2) przyjmuje zdarzenie **«M» wykonanie lotu** i zmienia swój stan na **w serwisie** (5), ponieważ nie ma przydzielonych więcej żadnych lotów.

- W jaki sposób reaguje obiekt **samolot** w stanie **przeznaczony do serwisu** (2) na zdarzenie **«M» przydział lotu**?

  Aby odpowiedzieć na to pytanie, należy sprawdzić, czy zdarzenie **«M» przydział lotu** jest dopuszczalne w stanie **przeznaczony do serwisu** (2). W naszym przykładzie nie istnieje tu zmiana stanu ani zmiana wewnętrzna. Oznacza to, że obiekt **samolot** w stanie **przeznaczony do serwisu** (2) nie akceptuje zdarzenia **«M» przydział lotu**. System informatyczny powinien poinformować użytkownika, dlaczego operacja przydziału lotu nie została zakończona pomyślnie.

- Jakie zdarzenia są dopuszczalne dla obiektu **samolot**?

  Odpowiedź: wszystkie zdarzenia zawarte na diagramie stanu obiektu **samolot** są dopuszczalne dla tego obiektu, co oznacza, że wszystkie zdarzenia są dopuszczalne w co najmniej jednym stanie tego obiektu. Wszystkie pozostałe zdarzenia są niedopuszczalne dla obiektu **samolot**. W takim razie jedynymi zdarzeniami dopuszczalnymi dla obiektu **samolot** są **«M» samolot zamówiony**, **«M» samolot dostarczony**, **«M» samolot dostępny**, **«M» samolot niedostępny**, **«M» przydział lotu**, **«M» wykonanie lotu**, **«M» przydzielenie samolotu do serwisu**, **«M» samolot wycofany z użytku**.

- Jakie zdarzenia obiektu **samolot** mogą spowodować przejście w inny jego stan **w serwisie** (5)?

  Aby odpowiedzieć na to pytanie, musimy przeszukać wszystkie przejścia (strzałki) wychodzące ze stanu **w serwisie** (5) do innego stanu. W naszym przykładzie występują dwa takie przejścia. Obiekt **samolot** w stanie **w serwisie** (5) może mienić stan wyłącznie wskutek zdarzeń **«M» samolot dostępny** lub **«M» samolot wycofany z użytku**.

- Wskutek jakich zdarzeń obiekt **samolot** może osiągnąć stan **gotowy do użytku** (1)?

  Aby odpowiedzieć na to pytanie, należy przeszukać wszystkie przejścia (strzałki) prowadzące do stanu **gotowy do użytku** (1). W naszym przykładzie występuje dokładnie jedno takie przejście: obiekt **samolot** może osiągnąć stan **gotowy do użytku** wyłącznie wskutek zdarzenia **«M» samolot dostępny** (przechodząc do niego ze stanu **w serwisie** (2)).

Omówione wyżej pytania wskazują, że informacje niezapisane na diagramie stanów są równie istotne, jak informacje tam zapisane. Zdarzenia niewystępujące na diagramie stanów nie są dopuszczalne dla danego obiektu. To oznacza, że zdarzenie takie nie będzie reprezentowane przez system informatyczny. W takim wypadku system musi wyświetlić stosowny komunikat o błędzie. Zdarzenia niewystępujące w żadnym ze stanów zawsze będą ignorowane przez system. Z diagramu stanów obiektu **samolot** można odczytać następujące własności:

- Nowy samolot dostarczony na lotnisko nigdy nie może od razu przejść do stanu gotowości do użycia; zawsze musi najpierw trafić do serwisu.
- Samolot gotowy do użycia nie może być bezpośrednio wycofany z użytku. Jeśli wystąpi taka próba, zdarzenie modyfikujące zakończy się niepowodzeniem z odpowiednim komunikatem o błędzie.

4.3.4. Konstruowanie diagramów stanów

Poniższa lista kontrolna przedstawia etapy niezbędne do utworzenia diagramu stanów klasy.

Lista kontrolna 4.6. Konstruowanie diagramów stanów w perspektywie zachowań

- Identyfikacja zdarzeń modyfikujących obiekt — jakie zdarzenia mają na niego wpływ?
- Pogrupowanie zadań w sposób chronologiczny — jak wygląda cykl życia obiektu?
- Modelowanie stanów i przejść — jakie stany może osiągać obiekt?
- Dodanie akcji do diagramu stanów — co robi obiekt?
- Weryfikacja diagramu stanów — czy wszystko jest wykonane prawidłowo?

Identyfikacja zdarzeń modyfikujących obiekt — jakie zdarzenia na niego mają wpływ?

Na pierwszym etapie należy zidentyfikować zdarzenia modyfikujące odpowiednie dla analizowanego obiektu. Chodzi o określenie, które z nich inicjują akcje i (lub) przejścia stanów. Tutaj pomocne będą odpowiedzi na następujące pytania:

- Które zdarzenia modyfikujące prowadzą do utworzenia lub usunięcia obiektu?
- Które zdarzenia modyfikujące definiują lub modyfikują wartości atrybutów?
- Które zdarzenia modyfikujące tworzą związki z innymi obiektami lub usuwają te związki?
- Które zdarzenia modyfikujące skutkują przejęciem stanów obiektu?
- Które zdarzenia modyfikujące zdefiniowane w diagramie przypadków użycia perspektywy zewnętrznej mają wpływ na obiekt?

Odpowiedzi na te pytania składają się na listę zdarzeń modyfikujących obiekt. Wszystkie zdarzenia modyfikujące wynikają z przypadków użycia; dla każdego zdarzenia modyfikującego należy znaleźć przypadek użycia na diagramie sekwencji przypadków użycia. Jeśli go tam nie ma, należy uzupełnić ten diagram. Jedyną drogą, jaką informacja o zdarzeniu może trafić do systemu informatycznego, jest obsługa przypadku użycia. Jeśli więc znajdziemy zdarzenie, dla którego nie zdefiniowano wcześniej przypadku użycia, należy uzupełnić diagram przypadków użycia.

W studium przypadku znaleźliśmy następujące zdarzenia modyfikujące dla obiektu **lot**:

- **«M» lot zdefiniowany**;
- **«M» lot rozpoczęty**;
- **«M» lot zakończony lądowaniem**;
- **«M» lot anulowany**;
- **«M» zmiana daty lotu**;
- **«M» lot nieprawidłowy**;
- **«M» numer lotu nieprawidłowy**.

Pogrupowanie zdarzeń w sposób chronologiczny — jak wygląda cykl życia obiektu?

Zdarzenia modyfikujące zidentyfikowane na poprzednim etapie należy podzielić na grupy: zdarzenia prowadzące do utworzenia obiektu (narodziny), zdarzenia istotne w trakcie normalnego życia obiektu (życie), zdarzenia prowadzące do usunięcia obiektu (śmierć). Tutaj odpowiadamy na pytanie:

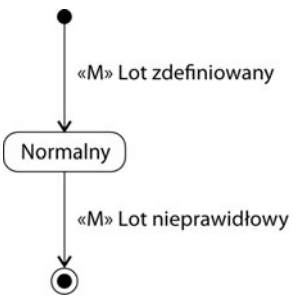
- Do którego stanu życia obiektu należy każde zdarzenie modyfikujące?

Zdarzenia modyfikujące z naszego studium przypadku dla obiektu **lot** pogrupowaliśmy następująco:

- narodziny: **«M» lot zdefiniowany**;
- życie: **«M» lot rozpoczęty**, **«M» lot zakończony lądowaniem**, **«M» lot anulowany**, **«M» zmiana daty lotu**;
- śmierć: **«M» lot nieprawidłowy**, **«M» numer lotu nieprawidłowy**.

Modelowanie stanów i przejść — jakie stany może osiągać obiekt?

W pierwszym szkicu można zdefiniować bardzo prosty diagram stanów, składający się ze stanów: początkowego, normalnego i końcowego. Tego typu szkic diagramu dla obiektu **lot** przedstawia rysunek 4.52:



Rysunek 4.52. Prosty diagram stanów

Przyjmując ten prosty diagram jako podstawę, można dodawać kolejne zidentyfikowane zdarzenia modyfikujące. Na tym etapie prac przydatne są odpowiedzi na następujące pytania:

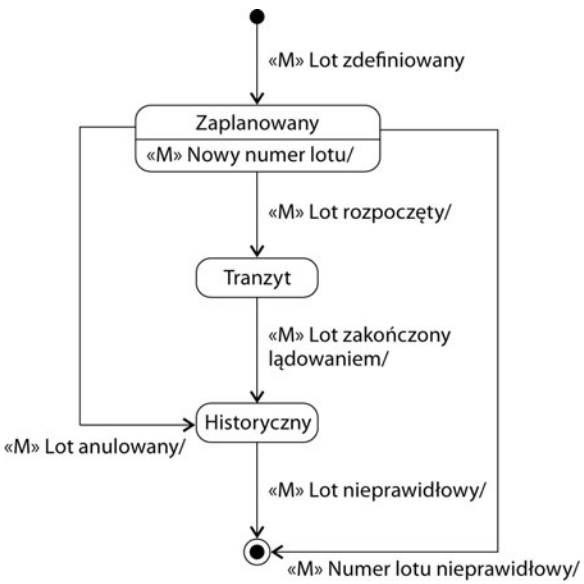
- Czy zdarzenie modyfikujące jest dozwolone we wszystkich przypadkach, czyli dla wszystkich stanów, czy też jedynie w wybranych stanach obiektu?

  Poszczególne przypadki określające możliwość wystąpienia zdarzenia odpowiadają stanom obiektu. Więcej informacji na ten temat można znaleźć w punkcie 4.2.3, „Statyczne i dynamiczne reguły biznesowe".
- W których stanach znajduje się obiekt po wystąpieniu zdarzenia modyfikującego?

  Nowy stan jest uzależniony od stanu obiektu przed wystąpieniem zdarzenia modyfikującego.
- Czy przejście do nowego stanu jest uzależnione od spełnienia określonych warunków?

  Aby udokumentować sytuację, w której zdarzenie modyfikujące może mieć wpływ na zmianę stanu (w zależności od warunku), lub aby tego stanu nie modyfikować, posługujemy się warunkami brzegowymi (rysunek 4.49).

Na przykład w naszym studium przypadku zdarzenie **«M» lot rozpoczęty** jest dopuszczalne jedynie w przypadku, gdy lot jeszcze się nie rozpoczął. Po udzieleniu odpowiedzi na wszystkie powyższe pytania dla każdego zdarzenia modyfikującego uzyskamy gotowy diagram stanów, którego przykład przedstawia rysunek 4.53.



Rysunek 4.53. Diagram stanów dla klasy „lot"

Dodanie akcji do diagramu stanów — co robi obiekt?

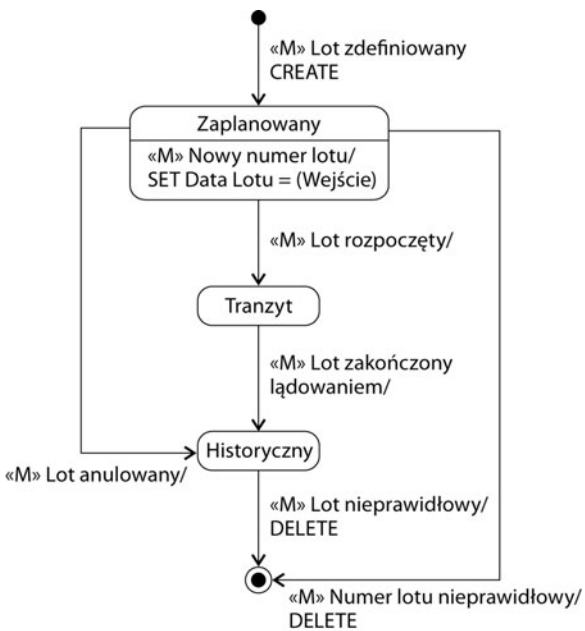
Po zidentyfikowaniu i zamodelowaniu zdarzeń modyfikujących obiektu należy określić ich skutki w postaci akcji. Należy odpowiedzieć na następujące pytania:

- W których przypadkach niezbędne są akcje podejmowane w celu modyfikacji wartości atrybutów?
- W których przypadkach niezbędne są akcje podejmowane w celu obsługi związków?
- W których pozostałych przypadkach niezbędne są akcje (takie jak na przykład aktywacja zapytań, obliczenia)?

Akcje wprowadza się do diagramu stanów. Na prezentowanym tu poziomie szczegółowości akcje możemy wprowadzać w sposób nieformalny, czyli w postaci opisu słownego. Z naszego doświadczenia wynika jednak, że lepiej sprawdza się pewien ustalony poziom formalizmu, gdzie do określania akcji stosuje się odpowiednie słowa kluczowe:

- **CREATE** i **DELETE**: utworzenie lub usunięcie obiektu klasy (nazwę klasy można pominąć);
- **SET** <atrybut> **:= …:** ustawienie wartości atrybutu;
- **TIE TO** <obiekt>**/CUT FROM** <obiekt>**:** ustanowienie i usunięcie związku z innym obiektem.

Rysunek 4.54 przedstawia diagram stanów dla klasy **lot** z uwzględnieniem akcji.



Rysunek 4.54. Diagram stanów dla klasy „lot", uwzględniający akcje

Weryfikacja diagramu stanów — czy wszystko jest wykonane prawidłowo?

Kompletny diagram stanów można zweryfikować za pomocą poniższej listy kontrolnej.

Lista kontrolna 4.7. Weryfikacja diagramu stanów w perspektywie zachowań

- Czy został zdefiniowany stan końcowy, czy obiekt będzie teoretycznie istniał bez końca (bez określonego zdarzenia śmierci)?
- Czy istnieje (bezpośrednie lub niebezpośrednie) przejście do stanu końcowego?
- Czy istnieje rozróżnienie (o ile ma ono znaczenie) między śmiercią logiczną (zamrożenie obiektu) a fizyczną (usunięcie z systemu)?
- Czy dla każdego stanu istnieje przynajmniej jedno zdarzenie, dla którego reakcja jest unikalna? Jeśli nie, stan należy zmodyfikować.
- Czy dwa lub więcej przejść inicjowanych przez to samo zdarzenie wychodzące z tego samego stanu jest kontrolowanych przez warunek **rozłączny** (gdy nie ma możliwości wystąpienia kilku przebiegów jednocześnie)?

4.4. Perspektywa interakcji

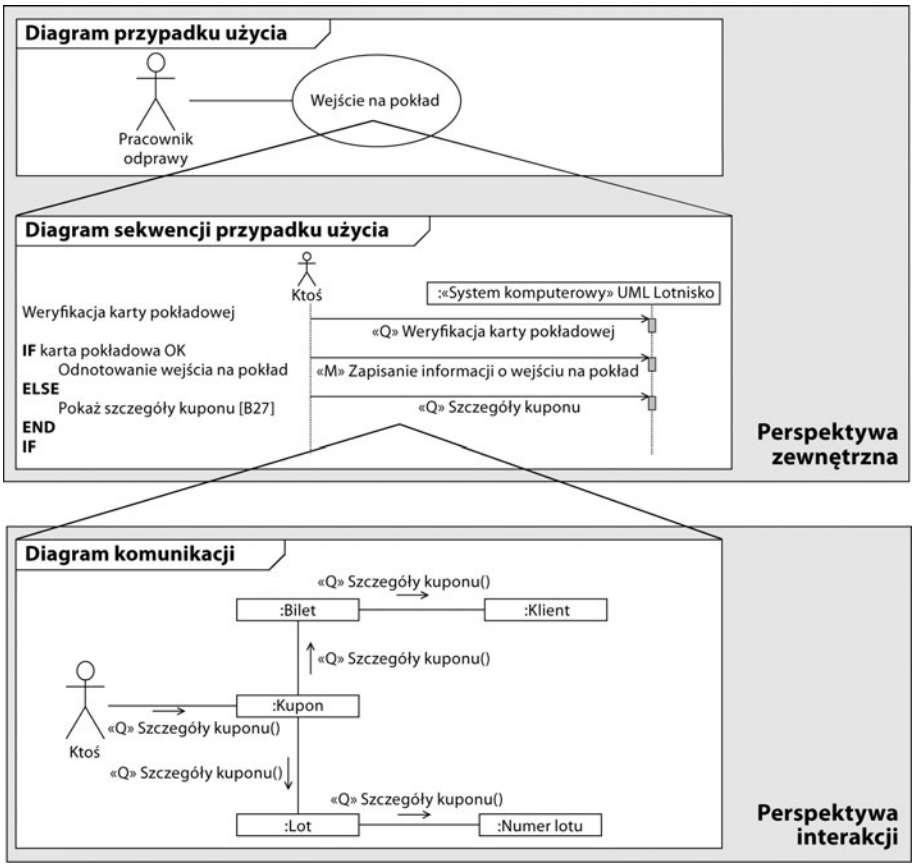
4.4.1. Obserwacja zdarzeń wewnątrz systemu informatycznego

W większości przypadków do zrozumienia systemu nie wystarczy określenie struktur. Nawet wyposażeni w doskonały projekt architektoniczny rafinerii ropy naftowej nie jesteśmy w stanie zrozumieć, w jaki sposób ropę przetwarza się w benzynę. Dopiero po zdefiniowaniu procesu ekstrakcji (przepływów) proces ten staje się zrozumiały. Identyfikacja przepływów jest ważnym narzędziem służącym do objaśniania mechanizmów działania. Ta sama zasada zachodzi również w procesie modelowania systemów informatycznych. Statyczna perspektywa klas nie wystarczy do zrozumienia działania systemu. Co na przykład stanie się w systemie, gdy pasażer zgłosi się do odprawy? Jakie przepływy są realizowane? Jakie obiekty ulegają zmianom? Na tego typu pytania ma odpowiedzieć perspektywa interakcji.

Bardzo ważny jest fakt, że perspektywa interakcji w znacznym stopniu opiera się na weryfikacji i uzupełnianiu elementów perspektywy statycznej. Mamy tu bowiem do czynienia z diagramami klas na potrzeby modelowania zapytań, schodzimy też do poziomu poszczególnych atrybutów. W ten sposób upewniamy się, czy zdefiniowane wcześniej diagramy klas spełniają założone oczekiwania. W zależności od tego, jak kompletny jest diagram klas, przy modelowaniu interakcji skupiamy się na innych szczegółach:

- W przypadku dość kompletnego diagramu klas perspektywa interakcji skupia się na jego weryfikacji.
- W przypadku niekompletnego diagramu klas etap przygotowania perspektywy interakcji polega na jego uzupełnianiu.

Równie silny związek istnieje między perspektywą interakcji a przypadkami użycia. Przypadki użycia przedstawiają system z punktu widzenia użytkownika. System informatyczny jest postrzegany jako czarna skrzynka. Perspektywa interakcji stanowi spojrzenie do wewnątrz tej czarnej skrzynki. Ilustruje ona obiekty niezbędne do realizacji określonych zadań systemu oraz sposób komunikacji poszczególnych obiektów. UML w perspektywie interakcji daje do dyspozycji dwa typy diagramów. W dalszej części tego rozdziału każdy z nich omówimy bardziej szczegółowo (rysunek 4.55).



Rysunek 4.55. Hierarchia diagramów

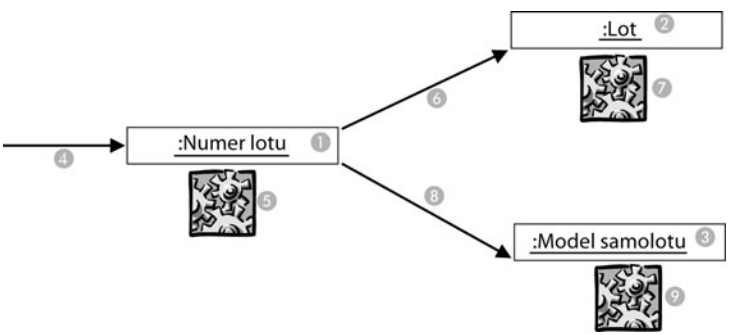
Na rysunku 4.55 został przedstawiony diagram związków między przypadkami użycia oraz diagram komunikacji w perspektywie interakcji:

- U góry znajduje się diagram przypadku użycia **wsiadanie na pokład**.
- Przypadek użycia **wsiadanie na pokład** jest opisany w ramach diagramu sekwencji przypadku użycia. Opis tego przypadku użycia stanowi łańcuch zdarzeń wydobywających dane oraz zdarzeń modyfikujących.
- Przepływ każdego **zapytania** jest opisany w diagramie komunikacji.

W przypadku zdarzeń modyfikujących proces jest analogiczny — ich przepływy są opisane na diagramach sekwencji.

Aby poprawnie interpretować diagramy w perspektywie interakcji, warto poznać od środka sposób działania systemów zorientowanych obiektowo. System działa za pośrednictwem obiektów, które realizują zadania samodzielnie lub delegują je do innych obiektów. Dokładnie w taki sam sposób systemy informatyczne modeluje się w UML-u. Nie ma tu znaczenia, czy system jest w rzeczywistości zaimplementowany za pomocą narzędzi i technik zorientowanych obiektowo. Model systemu informatycznego istnieje na takim poziomie abstrakcji, na którym nie ma nic wspólnego z rzeczywistą implementacją.

Rysunek 4.56 przedstawia współpracę między kilkoma obiektami w systemie informatycznym. Te obiekty to **numer lottu** (1), **lot** (2) i **model samolotu** (3). Obrazowane zadanie polega na usunięciu numeru lotu (na przykład SR9011) z sytemu informatycznego.



Rysunek 4.56. Współpraca obiektów

W naszym modelu z przypadku użycia **anulowanie numeru lotu** wysyłane jest do obiektu **numer lotu** (1) **zdarzenie modyfikujące** (4). W tym momencie uaktywniany jest obiekt **numer lotu** (5), który na przykład może sprawdzać, czy usunięcie jest możliwe. Obiekt ten może również przekazać (6) (8) zdarzenie do innych obiektów (2) (3), które muszą zostać uaktywnione (7) (9).

W diagramach komunikacji i sekwencji perspektywy interakcji są reprezentowane dokładnie tego typu sytuacje współpracy. W tym przypadku uwaga była skupiona na obiektach biorących udział w interakcji oraz na przesyłanych zdarzeniach. Działania podejmowane w ramach

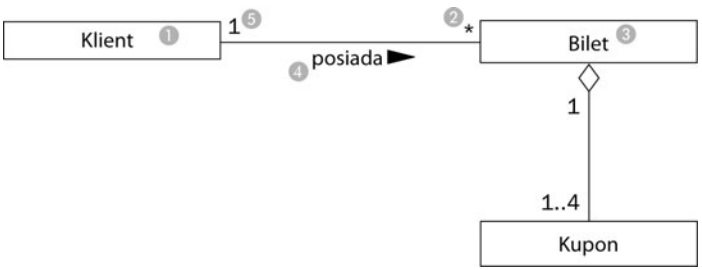
obiektów, czyli ich zachowanie (jak na przykład modyfikacja wartości poszczególnych atrybutów, obliczenia, usuwanie obiektów itp.) nie są uwzględniane na diagramach. Zachowanie obiektów jest reprezentowane w ramach perspektywy zachowań, a ona służy do obrazowania zachowania obiektów w reakcji na zdarzenia.

W biurze handlowym kupca Hanzy interakcje można udokumentować na przykład tak, jak pokazano poniżej. Udajemy się do sekretarza Hildebranda i obserwujemy go podczas wykonywania zadania, na przykład rozliczenia należności z dostawcą. Zapisujemy, w jakiej kolejności sekretarz podchodzi do poszczególnych urzędników, zlecając im podanie informacji z ich ksiąg (zapytanie) lub wprowadzenie informacji do ksiąg (zdarzenie modyfikujące).

Obiekty biorące udział we współpracy (przedstawione na rysunku 4.56) są oparte na mechanizmie wysyłania zdarzeń. Obiekty mogą wysyłać sobie wzajemnie zdarzenia, co z kolei może inicjować inne zdarzenia. Na przykład jeśli z obiektu **bilet** ma zostać wysłane zdarzenie do odpowiedniego obiektu **klient**, jest to możliwe wyłącznie w przypadku, gdy obiekt **bilet** zna związany z nim obiekt **klient**. Dokładnie tego typu informacje są udokumentowane w ramach diagramu klas perspektywy statycznej.

Fragment diagramu klas z rysunku 4.57 informuje nas, że **klient** (1) posiada (4) zero, jeden lub kilka (2) **biletów** (3) i że każdy **bilet** jest w posiadaniu dokładnie jednego (5) **klienta** (zobacz punkt 4.2.5, „Diagram klas"). Z punktu widzenia obiektów wygenerowanych z klas opisanych na tym diagramie oznacza to, że:

- każdy **klient** zna obiekty **biletów**, które ma w swoim posiadaniu;
- każdy **bilet** zna obiekty **kuponów**, które są z nim związane.



Rysunek 4.57. Asocjacje w diagramie klas

Obiekty powiązane znają się wzajemnie. To jest warunek konieczny, aby były one w stanie przesyłać zdarzenia. W perspektywie interakcji zdarzenia są przesyłane w ramach asocjacji zdefiniowanych w diagramach klas. Alternatywnie identyfikację obiektu, do którego jest wysyłane zdarzenie, można zapisać w ramach parametru zdarzenia.

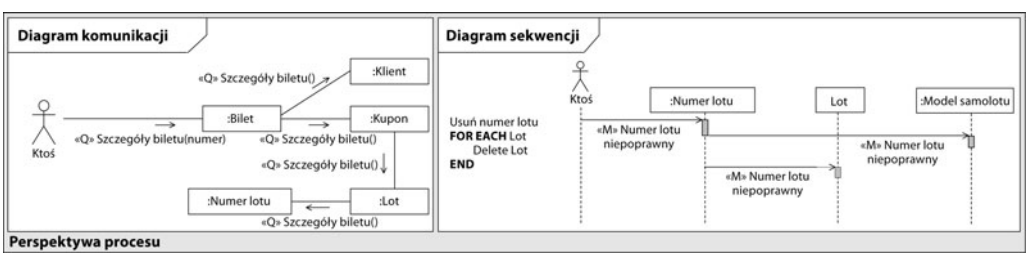
Diagramy UML-a przeznaczone do ilustrowania perspektywy interakcji to diagramy sekwencji i komunikacji. Obydwa przedstawiają zbliżone ujęcia systemu informatycznego. Jeden i drugi pokazuje przepływ współpracy między obiektami, czyli ich interakcje. Między tymi diagramami występuje jednak kilka różnic:

- Diagramy sekwencji przedstawiają chronologię zdarzeń. Oś czasu jest na diagramie umieszczona w pionie. Diagramy komunikacji nie mają osi czasu. Sekwencje na diagramach tego typu z reguły obrazuje się za pomocą numeracji zdarzeń.
- Diagramy komunikacji są podobne do statycznych diagramów klas. Można na nich przedstawić nazwy klas i atrybutów. Dzięki temu niektóre przepływy, takie jak odczyt informacji, można przedstawić w bardzo prosty sposób. W diagramach sekwencji nie przedstawia się atrybutów poszczególnych obiektów.
- Diagramy komunikacji doskonale nadają się do przedstawienia równoległych przepływów interakcji. Choć na diagramach sekwencji również jest to możliwe, w przypadku takiego poziomu szczegółowości stają się już zbyt skomplikowane.

Tego typu różnice stały się powodem, dla którego wybraliśmy właśnie te dwa typy diagramów. Diagramy komunikacji szczególnie dobrze nadają się do dokumentowania zdarzeń wydobywających dane, w których odczytywane są atrybuty, a obiekty nie wykonują żadnej dodatkowej pracy. Diagramy sekwencji lepiej nadają się natomiast do dokumentowania zdarzeń modyfikujących, w których obiekty wykonują znaczącą pracę i w których istotna jest kolejność podejmowania działań.

4.4.2. Elementy perspektywy

Perspektywa interakcji, której przykład przedstawia rysunek 4.58, składa się z dwóch elementów. Są to:



Rysunek 4.58. Perspektywa interakcji

- **Diagram komunikacji**, dokumentujący przepływ zapytań w systemie informatycznym. Każde zapytanie jest elementem przypadku użycia.
- **Diagram sekwencji**, dokumentujący przepływ zdarzeń modyfikujących dane w systemie informatycznym. Zdarzenia modyfikujące również są elementami przypadków użycia.

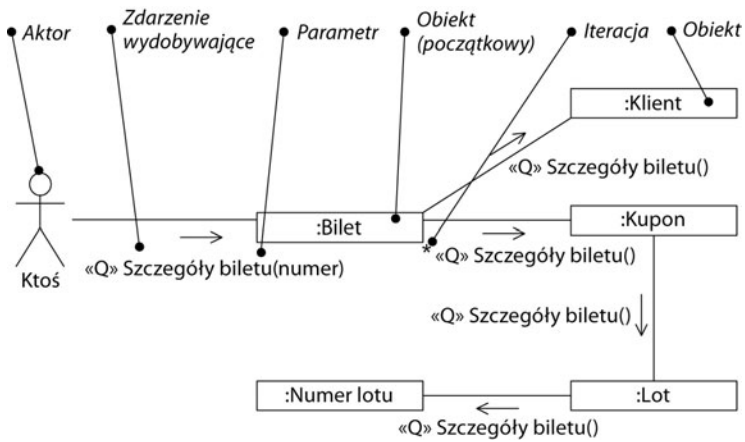
Obydwa diagramy zawierają obiekty systemu informatycznego współpracujące przy przetwarzaniu zdarzeń. Dzięki temu każde zdarzenie wydobywające dane staje się samodzielnym diagramem komunikacji, tak samo jak każde zdarzenie modyfikujące dane staje się samodzielnym diagramem sekwencji.

W rzeczywistości nie ma potrzeby indywidualnego obrazowania (w postaci diagramu) każdego przepływu zapytania i zdarzenia modyfikującego. Tego typu założenie jest oczywiście mało realne z punktu widzenia nakładu pracy. Dokumentować należy jedynie te przepływy, które są szczególnie ważne lub skomplikowane. Przy podejmowaniu decyzji warto wziąć pod uwagę następujące zagadnienia:

- Dla każdego zdarzenia wydobywającego dane należy zweryfikować, czy wszystkie klasy, atrybuty i asocjacje biorące udział w przepływie są uwzględnione na diagramie klasy. W przypadku prostych zapytań wystarczy zaznaczyć odpowiednie elementy danych na wzorcu wydruku lub na formatce ekranowej (zobacz punkt 4.4.5, „Konstruowanie diagramów komunikacji"). W przypadku takich zapytań nie ma sensu tworzenie diagramów. Ten etap dodatkowo pomaga zweryfikować kompletność diagramów klas.
- W formie diagramów komunikacji powinny być udokumentowane zapytania o dużym znaczeniu dla użytkownika lub o bardzo dużym poziomie komplikacji.

4.4.3. Diagram komunikacji

W diagramach komunikacji (przykład na rysunku 4.59) mamy do czynienia z następującymi elementami:



Rysunek 4.59. Elementy diagramu komunikacji

Aktor „ktoś"

Aktor „ktoś" reprezentuje dowolnego aktora z diagramu przypadku użycia. Zdarzenie wydobywające dane dokumentowane w formie diagramu komunikacji może obejmować kilka przypadków użycia, a w nich z kolei mogą brać udział różni aktorzy — dlatego dla uogólnienia stosujemy aktora o nazwie „ktoś".

Dzięki temu komunikacja nie jest przywiązana do konkretnego aktora. W diagramach komunikacji można również zupełnie pominąć aktorów. Z naszego doświadczenia wynika jednak, że taka decyzja powoduje obniżenie ich czytelności.

Zdarzenie wydobywające dane

Z reguły zdarzenie wydobywające dane (zapytanie) jest wysyłane do systemu informatycznego z poziomu przypadku użycia. Jako przykład może posłużyć zapytanie o szczegółowe informacje na temat biletu.

Parametr

Parametry pozwalają przypisać informację konkretnemu zdarzeniu. Parametrem może być na przykład numer biletu — podawany, aby z systemu zostały odczytane informacje na temat właściwego biletu.

Iteracja

Iteracja informuje o tym, że odczytywane są wszystkie obiekty objęte asocjacją, na przykład wszystkie kupony związane z biletem.

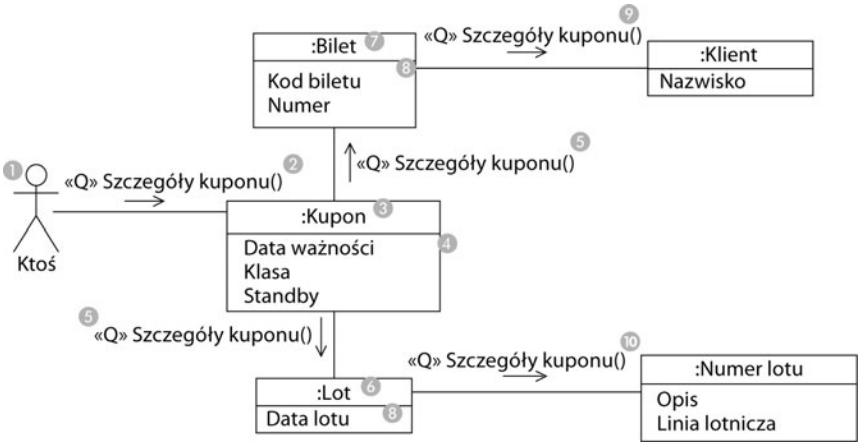
Obiekt i obiekt wejściowy

Obiekt reprezentuje egzemplarz klasy z perspektywy statycznej, na przykład „Janko Walski", który jest obiektem klasy pasażer.

Obiekt wejściowy jest pierwszym obiektem, który otrzymuje zdarzenie od aktora. Ścieżka interakcji rozpoczyna się od obiektu wejściowego.

Odczytywanie diagramów interakcji

Rysunek 4.60 przedstawia diagram komunikacji z aktorem **„ktoś"** i obiektami **bilet**, **klient**, **kupon**, **lot** i **numer lotu**. Diagram ten dokumentuje przepływ zdarzenia **«Q» szczegóły kuponu**.



Rysunek 4.60. Diagram komunikacji

Jeśli rozpoczniemy czytanie od lewej, diagram komunikacji przedstawia się następująco: aktor **„ktoś"** (1) wysyła zdarzenie wydobywające **«Q» szczegóły kuponu** (2) do obiektu klasy **kupon** (3).

Aktor „ktoś"

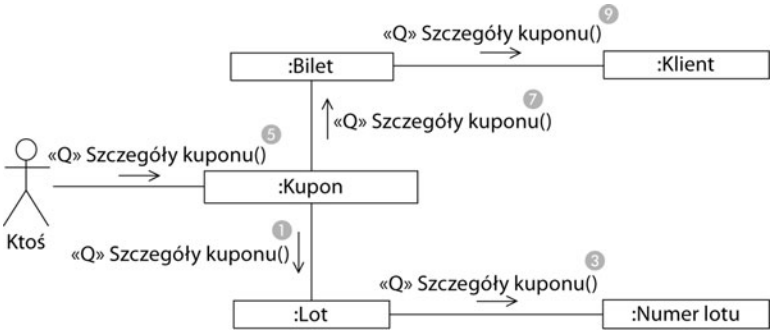
W naszym modelu systemu informatycznego źródłami zdarzeń są przypadki użycia. Diagram komunikacji dokumentuje zdarzenia występujące w kontekście przypadku użycia. Praktyka dowodzi, że warto zastosować niezdefiniowanego aktora **„ktoś"** (1), zamiast wykorzystywać rzeczywistych aktorów z przypadku użycia. Przepływ zdarzenia opisywanego diagramem komunikacji może obejmować różne przypadki użycia i różnych aktorów. Aktor **„ktoś"** (1) zastępuje wszystkich aktorów przypadków użycia, w których może wystąpić zdarzenie **«Q» szczegóły kuponu** (2):

Obiekt **kupon** (3) udostępnia swoje atrybuty: **data ważności**, **klasa** i **standby** (4), po czym przekazuje zdarzenie **«Q» szczegóły kuponu** (5) do dwóch kolejnych obiektów, z którymi łączą go asocjacje: **lot** (6) oraz **bilet** (7).

Te dwa obiekty również zwracają określone atrybuty (8), po czym przekazują zdarzenie **«Q» szczegóły kuponu** (9, 10). W ten sposób diagram komunikacji może służyć do zobrazowania procedury gromadzenia atrybutów z wielu różnych obiektów w wyniku pojedynczego zdarzenia wydobywającego.

W przeciwieństwie do diagramów sekwencji diagramy komunikacji nie zawierają informacji o czasie. Obiekty mogą być pogrupowane na diagramie w zupełnie dowolny sposób. Kolejność przetwarzania zdarzeń jest możliwa do odczytania, ale niebezpośrednio.

W oparciu o diagram sekwencji z rysunku 4.61 można wyciągnąć następujące wnioski (numeracja elementów diagramu została dobrana w sposób losowy, aby uniknąć sugestii co do kolejności ich odczytywania):



Rysunek 4.61. Sekwencja w diagramie komunikacji

- Na początku przez aktora „ktoś" wysyłane jest zdarzenie do obiektu kupon (5).
- Na następnym etapie nie ma możliwości określenia kolejności:
  - Z jednej strony zdarzenie trafia do obiektu **lot** (1), a następnie obiektu **numer lotu** (3);
  - Z drugiej strony zdarzenie trafia do obiektu **bilet** (7), a następnie do obiektu **numer biletu** (9).

Przepływ zdarzenia rozgałęzia się przy obiekcie **kupon**, lecz nie mamy żadnej informacji na temat kolejności. Najczęściej kolejność ta po prostu nie ma żadnego zdarzenia. Bywa jednak inaczej. W takich przypadkach UML umożliwia zastosowanie wskazania sekwencji zdarzeń w diagramie komunikacji za pomocą numeracji.

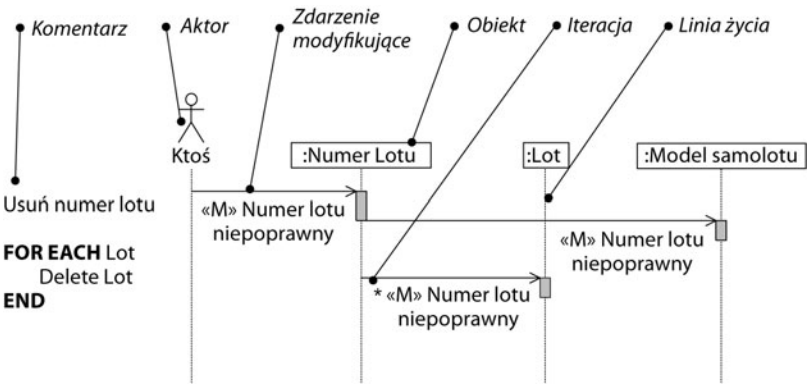
Iteracja informuje o tym, że przetwarzane są wszystkie obiekty związane asocjacją (rysunek 4.62).

Rysunek 4.62. Iteracja w diagramie komunikacji

Z rysunku 4.62 wynika, że zdarzenie **«Q» sztuki bagażu** (1) jest wysyłane do obiektu **kupon** (2), a z niego do **wszystkich** (3) (iteracja) obiektów asocjacji **sztuki bagażu** (4). Iteracja jest dokumentowana za pomocą gwiazdki (*) przed nazwą zdarzenia.

4.4.4. Diagram sekwencji

W diagramach sekwencji (zobacz rysunek 4.63) mamy do czynienia z następującymi elementami:



Rysunek 4.63. Elementy diagramów sekwencji

Komentarz

Przepływ zdarzenia modyfikującego jest dokumentowany z użyciem kombinacji opisu tekstowego i diagramu sekwencji:

Usuń numer lotu

...

W komentarzach ujmowana jest logika przepływu na najwyższym poziomie abstrakcji.

Aktor „ktoś"

Aktor „ktoś" reprezentuje dowolnego aktora z diagramu przypadku użycia. Zdarzenie modyfikujące dokumentowane za pomocą diagramu sekwencji może być powiązane z wieloma diagramami przypadków użycia, a każdy z nich może mieć wielu aktorów — zatem w diagramach sekwencji używa się nieokreślonego aktora „ktoś":

Dzięki temu nie skupiamy się na określonym aktorze.

Zdarzenie modyfikujące

Zdarzenie modyfikujące jest zdarzeniem wysyłanym przez przypadek użycia (najczęściej z interfejsu użytkownika) do systemu informatycznego.

«M» Zdarzenie/

Celem zdarzenia jest modyfikacja informacji zapisanych w systemie informatycznym. Polega ona na utworzeniu, zmianie lub usunięciu obiektów.

Obiekt

Obiekt diagramu sekwencji reprezentuje niezdefiniowany obiekt określonej klasy systemu informatycznego.

:Klasa

Iteracja

Iteracja informuje o tym, że odczytywane są wszystkie obiekty objęte asocjacją, na przykład wszystkie loty związane z numerem lotu.

*

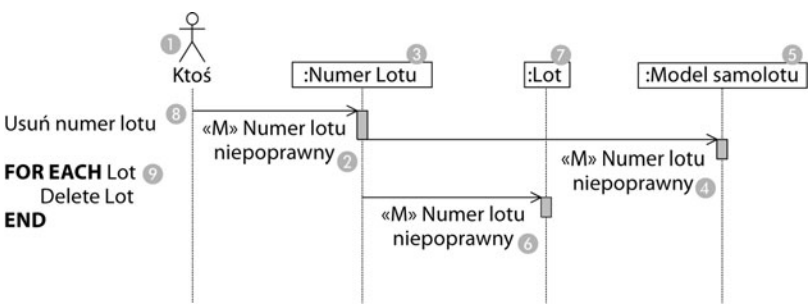
Linia życia

Linia życia jest obiektem reprezentującym życie obiektu (w czasie). Prostokąt („grubsza część") na linii życia wskazuje okresy aktywności obiektu.

W diagramach sekwencji nie ma znaczenia czynnik aktywujący obiekt.

Odczytywanie diagramów sekwencji

Rysunek 4.64 przedstawia diagram sekwencji z obiektami klas **numer lotu**, **lot** i **model samolotu**. Diagram jako całość dokumentuje przepływ zdarzenia modyfikującego **«M» numer lotu niepoprawny**.



Rysunek 4.64. Diagram sekwencji

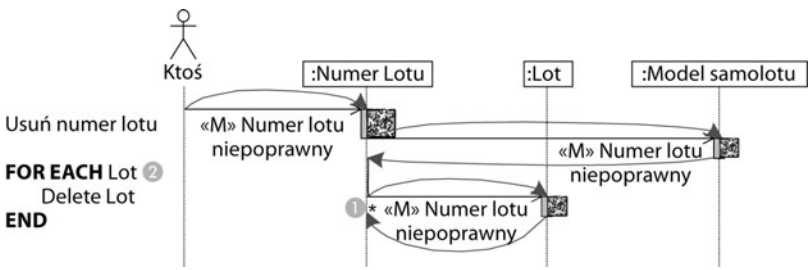
Diagram można odczytywać od góry do dołu. Przepływ rozpoczyna się od **aktora** (1) wysyłającego zdarzenie modyfikujące **«M» numer lotu niepoprawny** (2) do obiektu klasy **numer lotu** (3).

Należy pamiętać (dokładnie tak samo, jak w przypadku diagramu komunikacji), że źródłem zdarzenia modyfikującego jest przypadek użycia. W diagramie sekwencji **aktor** (1) reprezentuje przypadek użycia.

Sposób przetwarzania zdarzenia w ramach obiektu klasy **numer lotu** nie jest możliwy do odczytania z diagramu sekwencji. Pewne wskazówki można znaleźć jedynie w **komentarzach** (8). Dokładnego opisu przetwarzania należy szukać w diagramie stanów klasy **numer lotu** (zobacz podrozdział 4.3, „Perspektywa zachowań").

Na diagramie widać, że obiekt klasy **numer lotu** (3) przekazuje zdarzenie modyfikujące **«M» numer lotu niepoprawny** (4) do obiektu klasy **model samolotu** (5). Przetwarzanie zdarzenia w ramach obiektu nie jest w żaden sposób zobrazowane. Przetwarzanie zdarzenia kończy się na obiekcie **model samolotu** (5), po czym kontrola wraca do nadawcy zdarzenia, czyli do obiektu klasy **numer lotu** (3). Na diagramie nie ma osobnej strzałki informującej o tym powrocie.

Zdarzenie modyfikujące **«M» numer lotu niepoprawny** (6) jest wysyłane również do obiektu klasy **lot** (7). Z faktu, że obiekt klasy **numer lotu** zna więcej niż jeden obiekt klasy **lot** (tę informację można odczytać z diagramu klasy w perspektywie statycznej), wynika, że zdarzenie modyfikujące jest wysyłane do wszystkich obiektów klasy **numer lotu** biorących udział w asocjacji. Gwiazdka * przy zdarzeniu (1) w diagramie sekwencji (rysunek 4.65) informuje właśnie o wystąpieniu iteracji. Zaleca się uzupełnienie diagramu o opis na lewym marginesie (2), co ułatwia jego czytanie.



Rysunek 4.65. Przepływ sterowania w diagramie sekwencji

Rysunek 4.65 przedstawia przepływ sterowania diagramu sekwencji. W naszych przykładach diagramy sekwencji wykorzystujemy wyłącznie do dokumentowania zdarzeń modyfikujących. Jednak UML daje wiele innych możliwości wykorzystania tych diagramów. Z naszego doświadczenia wynika jednak, że w tym przypadku sprawdza się zasada „im mniej, tym lepiej" i że można komunikować się w sposób skuteczny w zakresie perspektywy interakcji przy wykorzystaniu diagramów sekwencji w ten ściśle ograniczony sposób.

4.4.5. Konstruowanie diagramów komunikacji

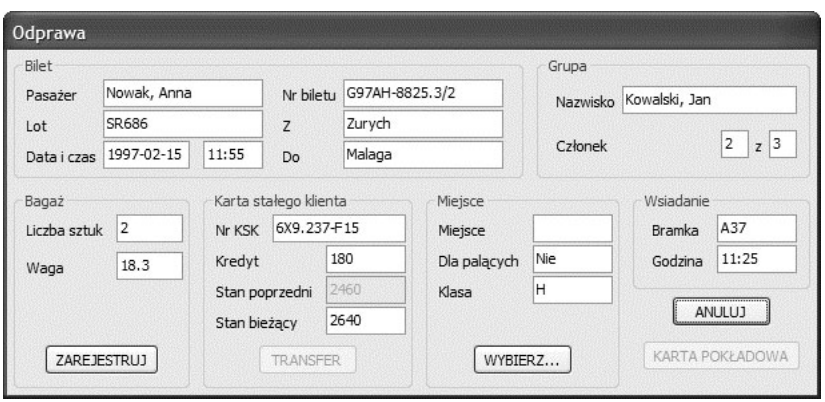
Poniższa lista kontrolna zawiera etapy niezbędne do skonstruowania diagramów komunikacji (po jednym na każde zdarzenie wydobywające dane z przypadku użycia). Każdy z etapów opisujemy poniżej.

Lista kontrolna 4.8. Konstruowanie diagramów komunikacji w perspektywie interakcji

- Szkicowanie wyników zapytań — czego oczekujemy?
- Identyfikacja klas biorących udział w zapytaniu: — które z nich są potrzebne?
- Definiowanie obiektów początkowych — od czego zaczyna się zdarzenie?
- Projektowanie ścieżki zdarzenia — którędy będziemy się poruszać?
- Modyfikacja ścieżki zdarzenia — które dokładnie obiekty są niezbędne?
- Identyfikacja niezbędnych atrybutów — co dokładnie chcemy wiedzieć?
- Weryfikacja diagramu komunikacji — czy wszystko zostało wykonane prawidłowo?

Szkicowanie wyników zapytań — czego oczekujemy?

Początkowym punktem w modelowaniu zdarzenia wydobywającego jest określenie oczekiwanego wyniku. Takim wynikiem może być zawartość monitora lub wydruku, na przykład lista lub raport. Rysunek 4.66 przedstawia okno prototypu systemu informatycznego, a rysunek 4.67 fragment karty pokładowej.



Rysunek 4.66. Prototyp interfejsu systemu odpraw

Dla każdego zapytania dokumentowanego w diagramie komunikacji należy określić oczekiwany wygląd wyniku. Należy odpowiedzieć na następujące pytania:

- W jaki sposób (dokładnie) ma wyglądać wynik zapytania?
- Jakie elementy ma zawierać wynik, które z nich są elementami stałymi, a które zmiennymi, zależnymi od danych?


EKONOMICZNA

Karta pokładowa

Nazwisko
KOWALSKI JAN M

Z ZURYCH
Do HERAKLION

Przewoźnik Lot	Klasa	Data	Czas
EDW761	Y	29MAY0745	

Brama	Godzina wejścia na pokład	Miejsce	Dla palących
B38	0715	12B	NO

Sztuk	Waga
2	34

Rysunek 4.67. Fragment karty pokładowej

Zalecamy naszkicowanie wyniku w postaci graficznej. Każdy element szkicu powinien zawierać informację o tym, czy mamy do czynienia z informacją z systemu informatycznego, czy ze stałym elementem, na przykład tekstowym lub graficznym. W przypadku formularza ekranowego etykiety pól z reguły nie pochodzą z systemu informatycznego. Jeśli jednak system informatyczny pozwala na prezentację interfejsu w wielu językach, możliwe, że etykiety ekranowe poszczególnych elementów formularza będą pobierane właśnie z niego. Innym przypadkiem jest karta pokładowa — tutaj wykorzystywany jest wstępnie drukowany formularz, a więc etykiety pól są stałe. Karta jest po prostu uzupełniana o brakujące informacje pochodzące z systemu informatycznego. Są one nadrukowywane na istniejące elementy stałe. Na rysunku 4.67 elementy zmienne pochodzące z systemu informatycznego są dodatkowo zaznaczone.

Karta pokładowa z naszego studium przypadku zawiera następujące elementy:

EKONOMICZNA, NOWAK, JAN, KRAKÓW, HERAKLION, EDW761, Y, 29 MAJA 0745, B38, 0715, NIE, 2 i 34.

Celem modelowania diagramów komunikacji jest wskazanie, w jaki sposób elementy informacji mogą być odczytane z diagramu klas.

Identyfikacja klas biorących udział w zapytaniu — które z nich są potrzebne?

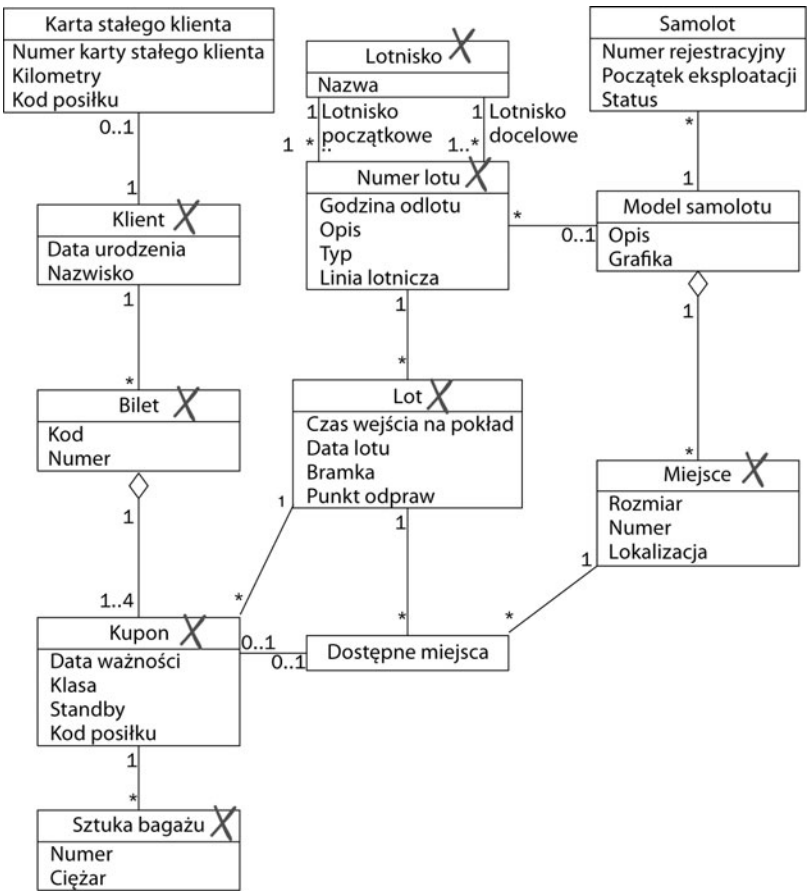
Na podstawie szkicu wyniku określamy klasy niezbędne do wygenerowania informacji. Jako źródło klas wykorzystujemy oczywiście diagram klas.

Na tym etapie przydatne są odpowiedzi na następujące pytania:

- Z których klas mają być odczytywane niezbędne informacje? Dla każdego elementu informacji naszkicowanej poprzednio należy określić klasy i ich atrybuty.

- Które klasy pojawiają się na ścieżce dostępu? Są one potrzebne jako źródła informacji (wartości atrybutów) niezbędnych do wygenerowania wyniku lub z powodu asocjacji, za pośrednictwem których można dotrzeć do klas zawierających informacje.
- Których klas brakuje w diagramie klas? W zależności od poziomu kompletności diagramu może się okazać, że ten etap prac ujawni braki, które trzeba będzie uzupełnić.

W pierwszym kroku niezbędne klasy można po prostu zaznaczyć ręcznie (na wydruku diagramu klas). Na rysunku 4.68 zostały zaznaczone klasy niezbędne do wygenerowania karty pokładowej z rysunku 4.67.



Rysunek 4.68. Uproszczony diagram klas z naniesionymi notatkami

Definiowanie obiektów początkowych — od czego zaczyna się zapytanie?

Dla każdego zapytania definiuje się obiekt początkowy. Ten obiekt może być klasą (rozumianą jako zbiór obiektów) lub określonym obiektem klasy. Jeśli bezpośrednio wskazany jest określony obiekt, na przykład określony obiekt klasy klient lub klasy bilet, musi on być znany przed wysłaniem zdarzenia wydobywającego. Jeśli wykorzystywana jest klasa, do zdarzenia należy dołączyć parametry pozwalające wskazać określony obiekt tej klasy. Każda następna klasa biorąca udział w zapytaniu musi być dostępna jako asocjacja poszczególnych klas. Na tym etapie należy udzielić odpowiedzi na następujące pytania:

- Jaki jest pierwszy obiekt, do którego trafia zdarzenie?
- Które obiekty są już znane?
- W którym momencie należy zacząć gromadzenie informacji?

Aby wygenerować kartę pokładową z naszego studium przypadku, należy zacząć od kuponu wchodzącego w skład biletu lotniczego, dla którego chcemy wygenerować tę kartę.

Projektowanie ścieżki zdarzenia — którędy będziemy się poruszać?

Rozpoczynając od obiektu początkowego, należy określić ścieżkę dostępu do poszczególnych obiektów. Należy zatem odpowiedzieć na pytanie:

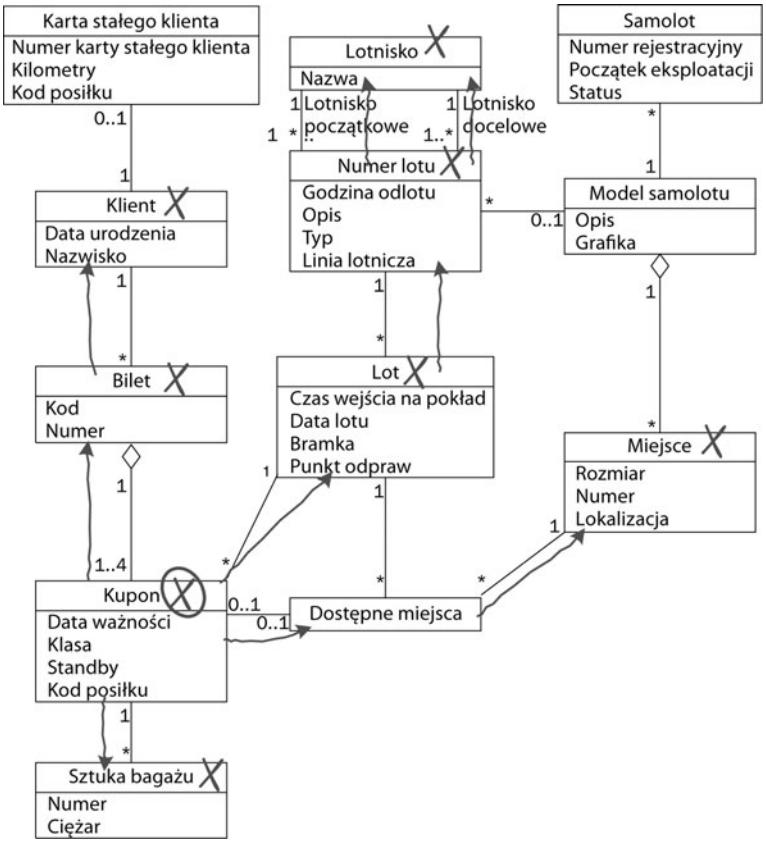
- Jaka ścieżka pozwala na uzyskanie dostępu do poszczególnych obiektów na diagramie klas?

W pierwszym szkicu ścieżka ta może być narysowana ręcznie na diagramie klas. Rysunek 4.69 przedstawia diagram klas ze ścieżkami naniesionymi na potrzeby diagramu komunikacji dotyczącego zdarzenia „wystawienie karty pokładowej".

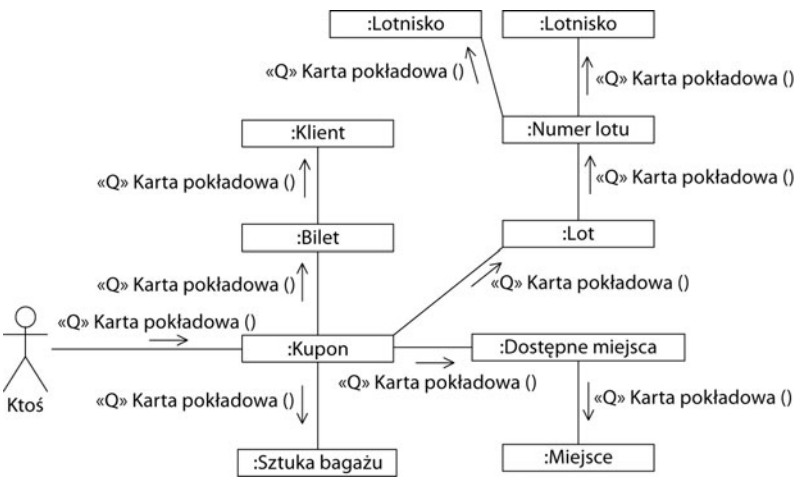
W wielu przypadkach wystarczy zaznaczyć ścieżkę zdarzenia bezpośrednio na diagramie klas. Taka ścieżka informuje o tym, jakie zadanie realizuje dane zapytanie, na przykład wystawienie karty pokładowej. W tym konkretnym przypadku nie ma nawet potrzeby modelowania diagramu komunikacji. Jednak przy skomplikowanych lub ważnych zapytaniach zaleca się tworzenie diagramów komunikacji w oparciu o naniesione ręcznie ścieżki dostępu do atrybutów. Wszystkie tego typu ścieżki na diagramie klas można sprowadzić do diagramu komunikacji; zdarzenie wydobywające informacje polega właśnie na „zbieraniu" informacji po takiej ścieżce. Rysunek 4.70 przedstawia przykładowy wynik tego etapu prac.

Modyfikacja ścieżki zdarzenia — które dokładnie obiekty są niezbędne?

Ścieżkę zdarzenia należy uzupełnić o sekwencje i iteracje. Ten etap jest stosowany wówczas, gdy na ścieżce występuje większa liczba obiektów w ramach asocjacji do innych obiektów. Chodzi o sytuacje, w których asocjacje obejmują większe grupy obiektów innych klas, czyli związek z diagramu klas ma liczność większą niż jeden (z reguły *). W takich przypadkach



Rysunek 4.69. Uproszczony diagram klas z naniesioną ścieżką zdarzeń

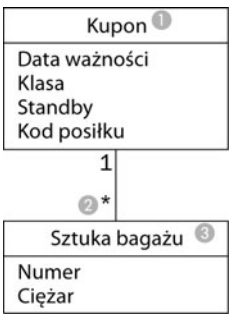


Rysunek 4.70. Pierwszy szkic diagramu komunikacji

należy określić, czy wszystkie obiekty asocjacji mają być uwzględnione kolejno, w ramach iteracji, czy też z takiej grupy należy wybrać tylko właściwe elementy w oparciu o określone kryteria. Na tym etapie należy odpowiedzieć na pytanie:

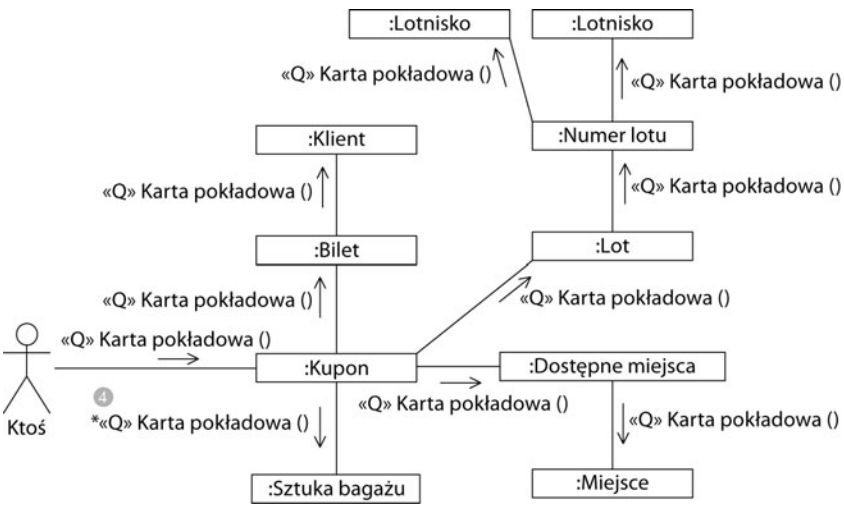
- Jeśli na ścieżce dostępu napotkam więcej niż jeden obiekt, czy mam przeanalizować wszystkie obiekty (iteracja), czy tylko wybrane (selekcja)?

W naszym studium przypadku, którego fragment przedstawia rysunek 4.71, znajdujemy **kilka** (2) **elementów bagażu** (3) raz ścieżkę zdarzenia rozpoczynającą się od **kuponu** (1). Potrzebujemy listy elementów bagażu dla danej karty pokładowej, ponieważ chcemy na tej karcie wydrukować informację na temat całkowitej masy bagażu. Z tego powodu odpowiednia do naszych celów będzie iteracja.



Rysunek 4.71. Selekcja czy iteracja?

Taka sytuacja na diagramie komunikacji przedstawionym na rysunku 4.72 jest zaznaczona przez niewielką gwiazdkę umieszczoną przy nazwie zdarzenia.



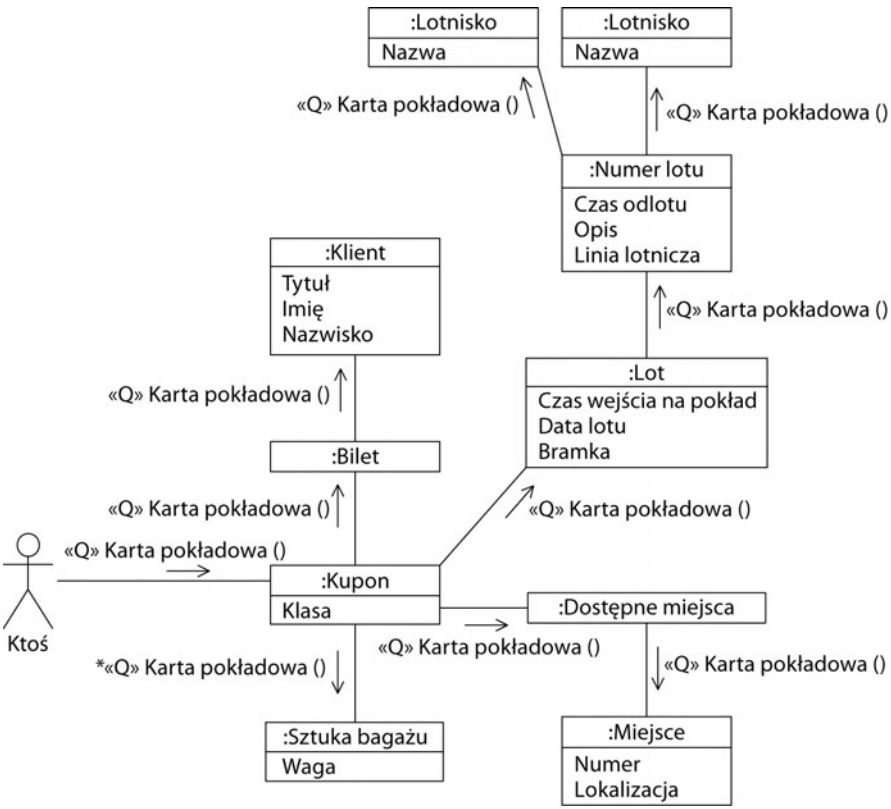
Rysunek 4.72. Diagram komunikacji z iteracją

Identyfikacja niezbędnych atrybutów — co dokładnie chcemy wiedzieć?

Na ostatnim etapie wskazujemy atrybuty niezbędne do wykonania zapytania. Odpowiadamy na następujące pytania:

- Które atrybuty są niezbędne do wygenerowania wyniku zapytania?
- Których atrybutów brakuje na diagramie klas?

Wszystkim elementom danych ze szkicu wyniku zapytania muszą odpowiadać atrybuty klas z diagramu klas perspektywy statycznej. W najprostszym przypadku można zwyczajnie zaznaczyć odpowiednie elementy na wydruku diagramu klas. Po zamodelowaniu diagramu komunikacji można uzupełnić go o atrybuty (rysunek 4.73).



Rysunek 4.73. Diagram komunikacji z atrybutami

Weryfikacja diagramu komunikacji — czy wszystko zostało wykonane prawidłowo?

Kompletny diagram komunikacji można zweryfikować za pomocą poniższej listy kontrolnej.

Lista kontrolna 4.9. Weryfikacja diagramu komunikacji w perspektywie interakcji

- Czy naszkicowane wyniki zapytania mogą być wygenerowane przy użyciu zamodelowanego diagramu komunikacji?
- Czy ścieżki zdarzenia w diagramie komunikacji są zgodne z asocjacjami w diagramie klas?
- Czy z diagramu komunikacji jasno wynika, w których miejscach mamy do czynienia z iteracją, a w których z selekcją — i czy to znaczenie jest zgodne ze zdefiniowanym w diagramie klas?
- Jeśli odpowiedź na wszystkie powyższe pytania brzmi „tak", można założyć, że wszystkie możliwe źródła błędów zostały wyeliminowane.

4.4.6. Konstruowanie diagramów sekwencji

Poniższa lista kontrolna przedstawia etapy niezbędne przy konstruowaniu diagramów sekwencji zdarzenia modyfikującego wynikającego z przypadku użycia. Każdy z etapów omówimy pokrótce.

Lista kontrolna 4.10. Konstruowanie diagramów sekwencji w perspektywie interakcji

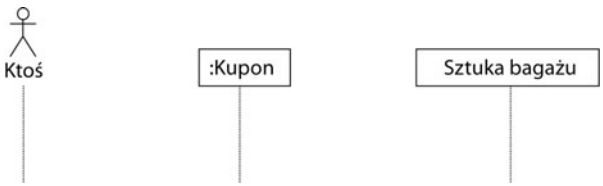
- Identyfikacja klas biorących udział w sekwencji — co ulega modyfikacji dzięki zdarzeniu?
- Określenie obiektu początkowego — od czego rozpoczyna się zdarzenie modyfikujące?
- Propagacja zdarzeń — w jaki sposób zdarzenie jest przekazywane?
- Określenie parametrów zdarzenia — jakie informacje muszą znać obiekty?
- Weryfikacja diagramu sekwencji — czy wszystko jest wykonane prawidłowo?

Identyfikacja klas biorących udział w sekwencji — co ulega modyfikacji dzięki zdarzeniu?

Na pierwszym etapie należy zidentyfikować klasy ulegające zmianom w wyniku zdarzenia modyfikującego. Zadanie to realizuje się w oparciu o diagramy stanów (zobacz podrozdział 4.3, „Perspektywa zachowań"). Należy odpowiedzieć na następujące pytania:

- Które klasy ulegają zmianie w wyniku zdarzenia modyfikującego?
  Odpowiedź na to pytanie wymaga analizy diagramu stanów zawierającego modelowane zdarzenie modyfikujące. Jeśli występuje ono na diagramie stanów dla określonej klasy, właśnie ta klasa ulega zmianom w jego wyniku, czyli jest ono wysyłane do tej klasy.
- Które pozostałe klasy ulegają zmianom w wyniku zdarzenia modyfikującego?
  Może zdarzyć się, że istnieją klasy ulegające zmianom w wyniku zdarzenia, lecz nieujęte w diagramie stanów.

Na pierwsze z tych pytań łatwo odpowiedzieć. Większość narzędzi CASE potrafi w sposób automatyczny wygenerować listę klas ulegających zmianom w wyniku zdarzeń uwzględnionych na innych diagramach. Aby znaleźć pozostałe klasy, należy przyjrzeć się diagramowi klas i zastanowić się nad każdą z nich: czy w wyniku zdarzenia modyfikującego ulega ona zmianom? Należy przyjrzeć się przynajmniej klasom ściśle związanym z klasami ulegającymi zmianom, czyli wyodrębnionym w ramach odpowiedzi na pierwsze pytanie. Oczywiście, w przypadku, gdy uda się zidentyfikować inne klasy ulegające zmianom, należy odpowiednio zmodyfikować diagram stanów, uzupełniając go o zdarzenie modyfikujące. W naszym studium przypadku tego typu zdarzeniem może być na przykład **«M» zarejestrowanie sztuki bagażu**, które modyfikuje klasy **kupon** i **sztuki bagażu**, co jest odzwierciedlone na rysunku 4.74.



Rysunek 4.74. Klasy diagramu sekwencji ulegające modyfikacjom w wyniku zdarzenia

Określenie obiektu początkowego — od czego rozpoczyna się zdarzenie modyfikujące?

Zdarzenie modyfikujące musi mieć określony obiekt początkowy. Może być nim na przykład klasa lub konkretny obiekt klasy. Od obiektu początkowego rozpoczyna się zapytanie. Jeśli określony obiekt klasy jest wskazany bezpośrednio, na przykład jest to konkretny obiekt klasy **lot** lub klasy **bilet**, musi być on znany przed wysłaniem zdarzenia modyfikującego. Jeśli wykorzystywana jest przy tym klasa, do zdarzenia należy dołączyć parametry pozwalające wskazać określony obiekt klasy. Każda następna klasa biorąca udział w zapytaniu musi być dostępna jako asocjacja poszczególnych klas. Na tym etapie należy udzielić odpowiedzi na następujące pytania:

- Jaki jest pierwszy obiekt, do którego trafia zdarzenie?
- Które obiekty są już znane?
- W którym momencie należy zacząć gromadzenie informacji?

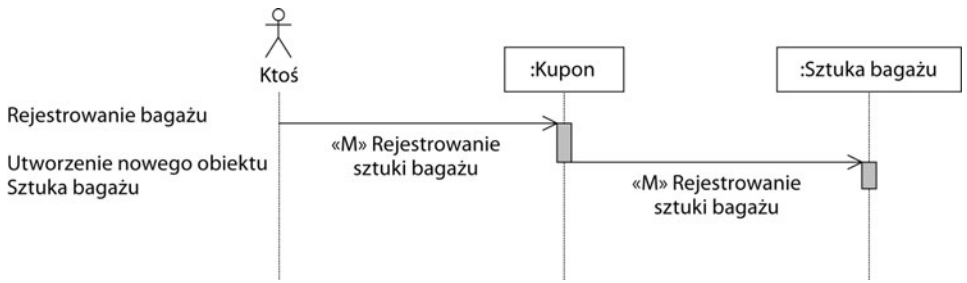
Wiemy już, że w naszym studium przypadku zdarzenie modyfikujące **«M» zarejestrowanie sztuki bagażu** modyfikuje obiekt klasy **kupon**. Nie istnieje jeszcze obiekt reprezentujący nową, rejestrowaną w tym zdarzeniu sztukę bagażu, zatem należy założyć, że obiektem początkowym będzie obiekt klasy **kupon**.

Propagacja zdarzeń — w jaki sposób zdarzenie jest przekazywane?

Rozpoczynając od obiektu początkowego, należy określić ścieżkę, którą będzie postępować modyfikacja obiektów w wyniku zdarzenia. Należy zatem odpowiedzieć na pytanie:

- Jaka jest kolejność modyfikowania poszczególnych obiektów z diagramu klas?

Zdarzenia modyfikujące propagują się zgodnie ze zdefiniowanymi w ramach diagramu klas asocjacjami obiektów modyfikowanych w wyniku zdarzenia. W diagramie klas naszego studium przypadku między obiektem klasy **kupon** a obiektem klasy **sztuka bagażu** istnieje asocjacja, za pośrednictwem której może propagować się zdarzenie modyfikujące. Zostało to przedstawione na rysunku 4.75.



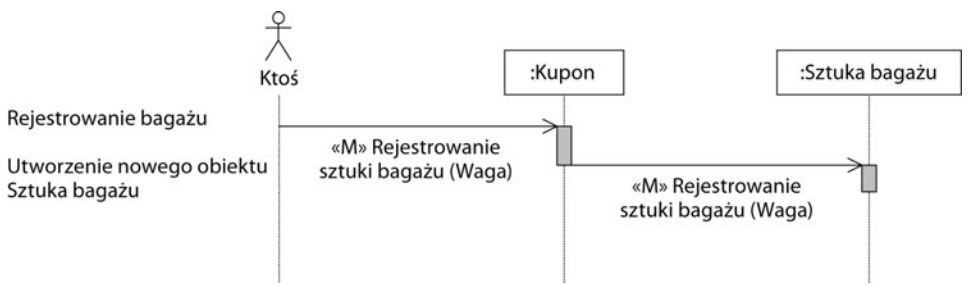
Rysunek 4.75. Zdarzenia w ramach diagramu sekwencji

Określenie parametrów zdarzenia — jakie informacje muszą znać obiekty?

Informacja niezbędna do przetwarzania zdarzenia modyfikującego musi być przekazana w postaci parametrów tego zdarzenia. Należy odpowiedzieć na pytanie:

- Jaka informacja na temat zdarzenia jest potrzebna obiektom do wykonania modyfikacji?

Aby utworzyć nowy obiekt klasy **sztuka bagażu**, potrzebujemy informacji o jego wadze. Waga musi więc być przesłana w postaci parametru zdarzenia, co przedstawia rysunek 4.76.



Rysunek 4.76. Parametr zdarzenia w ramach diagramu sekwencji

Weryfikacja diagramu sekwencji — czy wszystko jest wykonane prawidłowo?

Kompletny diagram sekwencji można zweryfikować za pomocą poniższej listy kontrolnej.

Lista kontrolna 4.11. Weryfikacja diagramu sekwencji w perspektywie interakcji

- Czy wszystkie klasy ulegające zmianom w wyniku zdarzenia modyfikującego są uwzględnione w diagramie sekwencji?
- Czy zdarzenie modyfikujące jest propagowane zgodnie z asocjacjami zdefiniowanymi w ramach diagramu klas?

Jeśli odpowiedzi na powyższe pytania brzmią „tak", oznacza to, że została wyeliminowana większość źródeł błędów.

5

ROZDZIAŁ

Modelowanie integracji systemów

Integracja systemów była przez długi czas najbardziej zaniedbaną dziedziną informatyki. Dopiero niedawno — wraz z rozwojem biznesów elektronicznych oraz metodologii Enterprise Application Integration (EAI) — problem ten zaczął cieszyć się większą uwagą. Mimo tego, iż integracja systemów informatycznych narodziła się z połączeniem wspólnym interfejsem pierwszych dwóch komputerów, dopiero ostatnie kilka lat przyniosło standaryzację w dziedzinach projektowania, metodologii i implementowania integracji. Niniejszy rozdział ilustruje, w jaki sposób można wykorzystać UML do modelowania komunikatów i procesów służących do wymiany tych komunikatów.

Integrację systemów rozumiemy jako **osadzanie** istniejących i nowych systemów informatycznych w istniejącym środowisku informatycznym. Osadzanie odbywa się w ramach organizacji, w której tworzy się interfejsy łączące te systemy informatyczne. Osadzanie może również obejmować kilka organizacji — w takim wypadku systemy informatyczne jednej organizacji łączy się z systemami informatycznymi innej. Rozstrzygnięcie kwestii, czy systemy informatyczne muszą być zintegrowane w ramach istniejącej infrastruktury i procesów w organizacji (ang. *in-house*), czy też w ramach struktur zewnętrznych, pozostaje drugorzędne z punktu widzenia modelowania.

Integracja systemu informatycznego wymaga wiedzy o środowisku systemu informatycznego oraz jego otoczeniu. Integrowany system informatyczny musi być osadzony w środowisku biznesowym, dlatego należy poznać otaczające go procesy biznesowe. Podstawą do tej analizy jest **model systemu biznesowego**, który skonstruowaliśmy i omówiliśmy w rozdziale 3. Aby system informatyczny współpracował z innymi systemami informatycznymi w sposób efektywny, należy skonstruować interfejsy do współpracujących z nim systemów w ramach organizacji, jak również do systemów spoza niej.

W tym rozdziale omówimy metodologię modelowania komunikatów wymienianych między różnymi systemami informatycznymi oraz procesy niezbędne do tej wymiany komunikatów.

5.1. Terminologia interfejsów integracji systemów

Interfejsy

Komunikacja między systemami informatycznymi odbywa się za pośrednictwem **interfejsów**. W związku z tym interfejs jest podstawowym elementem modelu integracji systemów. Za jego pośrednictwem system informatyczny (nadawca) przesyła informacje innemu systemowi informatycznemu (odbiorcy). Każdy z systemów informatycznych może być zarówno nadawcą, jak i odbiorcą.

Komunikaty

Systemy informatyczne połączone za pośrednictwem interfejsów wymieniają się **komunikatami**. Komunikat jest wysyłany przez system-nadawcę z oczekiwaniem, że w reakcji na niego natychmiast lub po ustalonym czasie oczekiwania system-odbiorca wykona określone działania (aktywności). Dla systemu-odbiorcy każdy otrzymany komunikat stanowi zdarzenie, na które powinien odpowiednio zareagować.

Załóżmy na przykład, że drogą elektroniczną przesyłana jest **informacja o fakturze** do zapłacenia przez odbiorcę. Po otrzymaniu faktury odbiorca ma określony czas na jej zapłacenie. System odbiorcy musi potwierdzić otrzymanie faktury, a także potencjalnie podjąć odpowiednie działania (na przykład w systemie finansowo-księgowym) zmierzające do wniesienia zapłaty.

Komunikaty mogą zawierać dodatkowe informacje niezbędne do zainicjowania aktywności systemu odbiorcy. Te informacje występują w postaci ustrukturalizowanych danych o określonej semantyce, takich jak dane faktury czy lista pasażerów. W metodyce UN/EDIFACT tego typu informacje określa się **danymi referencyjnymi** (ang. *reference data*).

Oprócz danych komunikaty zawierają informacje kontrolne, takie jak adres nadawcy i odbiorcy, metainformacje na temat zawartości komunikatu czy też sumy kontrolne. Informacje tego typu określa się czasem terminem **opakowania komunikatu** (ang. *packaging*).

Istnieje kilka metod przekształcania tych trzech elementów — zdarzenia, informacji oraz danych kontrolnych i referencyjnych — w gotowy komunikat, który można przesłać do innego systemu informatycznego. Informacje kontrolne można na przykład „ukrywać" w ramach programów interfejsu, mogą też być przesyłane jako dane dodatkowe. Zdarzenie może być przekazywane jako wywołanie procedury albo w postaci pliku zawierającego dane innych komunikatów.

W komunikatach w ramach metodyki UN/EDIFACT (nazywanych tu **dokumentami biznesowymi**) wszystkie trzy komponenty umieszcza się w jednym pakiecie wysyłkowym: zdarzenie (typ komunikatu), informacja (dane referencyjne) i informacje kontrolne (segmenty usługowe).

Komunikaty wysyłane w XML-u określa się również terminem dokumentów XML. Zawierają one przynajmniej dane referencyjne i metainformacje. Wielka zaleta XML-a w stosunku do UN/EDIFACT polega na tym, że komunikat XML zawiera odwołanie do opisu swojej struktury. Dzięki temu każdy odbiorca jest w stanie go odczytać.

Enterprise Application Integration

Enterprise Application Integration (**EAI**) zawiera metody, koncepcje i narzędzia do klasyfikacji, łączenia i koordynowania aplikacji w ramach organizacji. Celem tego standardu jest zapewnienie zintegrowanego przetwarzania informacji w ramach firmy za pomocą połączonych aplikacji — niezależnie od ich generacji i architektury. Taka sieć ulega stałym zmianom wskutek uaktualnień wersji i dodawania nowych aplikacji, a także dzięki unowocześnianiu technologii i wskutek innych czynników zewnętrznych.

Jednym z wymogów niezbędnych do osiągnięcia tego celu jest pozyskanie dokumentacji procesów biznesowych, sieci aplikacji, indywidualnych aplikacji i ich interfejsów, które powinny być w jak najszerszym zakresie zunifikowane pod względem wyglądu formularzy i innych elementów wizualnych.

UML jest prawie idealny do tego zadania, OMG zaimplementowała nawet specjalny profil — Enterprise Activation Integration — służący właśnie do tego celu. Objaśnione są tam zagadnienia implementacji technicznej tego typu instalacji w oparciu o UML.

Electronic Data Interchange

Termin **Electronic Data Interchange** (**EDI**) jest synonimem ustandaryzowanej wymiany dokumentów biznesowych (zamówienie, dostawa, faktura, informacja o transporcie itp.) między systemami informatycznymi organizacji i instytucji takich, jak dostawcy lub służby celne. Dokumenty biznesowe w tym standardzie również określa się mianem komunikatów.

Pod pojęciem EDI ukryte są obecnie liczne standardy, takie jak **Society for Worldwide Interbank Financial Telecommunication** (**SWIFT**), UN/EDIFACT i ANSI X12. Wraz z rozwojem internetu rozwinął się jakiś czas temu zupełnie nowy standard, XML. Wymiana danych między firmami jest jednak tylko jednym z aspektów XML-a, zaś możliwości tego standardu są znacznie szersze. Z tego powodu XML-a nie można postrzegać jako części EDI.

UN/EDIFACT

United Nations/Electronic Data Interchange for Administration, Commerce, and Transport (**UN/EDIFACT**) jest standardem międzynarodowym, opracowanym na potrzeby wymiany danych w administracji, ekonomii i transporcie. Standard ten definiuje reguły i typy komunikatów. Został opracowany przez **United Nations/Economic Commission for Europe** (**UN/ECE**), organizację działającą pod auspicjami ONZ i zajmującą się wsparciem handlu światowego za pomocą narzędzi elektronicznych. Odwołania do standardu UN/EDIFACT można znaleźć w wielu miejscach tego rozdziału. Mamy ku temu kilka ważnych powodów:

- UN/EDIFACT zawiera definicję umożliwiającą tworzenie skomplikowanych, hierarchicznych struktur komunikatów. Wszystkie trzy elementy komunikatu (zdarzenie, dane referencyjne i informacja kontrolna) zawarte są w pojedynczym pakiecie danych. Dzięki temu UN/EDIFACT koncepcyjnie jest jednym z najbardziej zaawansowanych standardów dotyczących integracji.
- Standard UN/EDIFACT jest dostępny w internecie bezpłatnie — wszystkie specyfikacje, regulacje i komunikaty wchodzące w jego skład można pobrać ze strony *http://www.unece.org/trade/untdid/welcome.htm*.

XML

Możliwości standardu **XML** (ang. **eXtensible Markup Language**) znacznie wykraczają poza zastosowania EDI. Integracja systemów jest zaledwie jednym z wielu obszarów zastosowań XML-a. W tym przypadku do jego wyróżnienia stosowany jest akronim XML/EDI. XML może być używany do integracji zarówno **w ramach jednej** organizacji, jak i **między różnymi** organizacjami.

Standard XML został opracowany i opublikowany przez **World Wide Web Consortium** (**W3C**: *http://www.w3.org*). Zawiera on w sobie inne standardy, takie jak **Extensible Stylesheet Language** (**XSL**) do opisu reprezentacji danych czy Xlink do standaryzacji odnośników. Z kolei XSL jest kolekcją zaleceń dotyczących przekształceń i formatowania dokumentów XML. XSL składa się z trzech części: XSLT, Xpath, i XSL-FO (zobacz *http://www.w3c.org/style/XSL*). Za pomocą eXML Linking Language w pojedynczym dokumencie mogą być łączone różne niezależne dokumenty. Zgodnie z oczekiwaniami te mechanizmy staną się w przyszłości podstawą narzędzi integracyjnych. XML rozwija się znacznie szybciej od UN/EDIFACT, a zastosowanie XML-a w integracji systemów zostało zaadaptowane metodą oddolną — najpierw nastąpiła szeroka adaptacja, a dopiero później rozpoczęto prace standaryzacyjne.

5.2. Komunikaty w UML-u

W diagramie sekwencji UML-a komunikaty są zobrazowane w postaci symbolu strzałki wraz z nazwą komunikatu i jego parametrami (o ile występują). Komunikaty w UML-u składają się z następujących elementów:

- Nazwa komunikatu określa **zdarzenie**.
- Argument komunikatu zawiera **informacje** uzupełniające, dzięki którym odbiorca może podjąć odpowiednie działania. W tej kategorii mieszczą się również informacje kontrolne.

Informacje przesyłane za pomocą komunikatu można nazwać **obiektem biznesowym**, jeśli spełnione są następujące warunki:

- informacja jest spójna;
- informacja jest ustrukturalizowana;

- komunikat jest dostosowany do określonego celu (na przykład zawiera dane faktury, listę pasażerów);
- komunikat jest samowystarczalny (nie wykorzystuje kluczy odwołań);
- jego ważność obejmuje indywidualne interakcje.

W modelu UML na potrzeby integracji systemów obiekty biznesowe są ustrukturalizowaną informacją przesyłaną od nadawcy do odbiorcy w charakterze argumentu komunikatu.

5.3. Jeden model — dwie perspektywy

Na potrzeby integracji systemów informatycznych musimy zdefiniować, **które** informacje mają być wymieniane, a także **sposób** ich wymiany. Z tego powodu **model integracji systemów** składa się z dwóch perspektyw — **perspektywy procesu** i **perspektywy statycznej**.

W procesie integracji systemów główną uwagę skupia się na etapach procesu o kluczowym znaczeniu dla interakcji, czyli na wymianie komunikatów między systemami informatycznymi lub innymi elementami biorącymi udział w tej integracji. Czasem opisuje się również procesy, które nie biorą bezpośredniego udziału w wymianie komunikatów, lecz są ważne dla pełnego zrozumienia całego procesu. Przykładem takiego procesu może być wyładowanie bagażu pasażerów, którzy nie wsiedli na pokład (akcja **wyładunek bagaży** związana z aktywnością **rezygnacji z wejścia na pokład**).

Perspektywa statyczna opisuje **zawartość** i **strukturę** obiektów biznesowych podlegających wymianie między partnerami.

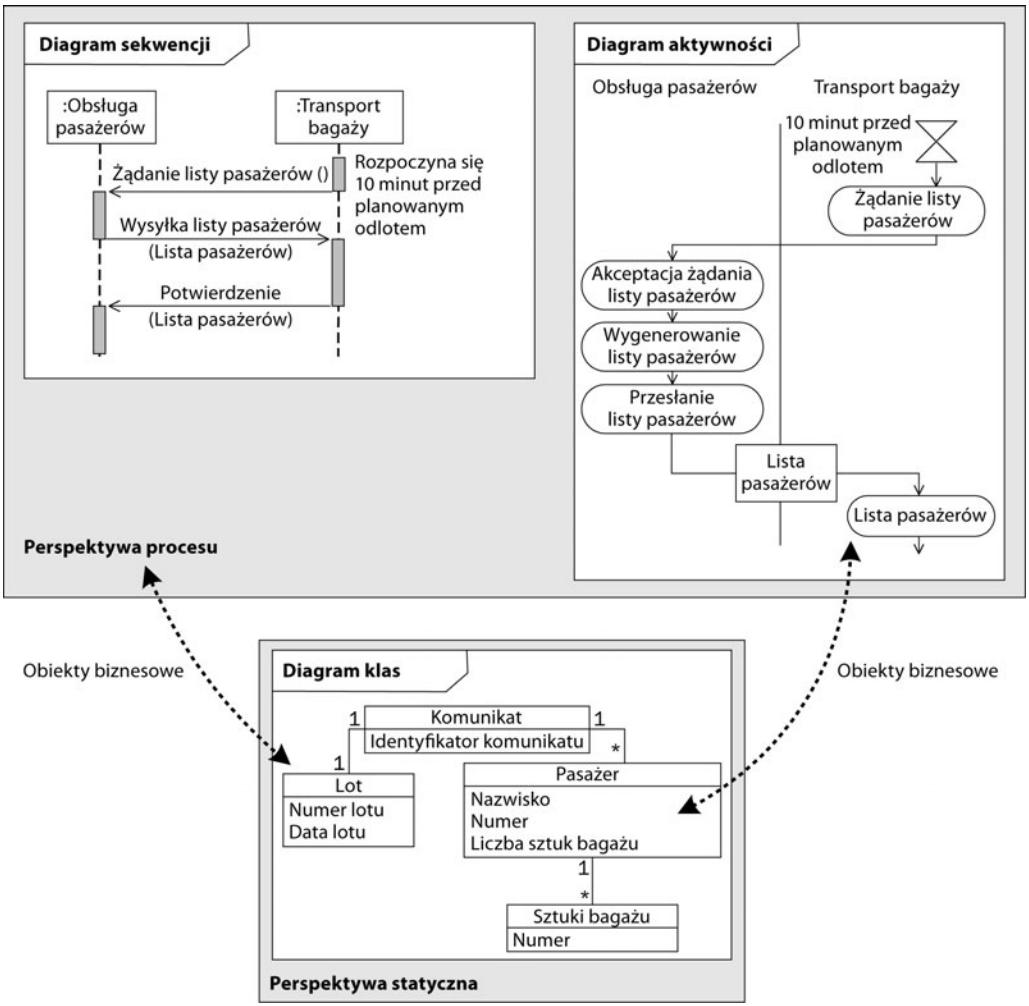
Perspektywy wykorzystywane w modelu integracji systemów oraz poszczególne diagramy UML wchodzące w skład tych perspektyw są pokazane na rysunku 5.1.

5.4. Perspektywa procesu

Perspektywa procesu obrazuje **aktywności** podejmowane przez system informatyczny podczas wymiany komunikatów z innymi systemami informatycznymi. Nie chodzi tu jednak o czysto techniczne procesy niezbędne do komunikacji między systemami informatycznymi, takie jak nawiązanie połączenia modemowego lub internetowego.

Przy modelowaniu elementów perspektywy procesu przydatne będą odpowiedzi na następujące pytania:

- Które aktywności będą brane pod uwagę w perspektywie procesu? Które interakcje zidentyfikowane w naszym studium przypadku, zachodzące między systemem obsługi pasażerów a innymi systemami, mają znaczenie w tej perspektywie?
- Czy te aktywności są interesujące wyłącznie z punktu widzenia integracji systemów?
- Czy te aktywności zostały już opisane w którymkolwiek z modeli i jego perspektywach?

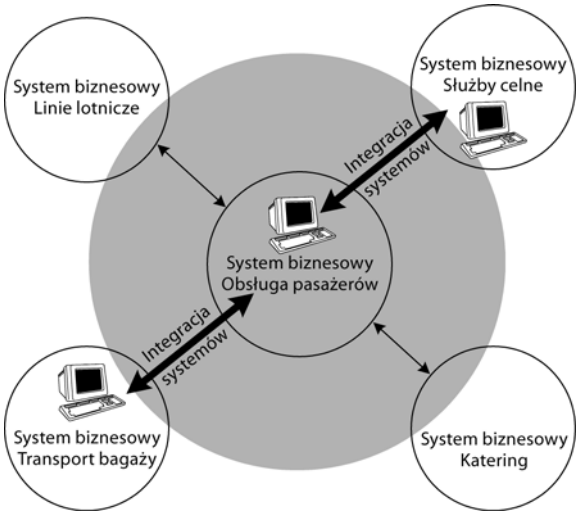


Rysunek 5.1. Perspektywa procesu i perspektywa statyczna w modelu integracji systemów

5.4.1. Model systemu biznesowego jako fundament

Wymiana komunikatów między systemami informatycznymi odbywa się w oparciu o **zdarzenia biznesowe**. Z tego powodu jest ona aktywnością **procesów biznesowych**.

W **modelu systemu biznesowego** naszego studium przypadku będziemy pracować nad integracją systemu informatycznego **obsługi pasażerów** (rysunek 5.2).



Rysunek 5.2. Integracja systemów informatycznych

Systemy informatyczne wymagające integracji należy wyodrębnić z procesów biznesowych zachodzących w ramach oraz wokół systemu biznesowego obsługi pasażerów. Jako fundament diagramów integracji mogą posłużyć diagramy wchodzące w skład poszczególnych perspektyw modelu biznesowego.

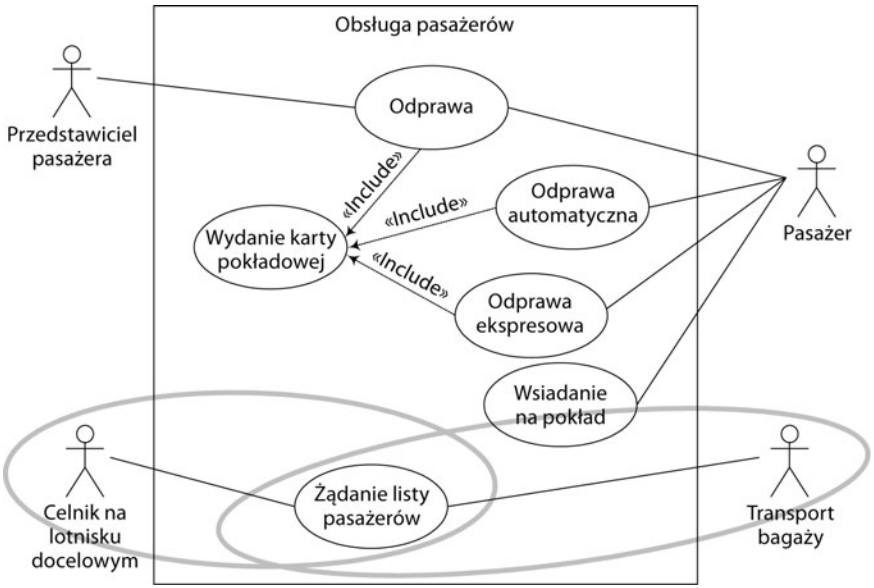
W celu modelowania integracji systemów informatycznych należących do **tego samego** systemu biznesowego bierzemy pod uwagę diagramy **perspektywy wewnętrznej** systemu biznesowego.

W celu modelowania interakcji między systemami informatycznymi **z różnych** systemów biznesowych wykorzystujemy diagramy **perspektywy zewnętrznej** systemu biznesowego. Nie ma znaczenia, czy istniejący system informatyczny faktycznie należy do tej samej organizacji. Szczegóły procesu najlepiej omówić w oparciu o diagramy perspektywy wewnętrznej systemu biznesowego.

Z naszego studium przypadku wybraliśmy dwa interfejsy. Oto one:

- **Lista pasażerów na potrzeby urzędu celnego**. Każdemu lotowi międzynarodowemu towarzyszy lista pasażerów przekazywana do urzędu celnego lotniska docelowego. Na tej liście wymieniony jest każdy pasażer i członek załogi samolotu. Przesyłanie danych odbywa się między startem samolotu a lądowaniem. Dzięki temu służby celne na lotnisku docelowym mają możliwość weryfikacji danych i czas niezbędny na podjęcie decyzji dotyczących odprawy celnej pasażerów i załogi.
- **Rezygnacja z wejścia na pokład**. Dziesięć minut przed **planowanym czasem odlotu** rozpoczyna się procedura **rezygnacji z wejścia na pokład**. W praktyce dziesięć minut przed odlotem uruchamiany jest licznik czasu. W tym momencie komórka transportu bagaży musi otrzymać listę pasażerów. Od strony obsługi pasażerów to zdarzenie (uruchomienie licznika czasu) inicjuje wygenerowanie listy pasażerów, którzy nie wsiedli na pokład. Po wyładowaniu bagaży pasażerów z tej listy obsługa otrzymuje z działu transportu bagaży listę pasażerów z odpowiednimi potwierdzeniami.

Rysunek 5.3 prezentuje diagram przypadku użycia z rozdziału 3. (rysunek 3.14), który posłuży nam jako podstawa diagramu dwóch opisanych wyżej przypadków użycia (zaznaczone na diagramie).



Rysunek 5.3. Diagram przypadków użycia systemu biznesowego „obsługa pasażerów"

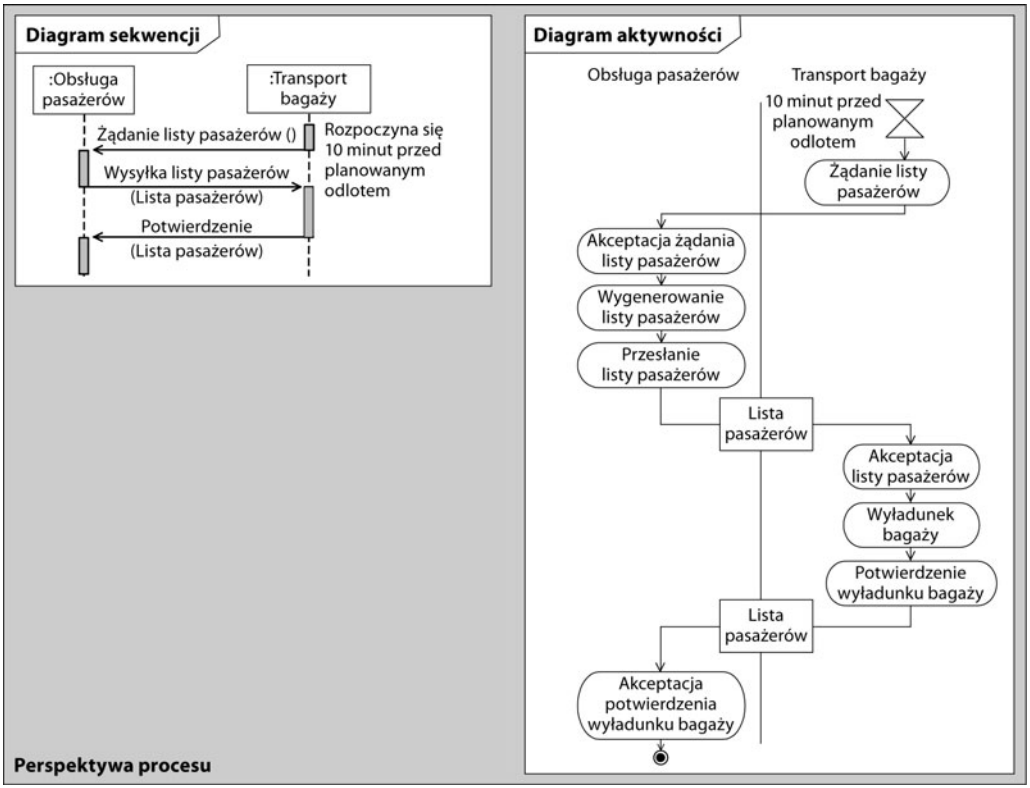
Obydwa przykłady wiążą się z koniecznością wdrożenia aktywności z przypadku użycia **żądanie listy pasażerów**. W pierwszym przykładzie jest to przesłanie listy pasażerów na potrzeby służb celnych, co stanowi interakcję między systemem informatycznym **obsługi pasażerów** a aktorem **celnik na lotnisku docelowym** (zaznaczono go na diagramie przypadku użycia).

Drugi przykład opisuje interakcję między systemem biznesowym **obsługa pasażerów** a aktorem **transport bagaży**, również oznaczonym na diagramie przypadku użycia.

Do opisu perspektywy procesu między zaangażowanymi systemami informatycznymi wykorzystujemy diagram aktywności i diagram sekwencji. W celu objaśnienia diagramu aktywności i diagramu sekwencji oraz sposobu odczytywania diagramów wykorzystamy przykład z listą pasażerów dla służb celnych. Do omówienia konstruowania diagramów posłużymy się przykładem zdarzenia **rezygnacji z wejścia na pokład**.

5.4.2. Elementy perspektywy

Aktywności, które muszą zostać wykonane w celu wymiany komunikatów między systemami informatycznymi, można zilustrować za pomocą diagramów sekwencji i diagramów aktywności (zobacz rysunek 5.4).



Rysunek 5.4. Perspektywa procesu

- **Diagram aktywności** opisuje przepływ działań. Obrazuje on zależności między indywidualnymi akcjami oraz przepływ obiektów biznesowych.
- **Diagram sekwencji** obrazuje chronologiczną kolejność wymiany komunikatów między systemami informatycznymi.

5.4.3. Diagramy aktywności

W punkcie 3.3.5, „Diagramy aktywności" omówiliśmy podstawowe elementy diagramów aktywności. W tym miejscu przejdziemy do specjalnych interpretacji i zastosowań tych diagramów na potrzeby integracji systemów.

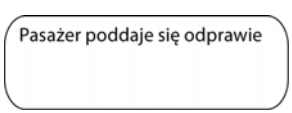
Z diagramu aktywności z rysunku 5.5 można wydobyć podstawowe informacje na temat integracji systemów informatycznych w procesie **rezygnacji z wejścia na pokład**. Na przykład możemy dowiedzieć się czegoś na temat obiektów biznesowych ulegających wymianie.

Rysunek 5.5. Diagram aktywności „rezygnacja z wejścia na pokład"

Aktywność

W diagramach aktywności pojedyncza aktywność jest obrazowana w sposób przedstawiony na rysunku 5.6. W omawianym kontekście aktywność reprezentuje proces biznesowy. Elementami aktywności są akcje, elementy kontrolne (decyzje, rozgałęzienia, złączenia, początek, koniec itp.) oraz obiekty.

Wymienione elementy są połączone za pomocą krawędzi (ang. *edge*). Połączone akcje i elementy kontrolne tworzą przepływ sterowania nazywany po prostu **przepływem**.



Przepływ obiektów reprezentuje ścieżkę obiektów w ramach aktywności. Można go pominąć przy tworzeniu diagramów aktywności. Aktywność może obejmować kilka równoległych przepływów.

Rysunek 5.6. Elementy diagramu aktywności

Przepływ obiektów (krawędź)

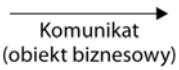
Krawędzie obrazowane w postaci strzałek łączą poszczególne elementy diagramu aktywności i reprezentują przepływ sterowania oraz przepływ obiektów (krawędź) aktywności. Przepływ sterowania determinuje przepływ w ramach aktywności.

Strzałka wchodząca rozpoczyna kolejny etap aktywności. Po zakończeniu danego etapu przepływ postępuje zgodnie z kolejną strzałką (wychodzącą). Przepływ obiektów opisuje sposób przekazywania obiektów i danych w ramach aktywności. Krawędzie mogą być opisane za pomocą etykiet z nazwą (w pobliżu strzałki).

Przepływ obiektów na diagramie aktywności prezentuje ścieżkę przesyłania jednego lub większej liczby obiektów pomiędzy różnymi aktywnościami.

Akceptacja sygnału (akcja)

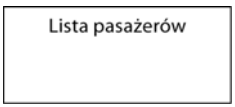
Sygnał jest wysyłany do określonej aktywności:



Aktywność otrzymująca sygnał z akcją **akceptacji sygnału** reaguje na niego, podejmując akcję akceptacji; może też zareagować w sposób zgodny z przepływem w danym węźle diagramu aktywności.

Obiekt biznesowy

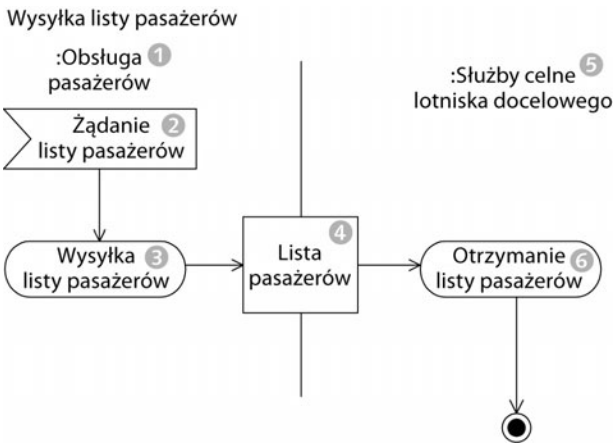
Obiekt biznesowy składa się z ustrukturalizowanych danych wymienianych między akcjami (zobacz podrozdział 5.2, „Komunikaty w UML-u"). Obiekt biznesowy jest wyjściem z jednej akcji, a jednocześnie wejściem do innej:



Obiekt biznesowy opuszczający partycję jednej aktywności jest przesyłany z jednego systemu informatycznego do innego.

Czytanie diagramów aktywności

Diagramy aktywności obrazują interakcje między różnymi systemami informatycznymi zaangażowanymi w wymianę komunikatów. Rysunek 5.7 przedstawia system informatyczny **obsługi pasażerów** (1), który inicjuje akcje przesłania **listy pasażerów** (3) wskutek otrzymania zdarzenia **żądanie listy pasażerów** (2). Obiekt biznesowy **lista pasażerów** (4) jest przesyłany do systemu informatycznego **służb celnych lotniska docelowego** (5). System informatyczny służb celnych lotniska docelowego akceptuje **listę pasażerów** (4) za pomocą zdarzenia **otrzymano listę pasażerów** (6).

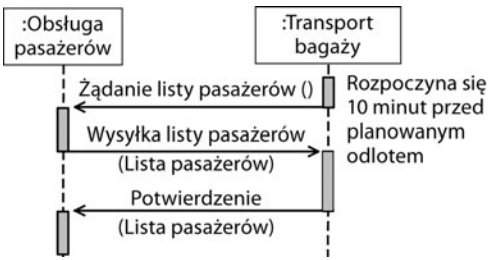


Rysunek 5.7. Diagram aktywności

Z tego diagramu nie wynika, że lista pasażerów jest przesyłana jako argument komunikatu. Aby to zobaczyć, musimy posłużyć się diagramem sekwencji.

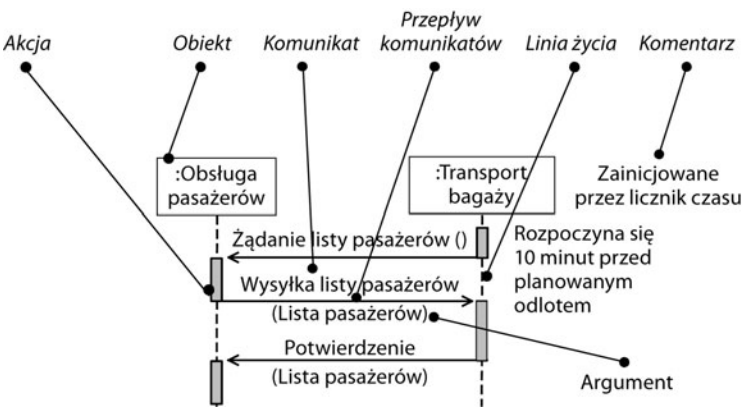
5.4.4. Diagramy sekwencji

Uwaga w diagramach sekwencji skupiona jest na zilustrowaniu chronologicznej sekwencji wymiany komunikatów między obiektami. Przykład takiego diagramu przedstawia rysunek 5.8.



Rysunek 5.8. Diagram sekwencji „rezygnacji z wejścia na pokład"

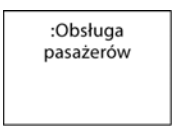
W diagramach sekwencji z rysunku 5.9 mamy do czynienia z następującymi elementami:



Rysunek 5.9. Elementy diagramu sekwencji

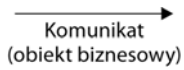
Obiekt

Obiekty wymieniają się komunikatami. W modelu integracji systemów obiekty te reprezentują komunikujące się systemy informatyczne:



Komunikat

W diagramach sekwencji komunikaty są rozumiane jako działania zdarzeń. Informacja jest przekazywana w argumentach komunikatów:



Argumentami komunikatów mogą być obiekty biznesowe (zobacz podrozdział 5.2, „Komunikaty w UML-u").

Przepływ komunikatów

Przepływ komunikatów odbywa się w kierunku od nadawcy do odbiorcy. W modelu integracji systemów przepływ komunikatów diagramu sekwencji odpowiada przepływowi obiektów diagramu aktywności:

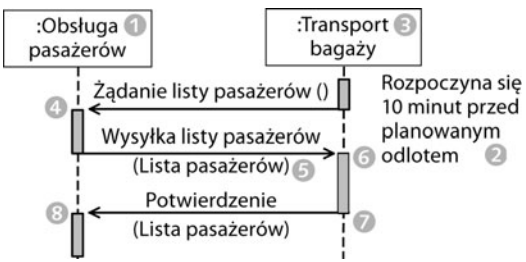
Różnica polega na tym, że w diagramie sekwencji dodatkowym aspektem jest chronologia interakcji.

Argument

Zobacz hasło *Komunikat*.

Czytanie diagramów sekwencji

Rysunek 5.10 obrazuje następującą sytuację: gdy tylko spełniony jest wymóg (2), transport bagaży wysyła do obsługi pasażerów (1) żądanie przekazania listy pasażerów. Obsługa pasażerów akceptuje (4) to żądanie, generuje listę pasażerów i odsyła ją do transportu bagaży (6).



Rysunek 5.10. Diagram sekwencji

W oparciu o listę pasażerów (5), którą otrzymali (6) pracownicy transportu bagaży (3), z samolotu wyładowywane są odpowiednie sztuki bagażu. Po wyładowaniu bagaży obsługa transportu bagaży odsyła obsłudze pasażerów potwierdzenie wykonania tej operacji wraz z listą pasażerów, których bagaże zostały wyładowane (7). Obsługa pasażerów potwierdza (8) otrzymanie informacji o wyładowaniu bagaży. W diagramie sekwencji nie ma możliwości stwierdzenia, jakie

działania są podejmowane w ramach wymiany tych komunikatów. Te informacje są zawarte w diagramie aktywności (rysunek 5.11). Informacje o poszczególnych akcjach mogą również być umieszczane w diagramach sekwencji w postaci komentarzy. Takie podejście niesie jednak ryzyko utraty czytelności diagramu sekwencji. W przeciwieństwie do diagramu aktywności z diagramu sekwencji jasno wynika, że obiekt biznesowy **lista pasażerów** jest przesyłany w charakterze argumentu komunikatu.

Rysunek 5.11. Aktywność „rezygnacja z wejścia na pokład"

5.4.5. Konstruowanie diagramów w perspektywie procesów

W charakterze przykładu do konstrukcji diagramów wykorzystamy interfejs do systemu transportu bagaży z naszego studium przypadku.

Poniższa lista kontrolna zawiera etapy niezbędne przy konstruowaniu diagramów aktywności i diagramów sekwencji w perspektywie procesu.

Lista kontrolna 5.1. Konstruowanie diagramów w perspektywie procesu

- Określenie interfejsów — pomiędzy jakimi systemami informatycznymi odbywa się komunikacja?
- Identyfikacja zaangażowanych systemów — które systemy informatyczne wymieniają się informacjami?
- Identyfikacja aktywności i przepływów — co należy wykonać i kto jest za to odpowiedzialny?
- Zdefiniowanie komunikatów — jakie komunikaty będą wymieniane?
- Zdefiniowanie reguł — od czego są uzależnione podejmowane akcje?
- Weryfikacja perspektywy — czy wszystko zostało wykonane prawidłowo?

Określenie interfejsów — pomiędzy jakimi systemami informatycznymi odbywa się komunikacja?

Aktywność **rezygnacja z wejścia na pokład** (rysunek 5.11) wymaga wymiany informacji (interakcji) między systemem informatycznym obsługi pasażerów a innymi systemami informatycznymi, co można odczytać z modelu systemu biznesowego (zobacz rozdział 3.). W naszym studium przypadku informacje te można znaleźć w przypadku użycia **żądanie listy pasażerów**. Zawiera on interfejsy do systemu transportu bagaży oraz systemu służb celnych lotniska docelowego. Ten etap tworzenia modelu omówimy w oparciu o integrację z systemem transportu bagaży (rysunek 5.3).

Z aktywności „rezygnacji z wejścia na pokład" (rysunek 5.11) wybieramy akcje związane z wymianą komunikatów, takie jak:

- żądanie listy pasażerów;
- akceptacja żądania listy pasażerów;
- wygenerowanie listy pasażerów;
- wysyłka listy pasażerów;
- akceptacja listy pasażerów;
- potwierdzenie wyładunku bagaży;
- akceptacja potwierdzenia wyładunku bagaży.

Z diagramu sekwencji modelu biznesowego przedstawionego na rysunku 5.8 możemy wyczytać dodatkowo, że wymianie ulegają następujące komunikaty:

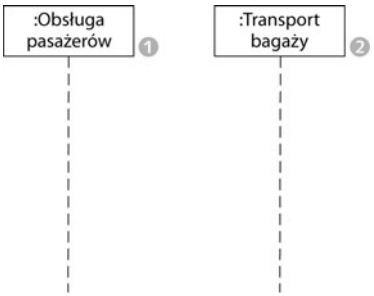
- **wysyłka listy pasażerów** (lista pasażerów);
- **potwierdzenie** (lista pasażerów).

Identyfikacja zaangażowanych systemów — które systemy informatyczne wymieniają się informacjami?

Aby komunikaty mogły być przesłane i otrzymane, należy zidentyfikować systemy informatyczne biorące udział w interakcji oraz role, jakie w niej odgrywają:

- Które systemy informatyczne są niezbędne podczas realizacji systemów biznesowych?

W naszym studium przypadku jest to z pewnością system informatyczny **obsługa pasażerów** (1). Systemy informatyczne spoza systemu biznesowego można zidentyfikować na podstawie aktorów biorących udział w przypadkach użycia modelu biznesowego. Dzięki temu znaleźliśmy system informatyczny **transport bagaży** (2), co przedstawia rysunek 5.12.

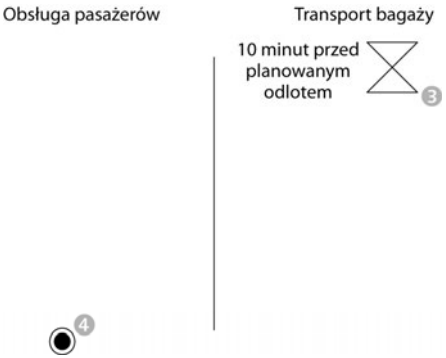


Rysunek 5.12. Konstruowanie diagramu sekwencji

- Które systemy informatyczne inicjują proces?

  W naszym przypadku inicjatorem procesu jest **transport bagaży** (2). Niezależnie od procedury wprowadzania na pokład przeprowadzanej w ramach obsługi pasażerów licznik czasu (nastawiony na 10 minut przed szacowanym czasem startu) inicjuje wysłanie żądania przekazania listy pasażerów, którzy nie wsiedli na pokład, aby przeprowadzić operację wyładunku ich bagaży.
- Który system informatyczny znajduje się na końcu procesu? W naszym studium przypadku aktywność **rezygnacja z wejścia na pokład** kończy się w momencie, gdy system **obsługa pasażerów** (1) otrzyma z systemu **transportu bagaży** komunikat zawierający listę pasażerów, których bagaże zostały wyładowane z samolotu.

Wyposażeni w te informacje możemy przystąpić do konstruowania diagramu aktywności (rysunek 5.13).



Rysunek 5.13. Konstruowanie diagramu aktywności

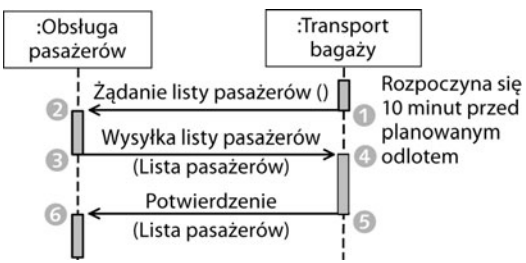
Identyfikacja aktywności i przepływów — co należy wykonać i kto jest za to odpowiedzialny?

Podczas identyfikacji akcji przepływu sterowania pomocne będą odpowiedzi na następujące pytania:

- Jakie działania muszą zostać wykonane, aby systemy informatyczne mogły wymieniać się komunikatami?
- W jakiej kolejności przetwarzane są aktywności?
- Czy występują akcje odbywające się równocześnie z innymi?
- Jakie warunki muszą zostać spełnione w celu wykonania akcji?
- Czy wszystkie wcześniejsze akcje muszą być zakończone, aby mogły rozpocząć się akcje następujące po nich?
- Kto jest odpowiedzialny za wykonanie akcji? Do której partycji należy każda z nich?

Akcja…	…jest wykonywana przez aktora
Żądanie listy pasażerów (1)	Transport bagaży
Akceptacja żądania listy pasażerów (2)	Obsługa pasażerów
Przesłanie listy pasażerów (3)	Obsługa pasażerów
Akceptacja listy pasażerów (4)	Transport bagaży
Potwierdzenie wyładunku bagaży (5)	Transport bagaży
Akceptacja potwierdzenia wyładunku bagaży (6)	Obsługa pasażerów

Te informacje są udokumentowane na diagramie sekwencji przedstawionym na rysunku 5.14 oraz na diagramie aktywności z rysunku 5.15.

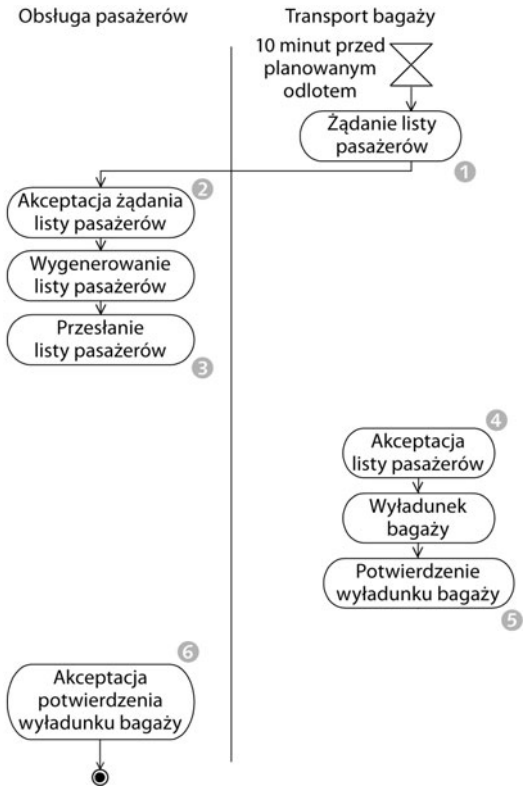


Rysunek 5.14. Konstruowanie diagramu sekwencji

Zdefiniowanie komunikatów — jakie komunikaty będą wymieniane?

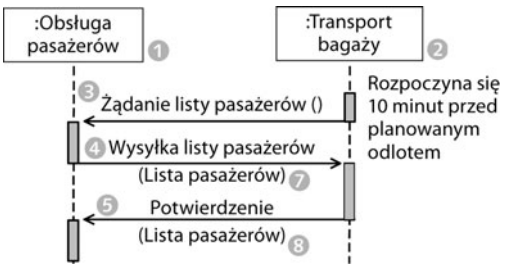
Z modelu systemu biznesowego wynika, że między systemami przesyłane są następujące dwa komunikaty:

- **wysyłka listy pasażerów** (lista pasażerów);
- **potwierdzenie** (lista pasażerów).



Rysunek 5.15. Konstruowanie diagramu aktywności

W pierwszym komunikacie **wysyłka listy pasażerów** (rysunek 5.16) do systemu informatycznego **transport bagaży** (2) przesyłana jest **lista pasażerów** (7) zawierająca nazwiska pasażerów, którzy nie wsiedli na pokład.



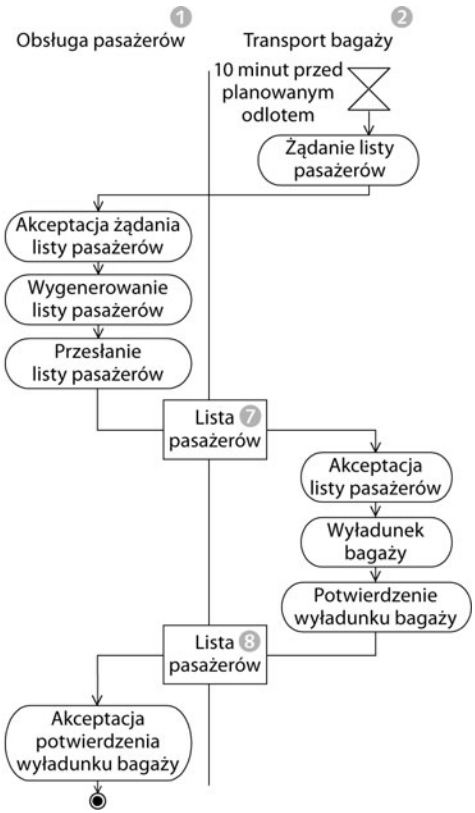
Rysunek 5.16. Konstruowanie diagramu sekwencji

Po wyładowaniu wszystkich sztuk bagażu **obsługa pasażerów** (1) otrzymuje **potwierdzenie** (5). Wraz z potwierdzeniem **lista pasażerów** (8) jest odsyłana do **obsługi pasażerów** (1).

Obydwa komunikaty jako argument przyjmują **listę pasażerów** (7) (8). W naszym przypadku lista ta była przekazywana z systemu **obsługi pasażerów** (1) do **transportu bagaży** (2), tam modyfikowana, po czym odsyłana z powrotem do **obsługi pasażerów** (1).

Na końcu lista pasażerów nie jest dokładnie taka sama, jaka była na początku aktywności. Jednakże **struktura** obiektu biznesowego **lista pasażerów** pozostaje bez zmian.

Jak wynika z rysunku 5.17, przepływ obiektów odbywa się zgodnie z przepływem sterowania, przesłaniając go.



Rysunek 5.17. Konstruowanie diagramu aktywności

Zdefiniowanie reguł — od czego są uzależnione podejmowane akcje?

Wymiana komunikatów między organizacjami podlega umowom uzgadniającym kwestie odpowiedzialności, organizacyjne itp., czyli zagadnienia współpracy. Równie ważny wpływ na wymianę komunikatów mają umowy i uzgodnienia międzynarodowe. Zarówno od strony technicznej, jak i prawnej kontrolą reguł wymiany danych zajmują się komisje standaryzacyjne. Jako przykład można podać specyfikację *Collaboration-Protocol Profile and Agreement Specification Version 2.0* autorstwa ebXML, dostępną pod adresem *http://ebxml.org/specs/ebcpp-2.0.pdf*.

Innym przykładem jest tu organizacja **International Air Transport Association** (**IATA**), która zajmuje się między innymi definiowaniem komunikatów i ich zastosowań w przemyśle lotniczym. Podobne regulacje stają się coraz bardziej powszechne również w innych branżach. Na tym etapie prac należy odpowiedzieć na następujące pytanie:

- Jakie ustalenia, kontrakty i inne regulacje prawne i techniczne należy wziąć pod uwagę przy definiowaniu wymiany danych?

Weryfikacja perspektywy — czy wszystko zostało wykonane prawidłowo?

Kompletne diagramy sekwencji i diagramy aktywności można weryfikować, wykorzystując poniższe listy kontrolne.

Lista kontrolna 5.2. Weryfikacja diagramów sekwencji w perspektywie procesu

- Czy wszystkie komunikaty wymagające potwierdzenia są wymienione w diagramie sekwencji z adnotacją o potwierdzeniu?
- Czy każdy system informatyczny i partner biznesowy zaangażowany w wymianę komunikatów jest uwzględniony na przynajmniej jednym diagramie sekwencji?
- Czy wszystkie przesyłane obiekty biznesowe są ujęte w diagramach sekwencji w postaci argumentów komunikatów?
- Czy wszystkie istotne komentarze są umieszczone na diagramach?
- Czy na diagramach nie ma zbyt dużej liczby komentarzy, co wpływa na obniżenie ich czytelności?
- Czy przepływ komunikatów odpowiada przepływowi obiektów na diagramie aktywności?

Lista kontrolna 5.3. Weryfikacja diagramów aktywności w perspektywie procesu

- Przy konstruowaniu diagramów aktywności w perspektywie procesu należy pamiętać, że znaczenie mają tu jedynie te przepływy, które biorą udział w wymianie komunikatów. Procesy ściśle biznesowe należą wyłącznie do modelu biznesowego.
- Czy przepływ obiektów jest czytelny i oczywisty?
- Czy wszystkie obiekty biznesowe są ujęte na diagramie aktywności?
- Warunki poszczególnych wyjść z rozgałęzienia nie powinny nakładać się na siebie. W przeciwnym wypadku przepływ sterowania jest niejednoznaczny, co oznacza, że nie jest oczywiste, którą drogą postępuje w węźle decyzyjnym.
- Warunki muszą uwzględniać wszystkie możliwości. W przeciwnym wypadku przepływ sterowania może ugrzęznąć. W przypadku wątpliwości należy zastosować dodatkowe wyjście z warunkiem **else**.
- Rozgałęzienia i złączenia powinny być zrównoważone. Liczba przepływów wychodzących z rozgałęzienia powinna odpowiadać liczbie przepływów wchodzących do odpowiedniego złączenia.

5.5. Perspektywa statyczna

Perspektywa statyczna opisuje strukturę obiektów biznesowych przesyłanych w charakterze argumentów komunikatów od nadawcy do odbiorcy (zobacz też podrozdział 5.2, „Komunikaty w UML-u").

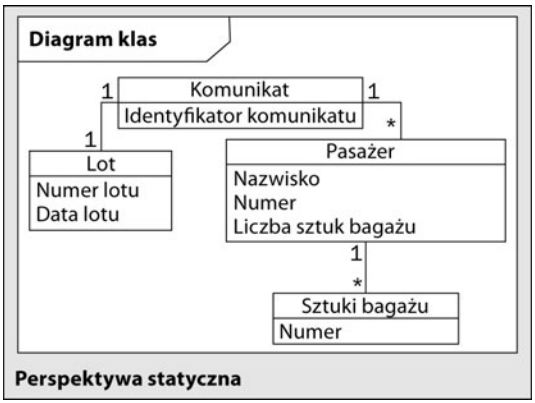
Podczas modelowania obiektów biznesowych należy wziąć pod uwagę następujące zagadnienia:

- Bardzo ważne jest zapewnienie **integralności semantycznej**. Struktura i zawartość obiektów biznesowych muszą być czytelne i łatwe do zrozumienia dla wszystkich osób uczestniczących w modelowaniu.
- Informacje muszą być spójne i wszyscy uczestnicy prac powinni umieć właściwie je interpretować.
- Obiekty biznesowe powinny nadawać się do wielokrotnego wykorzystania.
- Obiekty biznesowe muszą być kompletne, ponieważ muszą spełniać nawet rzadko pojawiające się wymagania i nie może być tu miejsca dla wieloznaczności.

Różne wymagania i podejścia do modelowania prowadzą do powstawania różnych struktur obiektów biznesowych. Niemożliwe jest spełnienie wszystkich wymagań jednocześnie. Struktura i zakres obiektów biznesowych **zawsze** muszą stanowić kompromis — nie ma możliwości opracowania idealnego obiektu biznesowego.

5.5.1. Elementy perspektywy

Do zilustrowania obiektów biznesowych w perspektywie statycznej modelu integracji systemów posłużymy się diagramem klas przedstawionym na rysunku 5.18.



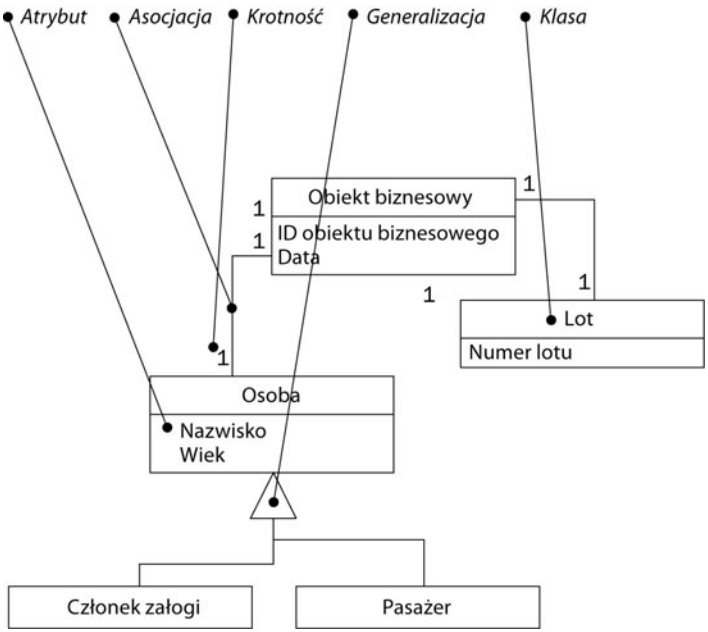
Rysunek 5.18. Perspektywa statyczna

5.5.2. Diagram klas

Szersze omówienie modelowania klas można znaleźć w podrozdziale 4.2, „Perspektywa strukturalna". Poniżej omówimy jedynie te zagadnienia, które są specyficzne dla modelowania integracji systemów.

Klasa

Klasa w diagramie klas (rysunek 5.19) modelu integracji systemów reprezentuje spójny zbiór informacji, z których składa się obiekt biznesowy.



Rysunek 5.19. Diagram klas dla modelu biznesowego

Czytanie diagramów klas

Czytane diagramów klas dla obiektów biznesowych nie różni się niczym od czytania zwykłych diagramów klas w modelu systemu informatycznego. Więcej informacji na ten temat można znaleźć w rozdziale 4., w punkcie 4.2.5, „Diagram klas".

5.5.3. Konstruowanie diagramu klas

Poniższa lista kontrolna przedstawia etapy niezbędne do konstruowania diagramów klas dla obiektów biznesowych. Każdy z tych etapów omawiamy poniżej.

Lista kontrolna 5.4. Konstruowanie diagramów klas w perspektywie statycznej

- Zgromadzenie informacji niezbędnych dla obiektu biznesowego — co chcemy przesyłać?
- Skonstruowanie diagramu klas — jaka jest struktura obiektu biznesowego?
- Adaptacja klas i atrybutów z diagramu klas systemu informatycznego — jakie informacje mają być uwzględnione na diagramie klasy?
- Pozyskanie innych elementów danych — skąd wziąć pozostałe informacje?
- Zdefiniowanie klas i związków obiektu biznesowego — jakie związki klas będą potrzebne?
- Weryfikacja perspektywy — czy wszystko zostało wykonane prawidłowo?

Zgromadzenie informacji niezbędnych dla obiektu biznesowego — co chcemy przesyłać?

W podejściu zstępującym podstawę modelowania obiektów biznesowych stanowią procesy biznesowe. Następnie definiuje się **informacje**, które muszą podlegać wymianie między systemami w celu realizacji tych procesów biznesowych.

Na podstawie informacji podlegających wymianie opracowujemy następnie strukturę **obiektów biznesowych**. Zaletą tego podejścia jest to, że specyfika systemu informatycznego nie warunkuje tu wyników modelowania. Modelowane obiekty biznesowe pozostają niezależne od systemu informatycznego, a dzięki temu nadają się do ponownego wykorzystania. Jednakże należy pamiętać, że powiązanie takich modeli z konkretnym systemem informatycznym zawsze wymaga dodatkowych nakładów pracy.

Podejście zstępujące skutkuje utworzeniem obiektów biznesowych, które można wykorzystać w wielu różnych systemach informatycznych. Ustandaryzowane komunikaty EDI dotyczące wymiany danych między firmami są modelowane właśnie zgodnie z tą metodologią.

Jeśli jako punkt wyjścia w modelowaniu obiektów biznesowych przyjmie się istniejący system informatyczny, jako podstawę wykorzystuje się klasy systemu informatycznego. Nie zawsze jednak udaje się uzyskać wszystkie informacje z systemu informatycznego. Informacje znaczące z punktu widzenia systemu biznesowego nie zawsze mają znaczenie z punktu widzenia systemu informatycznego. Przykładem takiej sytuacji może być liczba sztuk odprawianego bagażu. W obiekcie biznesowym ta informacja jest przypisana elementowi kontrolnemu, w systemie informatycznym jest ona zupełnie bez znaczenia. Jak zatem widać, diagram klas systemu informatycznego pokazuje informacje z innej perspektywy niż diagram klas dla obiektu biznesowego.

Z drugiej strony, wewnętrzne identyfikatory obiektów — kluczowe w systemie informatycznym — nie mają zastosowania w obiekcie biznesowym, niezależnym wszak od konkretnego systemu informatycznego.

Gdy w obiektach biznesowych ma znaczenie utrzymanie określonego standardu, stosowane są standardowe komunikaty z odpowiedniego katalogu. Więcej informacji na ten temat można znaleźć w serwisie UN/ECE (*http://www.unece.org*), na stronie ebXML (*http://wwwebxml.org*) oraz w wielu innych źródłach informacji odpowiednich dla różnych gałęzi przemysłu i gospodarki. Nawet wtedy, gdy nie ma konieczności zastosowania ustandaryzowanych komunikatów, warto rzucić okiem na tego typu katalogi. Zawartość standardowych komunikatów może pomóc w podejmowaniu decyzji projektowych i zasugerować właściwe podejście do tworzenia obiektów biznesowych.

Przy projektowaniu obiektów biznesowych warto odpowiedzieć na następujące pytanie:

- Jaki jest minimalny zakres informacji niezbędny dla systemu odbiorcy do wykonania aktywności?

Po dostosowaniu do naszego studium przypadku pytanie to będzie brzmiało następująco:

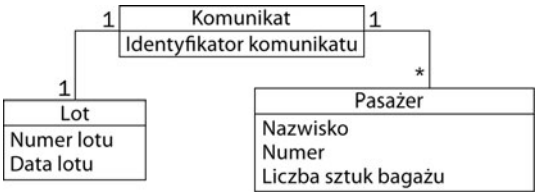
- Jaki jest minimalny zakres informacji związanych z listą pasażerów niezbędny dla obsługi transportu bagaży, aby można było wyładować bagaże pasażerów, którzy nie wsiedli na pokład samolotu?

Informacje niezbędne do sformułowania odpowiedzi na to pytanie można uzyskać od dostawców wiedzy (ekspertów) w dziedzinie transportu bagaży. Te informacje muszą zawierać:

- numer lotu, datę i czas planowanego startu, aby wyładować bagaże z właściwego samolotu;
- liczbę etykiet przyczepionych do bagażu pasażerów, którzy nie wsiedli na pokład.

Skonstruowanie diagramu klas — jaka jest struktura obiektu biznesowego?

Rysunek 5.20 przedstawia model klasy dla komunikatu przesyłającego **listę pasażerów**. Można w nim zamodelować dane niezbędne dla komórki transportu bagaży, ponadto posłuży on jako cel w procedurze przekształcania danych pozyskanych z systemu informatycznego.



Rysunek 5.20. Diagram klasy komunikatu „lista pasażerów"

Adaptacja klas i atrybutów z diagramu klas systemu informatycznego — jakie informacje mają być uwzględnione na diagramie klasy?

Obiekt biznesowy jest tworzony przez system informatyczny nadawcy, zatem w pierwszym kroku musimy przeanalizować diagram klas systemu informatycznego. Odpowiadamy na następujące pytanie:

- Jakie elementy danych obiektu systemowego mogą być zaadaptowane z diagramu klas systemu informatycznego?

W studium przypadku odpowiedzi na powyższe pytanie będą następujące (aby uwidocznić klasy, z których pochodzą poszczególne atrybuty, stosujemy notację **klasa.atrybut**):

- numer lotu: **numerlotu.opis**;
- data i czas planowanego startu: **lot.czasstartu**;
- numer pasażera: **bilet.numer**;
- nazwisko pasażera: **klient.nazwisko**.

Pozyskanie innych elementów danych — skąd wziąć pozostałe informacje?

Wszystkie elementy danych w obiekcie biznesowym, które nie mogą być bezpośrednio wygenerowane z diagramu klasy systemu informatycznego, trzeba zdobyć innym sposobem. Należy odpowiedzieć na następujące pytanie:

- W jaki sposób pozyskuje się elementy danych, które nie mogą być bezpośrednio wygenerowane z diagramu klas?

W naszym studium przypadku z systemu informatycznego nie da się wydobyć następujących informacji:

- **Jednoznaczna identyfikacja obiektu biznesowego**. Istnieje kilka możliwości. Na przykład możemy posłużyć się numerem seryjnym lub jednoznacznymi kluczami łączonymi (numer lotu z datą i czasem startu). Tutaj możemy wykorzystać jedną z dwóch opcji:
  - **Numer lotu z datą i czasem planowanego odlotu**. Te informacje są dostępne bezpośrednio w systemie informatycznym.
  - **Unikalny numer seryjny**. Tego typu numery generuje się w sposób automatyczny podczas tworzenia obiektu biznesowego i nie wchodzą one w skład diagramu klas systemu informatycznego.
- **Liczba sztuk bagaży należących do każdego pasażera**. Ta informacja nie jest dostępna w ramach diagramu klas systemu informatycznego. Można ją uzyskać, zliczając po prostu obiekty klasy **sztuka bagażu** przypisane danemu pasażerowi.

  Ta informacja jest potrzebna obiektowi biznesowemu, ponieważ w przypadku **rezygnacji z wejścia na pokład** umożliwia zweryfikowanie, czy zostały wyładowane już wszystkie sztuki bagażu.

- **Nadawca i odbiorca**. Szczególnie w przypadku wymiany danych między partnerami biznesowymi niezbędne jest przypisywanie do komunikatów dodatkowych informacji; dzięki temu mogą być one później łatwiej zidentyfikowane. Do informacji tego typu należą przede wszystkim identyfikacja nadawcy i odbiorcy, identyfikator (ID) komunikatu, czas wysyłki komunikatu itp. Na przykład na liście pasażerów PAXLST należy określić nazwisko osoby odpowiedzialnej za zawartość komunikatu.

Zdefiniowanie klas i związków obiektu biznesowego — jakie związki klas będą potrzebne?

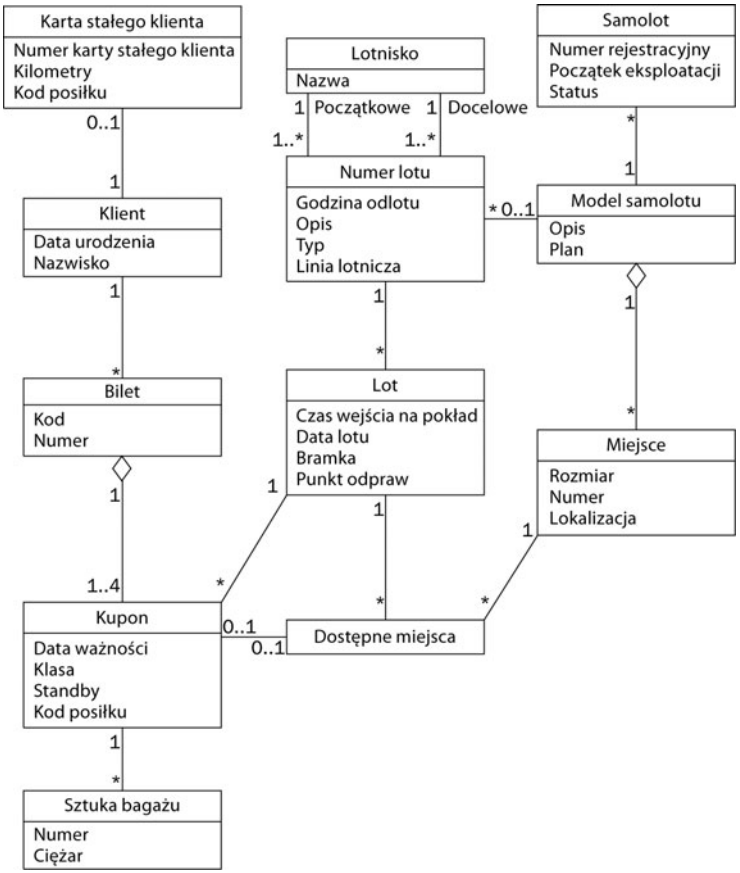
W diagramie klas obiektu biznesowego należy zdefiniować związki między klasami. Przydadzą się odpowiedzi na następujące pytania:

- W jaki sposób pozyskać odpowiednie klasy i ich związki z modelu biznesowego?

  Diagram klas modelu biznesowego zawiera obiekty biznesowe. W naszym studium przypadku **lista pasażerów** jest częścią diagramu klas systemu biznesowego **obsługa pasażerów**. Z opisu obiektu biznesowego w modelu biznesowym możemy pozyskać klasy, atrybuty i związki niezbędne do modelowania obiektów biznesowych wchodzących w skład komunikatu.

  Oprócz klas zawierających niezbędne dane często wykorzystujemy klasy umożliwiające połączenie między różnymi klasami w systemie informatycznym. W naszym przykładzie na rysunku 5.20 wzięliśmy dodatkowo pod uwagę klasy **kupon** i **bilet** — w przeciwnym wypadku nie mielibyśmy powiązania między lotem a pasażerem.
- W jaki sposób pozyskać dane i powiązania z innych standardowych komunikatów? Możemy zaadaptować elementy danych lub ich grupy z istniejących standardowych komunikatów o identycznej lub zbliżonej zawartości. Na przykład UN/EDIFACT daje do dyspozycji tak zwane standardowe segmenty danych (służące do grupowania elementów danych). Zawierają one takie informacje, jak **nazwisko i adres** lub **informacja o odbiorcy**. Nawet w przypadku, gdy nie zastosujemy standardu UN/EDIFACT, możemy wykorzystać klasy, powiązania i atrybuty z tych segmentów i elementów danych jako szkic do modelu obiektu biznesowego.
- W jaki sposób można przekształcać klasy z systemu informatycznego do obiektu biznesowego? Jak wspomnieliśmy wcześniej, klasy obiektu biznesowego mogą być pozyskane w części lub w całości z klas systemu informatycznego. Ponieważ wymagania w stosunku do klas systemu informatycznego są różne od wymagań w stosunku do klas obiektów biznesowych, należy dokonać transformacji klas i atrybutów z systemu informatycznego do obiektu biznesowego.

Rysunek 5.21 przedstawia diagram klas systemu informatycznego obsługi pasażerów. W ramach przykładu pokazującego transformację klas wykorzystaliśmy dane dotyczące lotu i pasażerów. W tym przykładzie pozostawiliśmy jednak klasy i atrybuty zupełnie niezwiązane z transformacją. Więcej informacji na temat tych zagadnień można znaleźć w dokumentacji *UML Profile for Enterprise Application Integration* dostępnej pod adresem *http://www.omg.org/technology/documents/formal/eai.htm*.



Rysunek 5.21. Diagram klas systemu informatycznego „obsługa pasażerów"

Weryfikacja perspektywy — czy wszystko zostało wykonane prawidłowo?

Kompletny diagram klas można zweryfikować za pomocą poniższej listy kontrolnej.

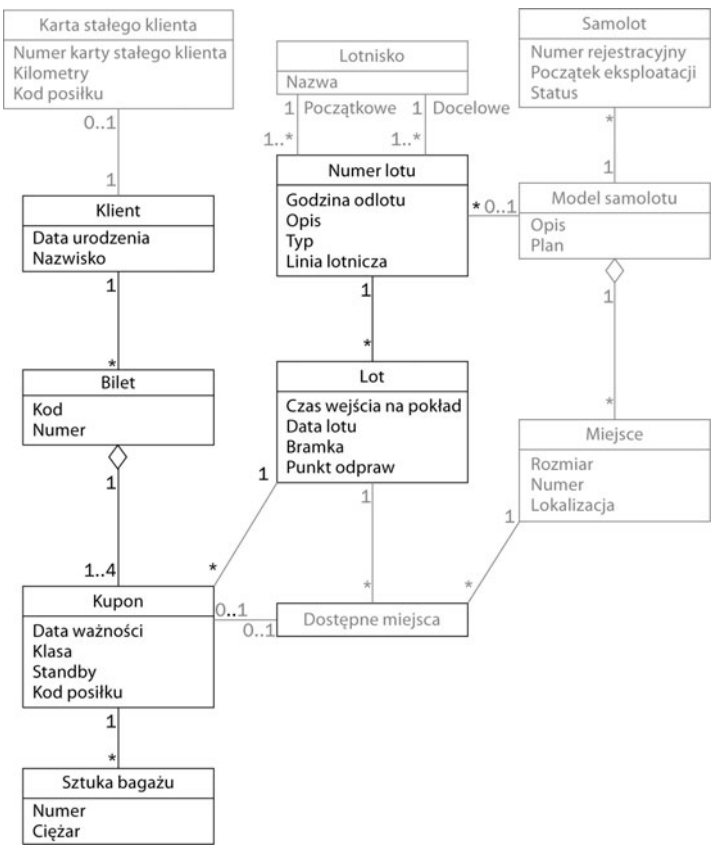
Lista kontrolna 5.5. Weryfikacja diagramu klas w perspektywie statycznej

- Czy diagram klas jest kompletny? Czy obiekty biznesowe zawierają wszystkie informacje potrzebne do tego, aby nadawca i odbiorca byli w stanie wykonywać aktywności związane z komunikatem?
- Czy obiekt biznesowy zawiera tylko te elementy danych, które mogą być zinterpretowane przez uczestników zewnętrznych?
- Czy w ramach interakcji mogą być wykorzystane standardowe komunikaty?
- Czy powiązania zostały oznakowane w czytelny sposób? Czy zwroty strzałek są skierowane w odpowiednie strony?
- Czy diagram klas jest prawidłowy? Większość błędów można zidentyfikować dzięki intensywnej analizie diagramu klas i śledzeniu poszczególnych scenariuszy z udziałem dostawców wiedzy.

5.5.4. Przekształcanie danych z systemu informatycznego do komunikatu „lista pasażerów"

W oparciu o dostępny diagram klas systemu informatycznego **obsługa pasażerów** oraz komunikat **lista pasażerów** objaśnimy tu niezbędne etapy doboru i transformacji danych związanych z lotem i pasażerami.

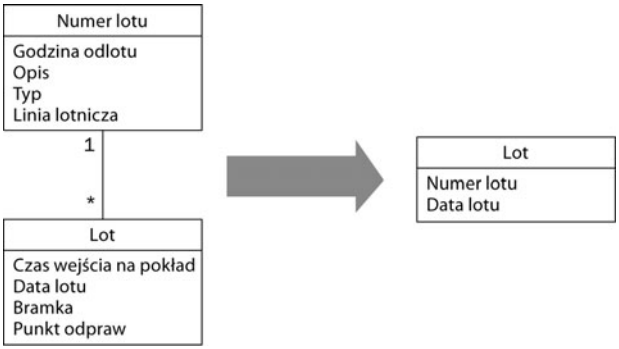
Na rysunku 5.22 wyodrębniliśmy klasy potrzebne do transformacji (dane lotu i pasażerów), a raczej wyszarzyliśmy klasy, które nie mają znaczenia przy tej operacji.



Rysunek 5.22. Diagram klas systemu „obsługa pasażerów"

Przekształcanie danych lotu

Na rysunku 5.23 widać, że każdy **numer lotu** ma związek z *n* lotów. Krotność klasy **lot** nie ma znaczenia, ponieważ w każdym komunikacie interesuje nas związek 1:1 — co znaczy, że za każdym razem interesuje nas tylko jeden lot. W obiekcie biznesowym dwie klasy systemu informatycznego można połączyć w jedną, zawierającą odpowiednie atrybuty każdej z klas początkowych.

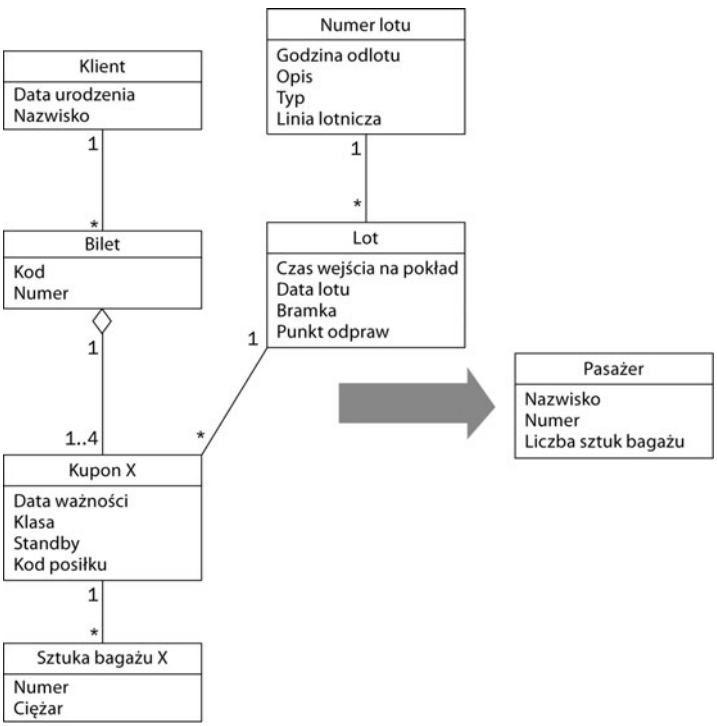


Rysunek 5.23. Przekształcenie klas „numer lotu" i „lot" w obiekt biznesowy „lot"

Aby pozyskać odpowiednie dane komunikatu z systemu informatycznego, jako kryterium wyboru musimy wskazać datę odlotu oraz numer lotu.

Przekształcanie danych pasażerów

Aby przekształcić dane pasażerów w dane składowe listy pasażerów, trzeba uwzględnić więcej etapów niż w przypadku danych lotu (rysunek 5.24).



Rysunek 5.24. Przekształcanie klas do obiektu biznesowego „lista pasażerów"

Definiujemy niezbędne dane:

- **klient.nazwisko**;
- **bilet.numer**;
- **liczba** (**sztuk bagażu**).

Sposób odczytu tych danych z systemu informatycznego jest zdefiniowany w regułach systemu. Należy przestrzegać tych reguł, a oprócz tego upewnić się, że na liście znajdą się tylko pasażerowie odbywający określony lot i jednocześnie tacy, którzy nie wsiedli na pokład samolotu.

5.5.5. Przekształcanie komunikatów z UML-a na różne formaty standardowe

Wszystkie standardowe formaty wymiany danych — ebXML, SWIFT czy UN/EDIFACT — mają zdefiniowane własne sposoby **reprezentacji** i **struktury** komunikatów. Komunikaty SWIFT na przykład są opisane nie tylko graficznie, lecz również tekstowo. Taka sama sytuacja ma miejsce również w przypadku UN/EDIFACT, gdzie ilustracje graficzne są ustandaryzowane i pokazane w postaci diagramów strukturalnych.

Istnieje jednak dążenie do modelowania komunikatów za pomocą jednolitej metodologii niezależnej od protokołu i implementacji, czyli za pomocą UML-a.

Z tego powodu wszystkie standardowe formaty można uzyskać w reprezentacji UML-a, macierzystego formatu wszystkich komunikatów. W zależności od dostępności można tego dokonać zgodnie ze ściśle zdefiniowanymi regułami przekształceń. W tym celu został opracowany profil *UML Profile for Enterprise Distributed Object Computing* wraz z zaleceniami dotyczącymi przekładu opisów utworzonych w UML-u na „rzeczywiste" systemy biznesowe. Ten profil jest z kolei oparty na standardzie ebXML.

Najważniejsze argumenty za stosowaniem UML-a jako neutralnej formy reprezentacji są następujące:

- Opis obiektów biznesowych i komunikatów jest niezależny od systemu i implementacji.
- UML to zaakceptowany i powszechnie stosowany standard.
- W UML-u istnieje możliwość tworzenia diagramów zawierających komunikaty w ramach procesów biznesowych.
- UML jest uniwersalnym językiem opisu całych systemów.

Podstawową zaletą wykorzystania neutralnego języka modelowania UML jest znacznie prostsza procedura przekształcania jednego formatu w inny. Z tego powodu zaleca się najpierw modelowanie komunikatów w UML-u, następnie przekształcanie ich do odpowiedniego formatu. Nie ma tu znaczenia, czy format docelowy jest standardowy, czy też samodzielnie opracowany na potrzeby organizacji. Szczególnie w tym drugim przypadku o wiele łatwiej przekształcać projekty z niestandardowych formatów na standardowe, jeśli istnieje ich specyfikacja w UML-u.

Szczegółowy opis reguł przekształcających znacznie wykracza poza tematykę tej książki. Zainteresowanych odsyłamy do dokumentów OMG: **Model Driven Architecture** (**MDA** *http://www.omg.org/mda/*) oraz definicji profili *UML Profile for Enterprise Distributed Object Computing* (*www.omg.org/technology/documents/formal/edoc.htm*) i *UML Profile for Enterprise Application Integration* (*http://www.omg.org/technology/documents/formal/eai.htm*), w których można znaleźć przekrojową analizę tematu z punktu widzenia UML-a.

Skorowidz

E

F

G

H

I

J

K

L

Ł

M

N

O

P

Q

R

S

Ś

T

U

W

X

Z