

Jedną z możliwych akcji jest akcja **send**, która specyfikuje zdarzenie, związane z nim atrybuty oraz obiekty, których to zdarzenie dotyczy. Wykonanie akcji oznacza generację wyspecyfikowanego w niej zdarzenia.

4.5. PERSPEKTYWA TRANSFORMACJI DANYCH - MODEL FUNKCJONALNY

Celem modelu funkcjonalnego jest opisanie dokonywanych przekształceń danych. Opis ten jest skoncentrowany na przedstawieniu zależności między danymi wejściowymi i wyjściowymi (na różnych poziomach szczegółowości), abstrahując od kolejności wykonywania przekształceń. Podstawowym pojęciem używanym w modelu jest *transformacja* danych (lub *proces*). Transformacja może odpowiadać operacji w modelu obiektowym lub akcji bądź aktywności w modelu dynamicznym. W modelu funkcjonalnym nie wnikamy w kolejność wykonania poszczególnych transformacji, z jednym zastrzeżeniem: model musi uwzględniać zależności występujące pomiędzy danymi związane z tym, że dana wyjściowa nie może być dostępna zanim nie będą dostępne związane z nią dane wejściowe (innymi słowy, skutek nie może poprzedzać przyczyny). Zależność ta jest przedstawiana w modelu w postaci powiązań wejść i wyjść poszczególnych transformacji.

Wzajemne wykorzystanie danych pomiędzy transformacjami jest zobrazowane w postaci *przepływów danych*. Transformacje danych są rysowane jako elipsy, a przepływy danych przedstawiane są w postaci łączących je strzałek. Każda transformacja może mieć wiele wejściowych i wyjściowych przepływów danych. Dla każdej transformacji, liczby różnych przepływów wejściowych i wyjściowych są ustalone. Zarówno transformacje jak i przepływy opatrzone są nazwami, które sugerują ich role w przetwarzaniu danych w modelowanym systemie. Pierwotne źródła danych oraz miejsca docelowe dla danych (tzn. miejsca gdzie dane „wsiadają”) nazywane są *aktorami* i reprezentowane są w postaci prostokątów. Aktor jest źródłem przepływu danych albo odbiorcą przepływu. Aktorzy reprezentują obiekty środowiska systemu, a związane z nimi transformacje danych leżą poza zakresem modelu.

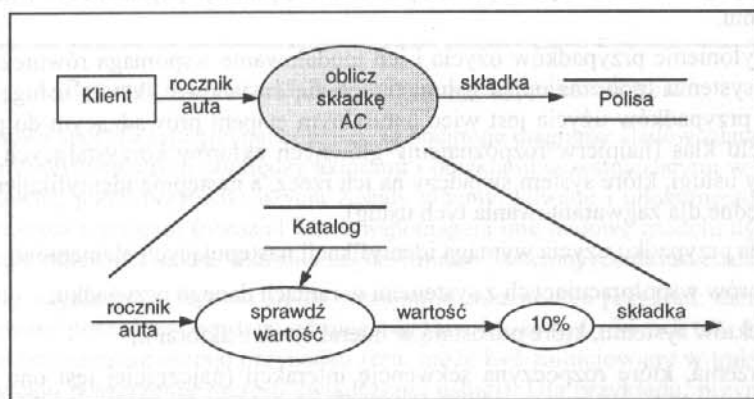
W pewnych sytuacjach występuje potrzeba modelowania danych, które wytwarzane są przez pewną transformację, a następnie, po pewnym okresie czasu, trafiają na wejście innej transformacji. Dane takie mogą być przechowane w *pojemniku danych* (ang. *data store*). Pojemnik danych reprezentuje pamięć, w której są przechowywane powierzone mu dane. Pojemnik danych nie dokonuje żadnej transformacji. Umożliwia jednak dostęp do danych w kolejności innej niż kolejność, w której dane zostały umieszczone w pojemniku. W notacji graficznej pojemnik jest przedstawiany jako para równoległych linii, z nazwą pojemnika umieszczoną pomiędzy tymi liniami.

Model funkcjonalny jest zbiorem *diagramów przepływu danych*. Diagram przepływu danych jest grafem, którego węzłami są transformacje danych, pojemniki danych i aktorzy, a węzły te są połączone przepływami danych. Diagramy składające się na model funkcjonalny tworzą hierarchię poziomów. Poziom najwyższy (najbardziej abstrakcyjny) odnosi się do całego systemu, a zawarte w nim transformacje reprezentują główne przekształcenia wykonywane przez system. W najprostszym przypadku przekształcenie takie jest funkcją (w sensie matematycznym), a więc nie ma żadnych efektów ubocznych. Transformacja taka może być dalej wyspecyfikowana używając jednej ze standardowych metod definiowania przekształceń *atomowych* (tzn. nie posiadających skutków ubocznych). Metody te obejmują:

- podanie wzoru matematycznego, wiążącego dane wejściowe z danymi wyjściowymi,
- podanie zależności wejścia i wyjścia w formie tabelarycznej (dla niedużych zbiorów wartości danych),
- podanie warunków wejściowych (*pre-condition*) i wyjściowych (*post-condition*) dla danych na wejściu i wyjściu z transformacji (aksjomatyczna definicja funkcji),
- specyfikację w postaci tablicy decyzyjnej,
- abstrakcyjny program (ang. *pseudocode*) specyfikujący sekwencję kroków obliczeniowych, poprzez które dane wejściowe są przekształcane w dane wyjściowe,
- opis w języku naturalnym.

Transformacje danych, które poddają się takiej specyfikacji, nie są dalej modelowane w postaci bardziej szczegółowych diagramów przepływu danych.

Natomiast transformacje, które posiadają efekty uboczne (tzn. reprezentują złożone obliczenia, uzależnione od stanu systemu), mogą podlegać dalszej dekompozycji. Dla każdej takiej transformacji tworzony jest oddzielny diagram przepływu danych. Wejścia i wyjścia tego diagramu są wejściami i wyjściami modelowanej transformacji. Proces ten może być kontynuowany aż do momentu gdy wszystkie wynikające z niego transformacje będą atomowe i nie będą wymagać dalszej dekompozycji. Przykład modelu funkcjonalnego związanego z obliczaniem składki ubezpieczenia „auto casco” przedstawiono na rysunku 4.14.



Rysunek 4.14. Dekompozycja transformacji „Oblicz składkę AC”

W pewnych sytuacjach może powstać potrzeba przesłania tych samych danych na wejście różnych transformacji. Jest to modelowane poprzez rozwidlenie przepływu danych i skierowanie go do różnych miejsc docelowych. Nazwa przepływu występuje raz i umieszczana jest na części wspólnej, przed rozgałęzieniem. W odróżnieniu od tej sytuacji, rozgałęziony przepływ może też oznaczać rozesłanie w różne miejsca różnych części danej zagregowanej. W takim przypadku, na każdej gałęzi umieszczana jest nazwa odpowiedniej części składowej.

Pojemniki danych umożliwiają przechowanie danych i następnie ich wykorzystanie, być może w innej kolejności. Dane zagregowane, umieszczone w pojemniku, mogą być również udostępniane w postaci swoich komponentów. Możliwe jest również dokonanie w ramach pojemnika agregacji różnych danych i dalsze ich udostępnianie jako jednego elementu.

Pojemniki danych odpowiadają obiektom w modelu obiektowym. Niektóre przepływy danych mogą być również obiektami, chociaż w większości wypadków są one wartościami bez własnej tożsamości. Przepływy danych, które przekazują obiekty są wyróżniane w modelu poprzez umieszczenie trójkąta zamiast strzałki na końcu przepływu. Notacja ta pozwala wyróżnić obiekty, które tworzone są lub wydobywane z innych, bardziej złożonych obiektów. W większości przypadków obiekty te są pojemnikami danych, a ich zawartość jest dalej przekształcana przez transformacje danych.

4.6. PERSPEKTYWA USŁUG - MODEL PRZYPADKÓW UŻYCIA

Przypadek użycia (ang. *use case*) reprezentuje ciągi interakcji pomiędzy *aktorem*, a systemem podlegającym analizie. Pod pojęciem aktora rozumiemy tu uogólnionego użytkownika, reprezentującego grupę konkretnych użytkowników, o podobnych oczekiwaniach i sposobie komunikacji z systemem. Przypadek użycia stanowi kompletny opis współpracy z systemem i odzwierciedla oczekiwania zainteresowanego użytkownika (a więc reprezentuje sensowną usługę ze strony systemu).

Przypadki użycia reprezentują więc usługi „widziane” przez użytkowników systemu. Identyfikacja i modelowanie przypadków użycia może być rozpoczęte wcześniej, już na etapie pozyskiwania wymagań, podczas kontaktów z różnymi udziałowcami systemu. Modele przypadków użycia wspomagają komunikację zespołu wykonawców z przyszłymi użytkownikami systemu. Mogą być również wykorzystane podczas weryfikacji poprawności i kompletności projektu systemu.

Jawne wyłonienie przypadków użycia i ich modelowanie wspomaga również identyfikację obiektów systemu (podczas opisu usługi trzeba się zastanowić, kto tę usługę świadczy). Modelowanie przypadków użycia jest więc naturalnym etapem prowadzącym do pełnej identyfikacji modelu klas (najpierw rozpoznajemy głównych aktorów korzystających z systemu, identyfikujemy usługi, które system świadczy na ich rzecz, a następnie identyfikujemy obiekty, które są niezbędne dla zagwarantowania tych usług).

Pełny opis przypadku użycia wymaga identyfikacji następujących elementów:

- aktorów współpracujących z systemem w ramach danego przypadku,
- obiektów systemu, które pozostają w interakcjach z aktorami,
- zdarzenia, które rozpoczyna sekwencję interakcji (najczęściej jest ono inicjowane przez aktora),
- warunków początkowych, od których uzależnione jest wystąpienie danego przypadku,
- opisu interakcji pomiędzy aktorami i obiektami, z wyszczególnieniem miejsc, w których mogą wystąpić sytuacje wyjątkowe,
- opisu niestandardowych przebiegów, dotyczących reakcji na sytuacje wyjątkowe,
- warunków końcowych, określających stan, w jakim dany przypadek użycia pozostawia uczestniczące w nim obiekty.

Punktem wyjścia do modelowania przypadków użycia jest rozpoznanie różnych aktorów współpracujących z systemem. W tym celu można wykorzystać Specyfikację Wymagań Systemowych (porównaj rozdział 2), która zawiera opis wszystkich udziałowców systemu. Spośród nich można wyłonić tych, którzy używają funkcji systemu (aktorzy pierwotni), tych