

4. Diagramy stanów odpowiadające śladom zdarzeń związanych z tą samą klasą są łączone w jeden wspólny diagram stanów. Diagram ten opisuje zachowanie danej klasy;
5. Należy dokonać weryfikacji, czy powstały w ten sposób diagram stanów jest spójny ze scenariuszami zdarzeń, w których dana klasa uczestniczy;
6. Należy uwzględnić i odwzorować w diagramie stanów awarie, błędy i inne sytuacje specjalne, które mogą wystąpić w trakcie wykonania diagramu.

4.10.4. Integracja modelu

Opis:

Celem tego kroku jest zapewnienie, że zbiór utworzonych diagramów stanu jest wewnętrznie spójny, a wytworzony model dynamiczny jest zgodny ze zidentyfikowanymi wcześniej interakcjami systemu z jego otoczeniem.

Wynik:

Model dynamiczny systemu.

Wskazówki:

1. Każde zdarzenie powinno mieć obiekt wysyłający i odbierający (może to być ten sam obiekt);
2. Stany nie poprzedzane przez inne stany oraz stany bez następnika powinny odpowiadać początkowym i końcowym stanom odpowiednich scenariuszy zdarzeń;
3. Diagramy stanów powinny być spójne ze śladami zdarzeń związanymi z daną klasą;
4. Model dynamiczny powinien być poddany zabiegom testowym mającym na celu zbadanie zdarzeń powstających w systemie w typowych sytuacjach związanych z jego użytkowaniem.

4.11. BUDOWA MODELU FUNKCJONALNEGO

Model funkcjonalny koncentruje się na przetwarzaniu danych. Reprezentuje przepływy danych oraz związane z nimi transformacje. Uwidocznione są jedynie zależności wejściowo-wyjściowe, z pominięciem kolejności, w której następuje wykonywanie poszczególnych przekształceń. Przepływ danych w modelu reprezentuje wartości atrybutów lub obiekty modelu obiektowego. Transformacje danych reprezentują operacje poszczególnych obiektów lub łączny efekt kilku operacji (być może z różnych obiektów).

4.11.1. Identyfikacja wejść i wyjść

Opis:

Następuje tu identyfikacja wejściowych i wyjściowych przepływów danych.

Wynik:

Diagram uwidaczniający przepływ danych z/do obiektów środowiska systemu.

Wskazówki:

1. Należy utworzyć listę danych wejściowych i wyjściowych systemu, które są parametrami zdarzeń wymienianych pomiędzy systemem i światem zewnętrznym. Zdarzenia bezparametrowe nie są brane pod uwagę, gdyż wpływają jedynie na przepływ sterowania bez skutków w przepływie danych;
2. Tworzony jest diagram uwidaczniający klasy (obiekty) środowiska oraz dane przepływające pomiędzy tymi obiektami i systemem (system jest reprezentowany w postaci jednej transformacji danych).

4.11.2. Budowa diagramu przepływu danych

Opis:

Powstaje diagram przepływu danych, który uwidacznia obliczenia mające na celu wywiedzenie danych wyjściowych na podstawie odpowiednich wejść.

Wynik:

Diagram przepływu danych systemu.

Wskazówki:

1. Analiza rozpoczyna się od transformacji obrazującej cały system z wejściowymi i wyjściowymi strumieniami danych skierowanymi od/do obiektów otaczającego środowiska;
2. Dla każdego wyjścia identyfikowane są niezbędne dane wejściowe oraz transformacja, która je przekształca w rozpatrywaną daną wyjściową. W sytuacji, gdy dana transformacja jest uzależniona od stanu systemu, stan ten jest reprezentowany przez odpowiedni pojemnik danych. Proces ten jest kontynuowany aż do momentu, gdy kolejna rozpatrywana transformacja wykorzystuje dane wejściowe dla całego systemu;
3. Powyższy proces jest kontynuowany oddzielnie dla każdej transformacji zdefiniowanej na poziomie wyższym, aż do momentu, w którym wszystkie transformacje reprezentują atomowe przekształcenia pomiędzy swoimi danymi wejściowymi i wyjściowymi.

4.11.3. Specyfikacja podstawowych transformacji danych

Opis:

Podstawowe (tzn. te, które nie podlegają dalszej dekompozycji) transformacje danych opisywane są jako zależność funkcyjna pomiędzy danymi wejściowymi i wyjściowymi (transformacje te nie są uzależnione od stanu systemu).

Wynik:

Specyfikacja zależności funkcyjnych w transformacjach podstawowych.

Wskazówki:

1. Specyfikacja powinna być dokonana bez odwoływania się do szczegółów implementacyjnych;
2. Wybór metody specyfikacji (np. język naturalny, równania matematyczne, definicja aksjomatyczna, pseudokod itp.) jest uwarunkowany wymaganiami odnośnie precyzji i jednoznaczności tworzonych specyfikacji.

4.11.4. Identyfikacja ograniczeń

Opis:

Identyfikowane są zależności występujące pomiędzy przepływami danych, które nie mogą być wyrażone w terminach relacji pomiędzy wejściem i wyjściem.

Wynik:

Lista ograniczeń dotyczących danych w systemie oraz niezbędne modyfikacje modelu funkcjonalnego.

Wskazówki:

1. Ograniczenia przyjmują zwykle postać warunków, które muszą być trwale spełnione przez wartości danych w systemie (tzw. *warunki niezmiennicze*). Przykładem takiego warunku dla aplikacji bankowej jest wymaganie: "saldo rachunku nie może przyjąć wartości niższej niż limit kredytu przyznanego temu rachunkowi";
2. Ograniczenia powinny znaleźć swoje odzwierciedlenie w modelu funkcjonalnym (pokazanie, w jaki sposób są sprawdzane) oraz w modelu dynamicznym (wskazanie na to, co się stanie, gdy ograniczenie nie jest spełnione).

4.12. INTEGRACJA MODELI

4.12.1. Identyfikacja operacji

Opis:

Mimo tego, że niektóre operacje mogą być zidentyfikowane stosunkowo wcześniej, już na etapie budowy modelu obiektowego, ich ostateczna lista jest wynikiem łącznej analizy wszystkich trzech modeli.