

Instrukcje i wskazówki do przygotowania diagramów dynamicznych

(IO 2025/2026)

Zakres zadania

- Oczekiwane jest opracowanie łącznie 6 diagramów czterech typów:
 - 3 diagramy sekwencji (dla wybranych przypadków użycia)
 - 1 diagram komunikacji (dla wybranego przypadku użycia)
 - 1 diagram czynności (do wyboru: dla wybranego przypadku użycia lub dla ogólniejszego procesu biznesowego)
 - 1 diagram stanów (dla wybranej klasy)
- Wybierane przypadki użycia nie powinny się powtarzać tzn. jeżeli dla danego przypadku użycia (PU) zrobiono diagram sekwencji, to nie robimy dla niego diagramu komunikacji lub czynności.
- Należy wybierać nietrywialne PU i klasę, w przeciwnym razie trudno liczyć na maksymalną punktację.

Diagramy sekwencji

1. Pokazują, jaka interakcja (wymiana komunikatów) pomiędzy obiektami musi nastąpić w celu dostarczenia zewnętrznemu aktorowi usługi opisanej w przypadku użycia. Ilustrują tę interakcję skupiając się na upływie czasu.
2. Standardowo diagram sekwencji robi się dla pojedynczego PU, można jednak również np. dla dwóch PU połączonych związkiem *include* lub *extend*.
3. Jako obiekty/linie życia diagram zwykle zawiera: aktora, obiekt ogólnie reprezentujący interfejs użytkownika i obiekty będące wystąpieniami klas zdefiniowanych w poprzednim etapie (podczas opracowywania diagramu klas).
4. W sumie jedynie obiekt reprezentujący interfejs użytkownika musi być dodany na diagram za pomocą menu Toolbox. Pozostałe elementy (aktorzy, obiekty istniejących klas) zostały już zdefiniowane w poprzednich etapach i wystarczy je „przeciągnąć” z Browsera na diagram sekwencji (wstawiając jako Lifeline!)
5. Jeżeli zespół dojdzie do wniosku, że jest taka potrzeba, możliwe jest zmodyfikowanie diagramu klas poprzez np. dodanie klasy czy związku. Często potrzebne okazuje się dodanie klasy kontenerowej, gdy w ramach tego, co opisuje PU, występuje potrzeba znalezienia jednego konkretnego obiektu spośród setek czy tysięcy innych obiektów tej samej klasy przechowywanych w systemie.
6. Interakcja pokazywana jest za pomocą komunikatów (Message dostępny w Toolbox). To, co dzieje się między aktorem a interfejsem można przedstawiać dość dowolnie (komunikaty synchroniczne lub asynchroniczne, komunikaty ogólniejsze niż każda decyzja/wprowadzana informacja), ale powinno odpowiadać opisowi scenariusza interakcji z PU. Jednakże to, co dzieje się dalej, „w głębi systemu”, jest już bardziej ustandaryzowane i powinno polegać na wymianie komunikatów synchronicznych.
7. Komunikat synchroniczny to nic innego jak wywołanie operacji w obiekcie, do którego kierowany jest ten komunikat, a komunikat zwrotny to powrót sterowania po wykonaniu operacji (i ewentualnie zwrócenie przez nią jakiejś wartości).

8. Komunikaty synchroniczne powinny zatem odpowiadać operacjom zdefiniowanym w klasie obiektu, do którego kierowany jest komunikat. Jeżeli takiej operacji nie ma, należy ją najpierw dodać do klasy, a potem wybrać ją dla danego komunikatu.
9. Jeśli w interakcji występują jakieś przebiegi alternatywne lub opcjonalne albo też jakaś część interakcji jest wykonywana iteracyjnie (w pętli), to należy to zilustrować za pomocą dostępnego w menu Toolbox elementu Fragment. Dla Fragmentu należy wybrać jego typ (Alt, Opt, Loop) i umieścić go na diagramie tak, aby część interakcji, której dotyczy, była zawarta w jego granicach.

Diagramy komunikacji

1. Pokazują, jaka interakcja (wymiana komunikatów) pomiędzy obiektami musi nastąpić w celu dostarczenia zewnętrznemu aktorowi usługi opisanej w przypadku użycia. Ilustrują tę interakcję skupiając się na strukturze powiązań między uczestniczącymi obiektami.
2. Diagramy komunikacji są de facto bardzo podobne do diagramów sekwencji, choć na pierwszy rzut oka wyglądają mocno odmiennie. Z uwagi na to podobieństwo, wiele z poniższych wskazówek jest analogicznych do tych wymienionych dla diagramów sekwencji.
3. Standardowo diagram komunikacji robi się dla pojedynczego PU, można jednak również np. dla dwóch PU połączonych związkiem *include* lub *extend*.
4. Jako obiekty diagram zwykle zawiera: aktora, obiekt ogólnie reprezentujący interfejs użytkownika i obiekty będące wystąpieniami klas zdefiniowanych w poprzednim etapie (podczas opracowywania diagramu klas).
5. W sumie jedynie obiekt reprezentujący interfejs użytkownika musi być dodany na diagram za pomocą menu Toolbox. Pozostałe elementy (aktorzy, obiekty istniejących klas) zostały już zdefiniowane w poprzednich etapach i wystarczy je „przeciągnąć” z Browsera na diagram sekwencji (wstawiając jako Object!)
6. Jeżeli zespół dojdzie do wniosku, że jest taka potrzeba, możliwe jest zmodyfikowanie diagramu klas poprzez np. dodanie klasy czy związku. Często potrzebne okazuje się dodanie klasy kontenerowej, gdy w ramach tego, co opisuje PU, występuje potrzeba znalezienia jednego konkretnego obiektu spośród setek czy tysięcy innych obiektów tej samej klasy przechowywanych w systemie.
7. Po dodaniu obiektów, kolejnym krokiem jest dodanie powiązań między nimi (Associate w Toolbox). Powiązania pokazują, które obiekty się „widzą” i mogą bezpośrednio wymieniać komunikaty. Powiązania będą przede wszystkim odwzorowywać związki z diagramu klas – jeżeli tam między dwiema klasami była np. asocjacja, to tutaj między obiektami stanowiącymi instancje tych klas będzie powiązanie. Nie jest to jednak jedyna możliwość, kiedy można dodać powiązanie, bo np.
 - Jeśli modelowany diagramem komunikacji PU rozszerza inny PU, w którym konkretny obiekt został już wyszukany i jest dostępny, to tutaj może już być dostępny np. dla obiektu reprezentującego interfejs użytkownika.
 - Klasy kontenerowe zwykle mają jedną jedyną instancję i można założyć, że taki obiekt będzie dostępny dla innych, bez konieczności modelowania tego związkami na diagramie klas.
8. Interakcja pokazywana jest za pomocą komunikatów. Komunikaty dodaje się do powiązań – kliknięcie prawym przyciskiem myszy na powiązaniu i wybór opcji „Add message from ... to ...”.
9. To, co dzieje się między aktorem a interfejsem można przedstawiać dość dowolnie (komunikaty synchroniczne lub asynchroniczne, komunikaty ogólniejsze niż każda decyzja/wprowadzana informacja), ale powinno odpowiadać opisowi scenariusza interakcji z PU. Jednakże to, co dzieje

się dalej, „w głębi systemu”, jest już bardziej ustandaryzowane i powinno polegać na wymianie komunikatów synchronicznych.

10. Komunikat synchroniczny to nic innego jak wywołanie operacji w obiekcie, do którego kierowany jest ten komunikat, a komunikat zwrotny to powrót sterowania po wykonaniu operacji (i ewentualnie zwrócenie przez nią jakiejś wartości).
11. Komunikaty synchroniczne powinny zatem odpowiadać operacjom zdefiniowanym w klasie obiektu, do którego kierowany jest komunikat. Jeżeli takiej operacji nie ma, należy ją najpierw dodać do klasy, a potem wybrać ją dla danego komunikatu.
12. Jeżeli chce się pokazać, że jakieś komunikaty są warunkowe albo wykonywane iteracyjnie, to można to zaznaczyć otwierając właściwości danego komunikatu i odpowiednio wpisując warunek w pole „Condition” lub zaznaczając checkbox „Is iteration”.

Diagramy czynności

1. Mogą pokazywać zarówno to, co dzieje się w ramach realizacji danego przypadku użycia (i wówczas mamy tam przedstawienie kroków interakcji oraz wewnętrznych działań systemu jako akcji). Mogą jednak również pokazywać ogólniejszy proces biznesowy np.
 - co się dzieje w pizzerii od momentu złożenia zamówienia telefonicznego przez klienta?
 - jak wygląda obsługa wniosku o odszkodowanie złożonego u ubezpieczyciela?i wówczas akcje będą przedstawiały różne działania, przede wszystkim ludzi, nie ograniczone jedynie do systemu, ale pokazujące ogólniej, co dzieje się w obszarze problemowym. Wybór, który wariant diagramu czynności zostanie przygotowany należy do zespołu.
2. Dla tych diagramów wszystkie elementy dodaje się z menu Toolbox, niczego nie „przeciągamy”.
3. Podstawowe elementy diagramu to akcje (Action) łączone w proces za pomocą przejść (ProcessFlow). Poza tym używane są węzły pokazujące początek (Initial) i koniec (Final) procesu oraz węzły kontrolne: decyzyjne (Decision) pokazujące wybory i ścieżki alternatywne oraz zrównoleglenia (Fork) pokazujące, że od tego miejsca pewne ścieżki realizowane są równolegle (współbieżnie). Jeżeli po jakimś czasie alternatywne ścieżki znowu łączą się w jedną należy zastosować węzeł Merge. Analogicznie, kiedy kończy się zrównoleglenie, stosowany jest węzeł Join.
4. Dodatkowo można zastosować elementy Partition do pokazania kto lub co wykonuje daną akcję. Dla diagramu czynności ilustrującego PU może to być proste rozróżnienie aktor/system albo też dokładniejsze wskazanie, które klasy w ramach systemu będą realizować dane akcje. Dla diagramu czynności ilustrującego proces biznesowy będą to zwykle osoby/jednostki realizujące poszczególne akcje.

Diagramy stanów

1. W przeciwieństwie do wszystkich opisanych powyżej diagramów, diagramy stanów nie modelują PU, ale zachowanie obiektów określonej klasy.
2. Po wyborze klasy, dla obiektów której ma być robiony diagram stanów, proszę się zastanowić czy będą to obiekty informacyjne (i diagram będzie pokazywał pewien „workflow”, któremu podlegają) czy obiekty czasu rzeczywistego (dla których diagram będzie faktycznie pokazywał co się z nimi dzieje w każdej sekundzie). Proszę zajrzeć do slajdów wykładowych, gdzie opisano różnice w modelowaniu tych wariantów.

3. Dla tych diagramów wszystkie elementy dodawane są z menu Toolbox.
4. Podstawowe elementy to stany (State). Zawsze występuje również stan początkowy (Initial). Należy natomiast być ostrożnym z używaniem stanu końcowego (Final), ponieważ oznacza on „koniec obiektu” czyli jego zniszczenie/usunięcie – może zatem wystąpić gdy świadomie chcemy pokazać, że obiekt może zostać usunięty, ale na pewno nie jako ostatni element dla workflow pokazującego przetwarzanie zamówienia klienta.
5. Zmiany stanów mogą następować poprzez przejścia między nimi (Transition). Przejście powinno zostać opisane, „gołe” przejście ze stanu A do stanu B oznacza, że zmiana stanu nastąpi natychmiast tzn. gdy tylko obiekt znajdzie się w stanie A, od razu przejdzie do stanu B (nie czekając na żadne zdarzenia ani spełnienie jakichkolwiek warunków).
6. Opis przejścia może obejmować:
 - zdarzenie, które inicjuje przejście pomiędzy stanami;
 - warunek konieczny do spełnienia (jeśli zajdzie zdarzenie, ale warunek nie będzie spełniony, to zmiana stanu nie nastąpi);
 - dodatkową akcję wykonywaną podczas przejścia np. wysłanie komunikatu do innego obiektu.
7. Powyższe elementy można wprowadzić poprzez otwarcie okna „Transition properties” (dwuklik na przejściu), w polach:
 - Zdarzenie – Triggers.Name
 - Warunek – Guard
 - Akcja - Effect
8. Możliwe jest tworzenie stanów złożonych (zawierających podstany). W tym celu należy po prostu powiększyć stan, który ma być tym złożonym i „przeciągnąć” w jego granice stany, które mają być jego podstanami. Gdyby potem chcieć z podstanu zrobić znowu stan „samodzielny” należy znaleźć odpowiednie miejsce w Browser (okno pokazujące hierarchiczną strukturę pakietów i elementów całego modelu) – podstan będzie tam „podczepiony” pod stan złożony, należy do „przeciągnąć” o poziom wyżej.
9. Gdyby potrzebne było modelowanie stanów równoległych, wywołuje się to następująco: prawy przycisk myszy na wybranym stanie i opcja: Advanced->Define Concurrent Substates.