

Wprowadzenie do modelowania i analizy systemów informatycznych

mgr inż. Stanisław Lota

Co będziemy robić?

- **Analizy i modelowania systemów informatycznych z wykorzystaniem języka Unified Modeling Language (UML 2.x)**
- Modelowanie procesów biznesowych
- Dokumentowanie wymagań funkcjonalnych
- Budowanie modelu dziedzicznego przy zastosowaniu wzorców
- Modelowanie złożonych aspektów systemów informatycznych (np. diagramy stanów, czynności, sekwencji)
- Transformacja modelu UML w kod określonym języku programowania.

Literatura podstawowa

Mariusz Trzaska : Modelowanie i implementacja systemów informatycznych

Jarosław Żeliński : Analiza biznesowa. Praktyczne modelowanie organizacji

Stanisław Wrycza, Bartosz Marcinkowski, Krzysztof Wyrzykowski : Język UML 2.0 w modelowaniu systemów informatycznych



Co to jest model

Modele są budowane w kontekście biznesowym lub informatycznym w celu lepszego zobrazowania istniejących lub przyszłych systemów.

Elementy świata i modelu:

- użytkownicy, systemy zewnętrzne;
- dane, ich struktura, sposób przetwarzania, zależności statyczne i dynamiczne;
- procesy, ich struktura i rozmieszczenie;

Należy pamiętać, że model nigdy nie będzie w pełni odpowiadał rzeczywistości.

Co to jest modelowanie

Modelowanie można określić jako próbę uchwycenia w kategoriach informatycznych i wyrażenia za pomocą specjalnej notacji graficznej najważniejszych cech rozwijanego systemu oraz jego otoczenia

Powstały w efekcie model ma umożliwić łatwe zaprojektowanie i implementację kodu.

Co to jest modelowanie

Metodyka modelowania jest opisem czynności, sposobu i kolejności ich realizacji; czynności te mają prowadzić ku MODELOWI, zapewniając jednocześnie metody utrzymania wysokiej jakości (spełnienia wymagań użytkownika).

Co to jest modelowanie

Na każdym etapie tworzenia oprogramowania modelowanie ma dostarczać tylko tyle informacji ile jest w danym momencie potrzebne **(pomijając nieistotne szczegóły – stąd modelowanie jest sztuką abstrakcji)**.

Modelowanie jest nieodłącznie związane z **podkreślaniem i pomijaniem: podkreślane są szczegóły istotne, pomijane zaś szczegóły zbędne**. Nie ma uniwersalnej odpowiedzi na pytanie o to, co wyróżnić a co pominąć.

Co to jest modelowanie

Dla niektórych projektów, zwłaszcza przy **tworzeniu nowatorskiego oprogramowania w dotąd niezinformatyzowanych dziedzinach**, modelowanie jest być może **najważniejszym elementem rozwijania oprogramowania**, od którego w największym stopniu zależy sukces całego przedsięwzięcia

W innych projektach zdarza się przechodzenie bezpośrednio z fazy wymagań do bardzo konkretnego projektu kodu (można powiedzieć, że faza modelowania jest tutaj niejawna, a model systemu odbija się w architekturze kodu).

Co to jest modelowanie

Wytwarzanie współczesnego oprogramowania to proces bardzo skomplikowany. Bierze w nim udział cały sztab ludzi, z których każdy ma konkretne zadanie do wykonania. **Aby te osoby mogły się wzajemnie porozumieć, muszą mówić wspólnym językiem.**

W przypadku projektowania systemu informatycznego do tego celu najczęściej używa się notacji UML (Unified Modeling Language). Umożliwia ona w miarę precyzyjne opisanie wszystkich elementów składających się na projekt nowoczesnego oprogramowania.

Rodzaje modeli

Modele systemu można podzielić na dwie podstawowe grupy:

- **modele zachowania systemu (aspekt dynamiczny)** reprezentują zestaw działań realizowanych przez system. Działania należą zazwyczaj do dwóch grup: zmiana stanu systemu oraz przetwarzanie danych wejściowych w dane wyjściowe.
- **modele struktury systemu (aspekt statyczny)** pokazują statyczne elementy składowe systemu i ich wzajemne zależności.
- Istnieją rodzaje modeli ukazujące oba aspekty systemu.

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.
- 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.
- Ważne aby decyzje były później tak samo czytelne, jak w momencie ich podejmowania.
- Musi być weryfikacja faktów pod kątem kompletności spójności i poprawności.

Ogólny proces tworzenia modeli możemy podzielić na:

- Analizę obecnych elementów systemu
- Obserwację zachowania się elementów
- Zastosowanie praw/zasad do określenia postaci modelu i wartości jego parametrów
- Odtworzenie funkcji wykonywanych przez elementy w modelu
- Identyfikację brakujących elementów modelu
- Zaprojektowanie brakujących elementów
- Weryfikację utworzonego modelu przez porównanie jego odpowiedzi z odpowiedzią obecnych elementów

Proces modelowania

Tworzenie modelu systemu informatycznego odbywa się często w ramach wybranej metodologii rozwijania oprogramowania.

Stosowane narzędzia wspomagające i produkowane modele (najczęściej w postaci diagramów i innych form graficznych) związane są z metodologią procesu wytwarzania.

Istnieje wiele specyficznych metodologii (metody Coada i Yourdona, OOSE, OMT, RUP, Extreme Programming i wiele innych).

Proces modelowania

W ramach kaskadowego modelu rozwijania oprogramowania modelowanie towarzyszące procesowi tworzenia oprogramowania można podzielić na etapy:

- modelowanie w fazie określania wymagań (często określane mianem analizy)
- modelowanie w fazie projektowania kodu
- modelowanie w fazie implementacji

W modelu przyrostowym modelowanie w każdej iteracji może dotyczyć aktualnie rozważanego przyrostu i może także być podzielone (mniej lub bardziej formalnie) na etapy.

Proces modelowania

Pierwszą fazą modelowania może być próba opisu dziedziny zastosowań, środowiska, w którym ma funkcjonować system informatyczny. Początkowo można modelować istniejący stan otoczenia bez systemu i następnie zmodyfikować model przez odpowiednie umieszczenie systemu.

Drugą fazą może być modelowanie funkcjonowania systemu w ramach środowiska.

Wreszcie na końcu następuje modelowanie samego systemu – jego wewnętrznej struktury i działania. W miarę prac nad projektem, model systemu rozrasta się i staje się coraz bardziej szczegółowy.

Proces modelowania

W fazie modelowania systemu w jego otoczeniu ważną decyzją jest określenie granic systemu – jakie czynności (obowiązki) powierzyć systemowi, a jakie pozostawić w aktualnym stanie.

Decyzja taka ma istotne znaczenie dla organizacji procesu tworzenia oprogramowania i powinna zostać podjęta odpowiednio wcześnie.

Proces modelowania

Pierwszym modelem systemu, może być jego graficzna prezentacja jako pojedynczego symbolu wraz ze wszystkimi zewnętrznymi elementami, z którymi oddziałuje (“punktami widzenia”, “aktorami”). Taki model systemu jest przydatny w fazie odkrywania i ustalania wymagań dla systemu (jego ilustracją bywają diagramy przypadków użycia systemu).

Wizualizacja wszystkich procedur z punktu widzenia klientów, ekspertów i użytkowników jest bardzo ważna. Wszystkie skumulowane procedury związane z systemem powinny być zaprezentowane w taki sposób, aby każdy zainteresowany mógł je zrozumieć.

Grupa docelowa modelu

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

- Jakiego poziomu zaawansowania biznesowego należy oczekiwać 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?
- Ile czasu grupa docelowa może poświęcić na analizę i interpretację modelu?

Grupa docelowa modelu

- Jakim językiem posłużyć się przy definicji modelu?
- Czy grupa docelowa zrozumie biznesową terminologię techniczną?
- Czy zrozumie terminologię informatyczną?
- Jaki zastosować poziom abstrakcji?

Grupa docelowa 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.

Aby uniknąć przysypania modelu stosem papierów, należy zadbać o odpowiedni poziom szczegółowości i komplikacji — w przeciwnym wypadku nikt nie będzie zaprzętał sobie nim głowy.

Przykład:

Jeśli na butelce z wodą widnieje etykieta z napisem „woda”, mamy pewność, że prawie każdy, kto ją przeczyta, domyśli się, jaka jest zawartość butelki.

Jeśli jednak na etykiecie napisać „H₂O”, 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.

Grupa docelowa modelu

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.

Grupa docelowa modelu

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.

Przykłady modeli

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

Proces analizy

Proces analizy składa się z faz:

- pozyskiwania informacji od dostawców wiedzy;
- reprezentowania (specyfikowanie);
- weryfikacji.

Dostawcy wiedzy

Jako dostawcy wiedzy mogą służyć

- uczestnicy i kontrolerzy procesów biznesowych;
- użytkownicy systemów informatycznych o funkcjonalności zbliżonej do modelowanego systemu lub związanej z nim;
- klienci;
- eksperci w analizowanej dziedzinie;
- niezależni obserwatorzy.

Proces analizy

Proces analizy realizowany jest 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.

Proces analizy

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, np.klienta;
- ankiety i 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.

Proces analizy

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.