PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU nr 118
Informatyka Ekonomiczna 17 2010

Bogdan Burkot
Uniwersytet Ekonomiczny we Wrocławiu

MODELOWANIE WYMAGAŃ Z ZASTOSOWANIEM METOD OBIEKTOWYCH

Streszczenie: Artykuł przedstawia możliwość zastosowania obiektowych metod modelowania, szczególnie ukierunkowanych na tworzenie modelu wymagań funkcjonalnych. Modelowanie wymagań zostaje przedstawione w kontekście szerszego procesu, jakim jest analiza biznesowa. Zawiera on opis podstawowych pojęć metod obiektowych. Następnie przedstawiono procedurę modelowania wymagań oraz zastosowanie diagramów przypadków do ich przedstawiania.

Słowa kluczowe: analiza biznesowa, metody obiektowe. modelowanie wymagań użytkowników.

1. Wstęp

Do głównych przyczyn niepowodzeń realizacji przedsięwzięć tworzenia oprogramowania można zaliczyć: niedokładne zrozumienie potrzeb użytkowników, zmieniające się wymagania oraz niską jakość oprogramowania [Kruchten 2007, s. 4]. Przyczyny te są związane z produktami analizy biznesowej. Dobrze zorganizowany proces analizy biznesowej umożliwia pozyskanie szczegółowych wymagań, zarządzanie ich zmianami oraz wykorzystanie jego wyników w procesie testowania oprogramowania.

Trudność tworzenia oprogramowania wynika również z dużej jego złożoności, co utrudnia całościowe rozumienie systemu. Wzrost złożoności systemu jest jedną z głównych przyczyn konieczności wyeliminowania dwuznaczności opisu systemu w języku naturalnym. Rozwiązaniem tego problemu może być zastosowanie metod analizy oraz modelowania systemu z wykorzystaniem powszechnie stosowanych notacji.

Celem artykułu jest charakterystyka metod obiektowych oraz prezentacja zastosowania modeli przypadków użycia do opisu wymagań funkcjonalnych realizowanego w procesie analizy biznesowej.

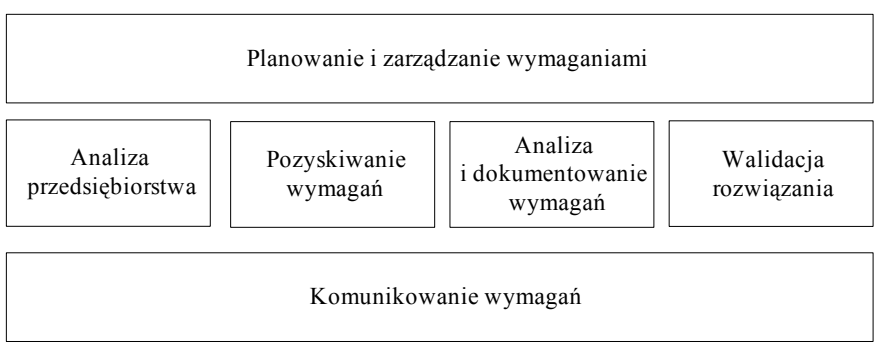
2. Proces analizy biznesowej

Analiza biznesowa jest procesem ukierunkowanym na badanie problemu oraz analizę i dokumentowanie wymagań, które musi spełniać rozwiązanie. Problem jest rozumiany jako niepożądany stan początkowy, którego transformacja w stan pożądany wymaga twórczego podejścia [Pawlak 2007, s. 33]. Wymaganie „to warunek, który musi być spełniony, albo zdolność do wykonania czegoś, którą musi mieć system" [Kruchten 2007, s. 130]. Wymagania wobec systemu można podzielić na dwie grupy [Sommerville 2001, s.100]:

– funkcjonalne – opisują, jakie czynności system powinien wykonywać,
– niefunkcjonalne – opisujące właściwości odnoszące się do całości systemu (np. użyteczność, niezawodność, efektywność, łatwość wspierania).

Zgodnie z wytycznymi BABOK (Business Analysis Body of Knowledge) analiza biznesowa obejmuje następujące etapy (rys. 1) [Zmitrowicz 2009, s. 58]:

– analizę przedsiębiorstwa,
– planowanie wymagań i zarządzanie wymaganiami,
– pozyskiwanie wymagań,
– analizę i dokumentowanie wymagań,
– komunikowanie wymagań,
– walidację rozwiązania.



Rys. 1. Etapy analizy biznesowej

Źródło: na podstawie [Zmitrowicz 2009, s. 59].

Analiza przedsiębiorstwa pozwala określić kontekst dla wymagań projektowych. Na tym etapie jest opisywany bieżący oraz przyszły stan organizacji. Dodatkowo następuje w nim opracowanie studium wykonalności, mającego na celu wybór optymalnego rozwiązania, a także formułowany jest zakres projektu.

Planowanie i zarządzanie wymaganiami to etap, w ramach którego następuje: określenie udziałowców i przypisanie im ról, zdefiniowanie zadań związanych z gromadzeniem wymagań oraz ich organizacja, zarządzanie zakresem wymagań,

zarządzanie zmianą wymagań oraz komunikowanie aktualnego stanu gromadzenia wymagań.

Pozyskiwanie wymagań polega na ich odkrywaniu przez wchodzenie w interakcje z udziałowcami projektu. Pozyskiwanie wymagań może odbywać się przy zastosowaniu następujących technik: burza mózgów, analiza dokumentacji, analiza interfejsów, wywiad, obserwacja, prototypowanie, warsztaty itp. Wymagania powinny być [Sommerville 2001, s. 137]:

– poprawne – zgodne z pożądanym stanem rzeczy,
– kompletne – określające wszystkie funkcje oraz ograniczenia zdefiniowane przez użytkownika,
– spójne – nie są sprzeczne pomiędzy sobą lub nie ma różnych opisów tej samej funkcji systemu,
– weryfikowalne – można dla nich zaprojektować sprawdzenia ich spełnienia przez system.

Analiza i dokumentowanie wymagań obejmuje: określenie zależności pomiędzy wymaganiami, projektowanie struktury funkcjonalnej systemu oraz szczegółową specyfikację wymagań. Analiza i dokumentacja powinny przebiegać iteracyjnie. Wyniki każdej z iteracji powinny być weryfikowane przez zespół twórców oprogramowania pod kątem możliwości jednoznacznej implementacji oraz przez przedstawicieli klienta pod kątem zgodności z oczekiwaniami.

Komunikowanie wymagań wiąże się z zapewnieniem dostępu do dokumentacji analitycznej wszystkim członkom zespołu oraz z przekazywaniem informacji o jej zmianach. Komunikowanie wymagań może odbywać się za pomocą repozytorium z zastosowaniem mechanizmu kontroli wersji. Każde z wymagań powinno mieć określone następujące informacje: numer referencyjny, treść, priorytet oraz opis sposobu realizacji. Dodatkowo ważne jest informowanie o dokonanych zmianach wymagań.

Walidacja rozwiązania obejmuje prace związane ze wsparciem przez analityka biznesowego w zakresie: prac programistycznych, przeglądu produktów implementacji, przygotowania planów, przypadków i scenariuszy testowych, przygotowania i realizacji testów akceptacyjnych.

3. Metody obiektowe

Wyeliminowanie dwuznaczności opisu systemu informatycznego w języku naturalnym można osiągnąć przez zastosowanie sformalizowanych metod, definiowanych jako „świadome i celowe działania zmierzające do rozwiązania danego problemu w skończonej liczbie kroków" [Nowicki 2006, s. 160].

Metody obiektowe wykorzystują pojęcie obiektowości do modelowania pojęciowego oraz analizy i projektu systemu [Beynon-Davies 2004, s. 228]. System przedstawiany jest jako zbiór obiektów komunikujących się pomiędzy sobą za pomocą komunikatów. Obiekt reprezentuje zarówno strukturalny (dane), jak i behawioralny aspekt systemu.

W celu wyjaśnienia istoty podejścia obiektowego zostaną zdefiniowane podstawowe pojęcia z nim związane, takie jak: obiekt, klasa, abstrakcja, enkapsulacja oraz dziedziczenie. Obiekt można rozumieć jako element z pewnej dziedziny. Każdy obiekt cechuje się [Płodzień 2003, s. 39]:
– tożsamością, która wyróżnia go ze zbioru innych obiektów;
– stanem, opisywanym przez aktualne w danym czasie wartości jego atrybutów oraz powiązań z innymi obiektami;
– zachowaniem, reprezentowanym przez operacje możliwe do wykonania na nim;
– typem, określającym budowę oraz kontekst, w którym można się do niego odwoływać.

Zbiór obiektów mających takie same właściwości to klasa. Każdy obiekt ma wartości atrybutów, które są zgodne z dziedzinami określonymi przez klasę, jak również może on uruchomić zdefiniowane przez nią operacje.

Podejście obiektowe w modelowaniu może być scharakteryzowane za pomocą takich kluczowych cech, jak [Flasiński 1997, s. 111-114; Graham 2004, s. 18-38]:
– abstrakcja – rozumiana jako uogólniony model obiektu, opisujący istotne z pewnego punktu widzenia jego cechy, a pomijający nieistotne;
– enkapsulacja – polegająca na ukrywaniu informacji o obiekcie, co jest realizowane przez wyróżnienie warstwy zewnętrznej definiującej sposób zachowania się klasy w stosunku do innych oraz warstwy wewnętrznej obejmującej opis sposobu realizacji operacji oraz opis struktur danych;
– dziedziczenie – umożliwia klasyfikację, uogólnienie oraz specjalizację. Obiekty należące do jednej klasy dziedziczą wszystkie jej własności. Podklasy dziedziczą właściwości po swoich nadklasach oraz mogą one mieć indywidualne dodatkowe własności.

4. Modelowanie wymagań z zastosowaniem metod obiektowych

Modelowanie to tworzenie uproszczonych obrazów rzeczywistości w celu lepszego jej zrozumienia. Rozpowszechnionym językiem modelowania obiektowego jest UML (*The Unified Modeling Language*), który umożliwia obrazowanie, specyfikowanie, tworzenie oraz dokumentowanie artefaktów powstających podczas tworzenia systemu informatycznego [Booch, Raumbaugh, Jacobson 2002, s. 14]. Cechuje się on otwartością i rozszerzalnością. Język ten dysponuje wieloma diagramami prezentującymi zbiór abstrakcji oraz powiązania pomiędzy nimi. Przykładowe diagramy UML to diagramy: klas, obiektów, kolaboracji, komponentów, sekwencji, aktywności oraz przypadków użycia.

Do przedstawiania wymagań funkcjonalnych wobec systemu można wykorzystać przypadki użycia. Przypadek użycia można określić jako „sekwencje czynności, które system wykonuje, aby uzyskać obserwowalny wynik przynoszący korzyść dla konkretnego aktora, reprezentującego zewnętrzne role odgrywane przez użytkow-

ników będących w interakcji z systemem" [Kruchten 2007, s.80]. Aktorami mogą być użytkownicy, systemy zewnętrzne lub zdarzenia czasowe. Przypadek użycia reprezentuje klasę podobnych scenariuszy. Scenariusze są wykorzystywane do wyodrębniania i przedstawiania charakterystycznych czynności albo wątków. Wybór realizowanego scenariusza zależy od: danych przekazywanych przez aktora, stanu wewnętrznego systemu, przekroczenia dopuszczalnego czasu reakcji aktora albo błędu.

Model przypadków użycia jest zbiorem diagramów i opisów, dokumentujących oczekiwania dotyczące interakcji z systemem. Do reprezentowania przypadków użycia można wykorzystać następujące elementy:

– diagramy przypadków użycia – przedstawiające zbiór przypadków użycia, aktorów oraz związki pomiędzy nimi,
– opis przypadków – opisujący komunikację pomiędzy przypadkiem użycia a aktorem,
– scenariusz przypadku – prezentujący jedną z możliwych sekwencji zdarzeń.

Proces tworzenia modelu przypadków może składać się z następujących etapów [Płodzień, Stemposz 2003, s. 32-36]:

– opracowanie słownika pojęć dziedziny problemu;
– określenie aktorów (systemów lub osób wchodzących w interakcje z systemem);
– określenie przypadków użycia – w tym celu dla każdego z aktorów specyfikowane są zadania, a następnie grupy zadań służących osiąganiu tych samych celów są łączone w jeden przypadek użycia;
– opracowanie diagramów przypadków użycia;
– opracowanie opisów ogólnych i szczegółowych każdego z przypadków użycia (np. scenariuszów, diagramów interakcji).

Diagram przypadków użycia zawiera takie elementy, jak: przypadki użycia, aktorów, zależności oraz systemy grupujące przypadki użycia. Może on zawierać następujące relacje pomiędzy przypadkami użycia:

– Włączenie – odzwierciedlające zawieranie się jednego przypadku w innym. Może być stosowane wtedy, gdy jeden przepływ zdarzeń jest wspólny i wykorzystywany przez wiele przypadków użycia.
– Rozszerzenie – stosuje się je w celu uszczegółowienia podstawowego przypadku użycia, szczególnie, gdy istotne jest jego uzupełnienie o prezentację opcjonalnych zachowań.
– Uogólnienie/specjalizacja – przypadek może być uszczegółowieniem innego.

Szczegółowy opis przypadku użycia powinien zawierać następujące informacje:

– nazwę przypadku użycia,
– opis przypadku,
– listę scenariuszy (powodzenia, alternatywnych, wyjątków),
– listę interesariuszy – osób zainteresowanych realizacją przypadku użycia,

– listę aktorów – interesariuszy, którzy uczestniczą w bezpośrednich interakcjach z systemem,
– bodziec – zdarzenie uruchamiające realizację przypadku użycia,
– warunki wstępne – opisujące stan systemu bezpośrednio przed rozpoczęciem jego realizacji,
– wynik gwarantowany – opisujący rezultat po realizacji scenariusza wyjątku,
– wynik sukcesu – opisujący rezultat po realizacji głównego scenariusza powodzenia lub alternatywnego.

Poza opisem tekstowym do uszczegółowienia każdego przypadku użycia wykorzystać można diagram czynności oraz sekwencji. Diagram czynności służy do modelowania dynamicznych aspektów zachowania systemu, prezentuje przepływ danych pomiędzy czynnościami, daje możliwość przedstawienia sekwencyjnych lub współbieżnych czynności oraz nie prezentuje historii zmian stanów obiektów. „Czynność jest wieloetapowym działaniem wykonanym na maszynie stanowej" [Booch, Raumbaugh, Jacobson 2002, s. 268]. Skutkiem realizacji czynności jest akcja, która prowadzi do zmiany stanu systemu.

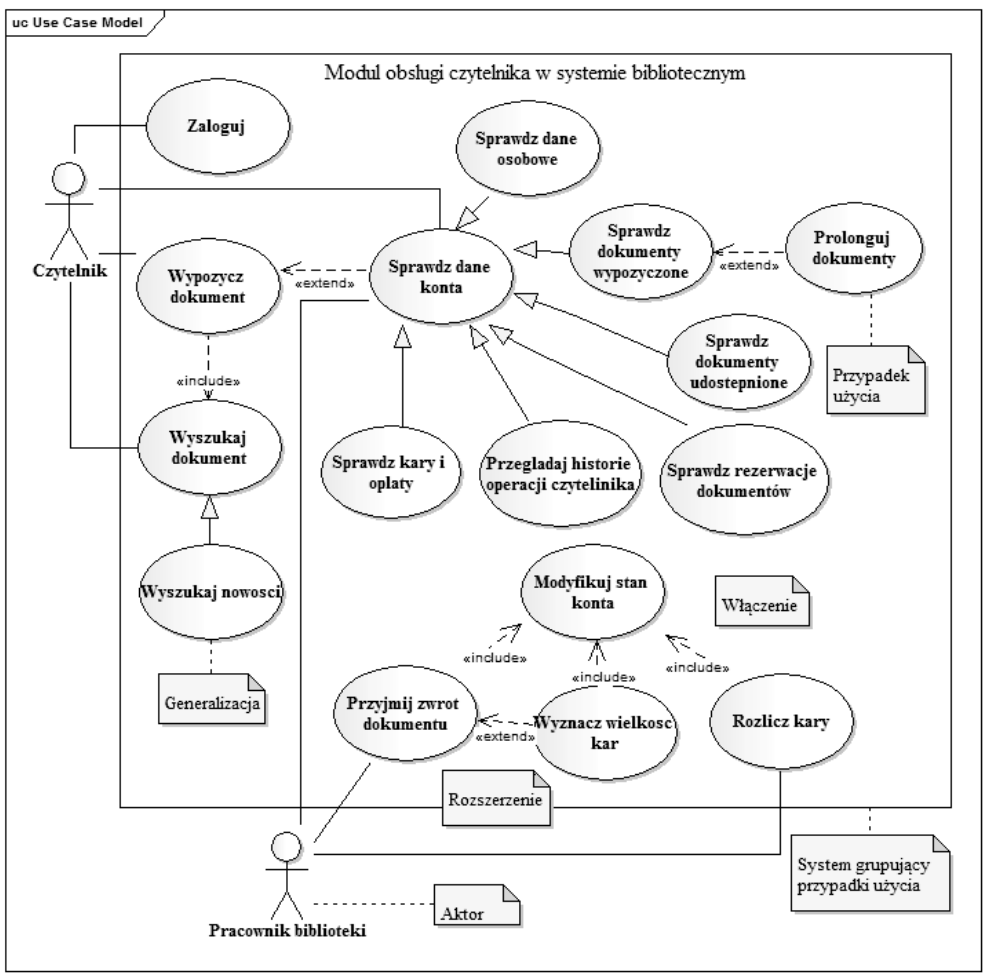
Diagram sekwencji przedstawia chronologiczny przebieg wymiany komunikatów pomiędzy aktorami. Aktorzy biorący udział w interakcjach mogą być nadawcami lub odbiorcami informacji. Komunikaty prezentują przepływ informacji pomiędzy aktorami i są uporządkowane chronologicznie.

5. Przykład modelowania wymagań z wykorzystaniem metod obiektowych

Do modelowania wymagań został zastosowany diagram przypadków użycia. Na rysunku 2 z jego wykorzystaniem zostały przedstawione wymagania funkcjonalne wobec modułu obsługi czytelnika w systemie bibliotecznym. Każde wymaganie funkcjonalne zostało odzwierciedlone w postaci przypadku użycia używanego przez wskazanego aktora oraz dodatkowo zostały zaprezentowane zależności pomiędzy poszczególnymi przypadkami.

Na diagramie przedstawiono dwóch aktorów wchodzących w interakcje z modułem, a mianowicie czytelnika oraz pracownika biblioteki. Z perspektywy czytelnika moduł obsługi powinien umożliwiać: jego logowanie do systemu, sprawdzanie konta czytelnika, wyszukiwanie dokumentów oraz ich wypożyczanie. Realizacja przypadku „Wypożycz dokument" zawiera przypadek „Wyszukaj dokument", co oznacza, iż nie można wypożyczyć dokumentu bez wcześniejszego jego wyszukania. Przypadek „Wyszukaj dokument" jest uogólnieniem przypadku „Wyszukaj nowości". Moduł ten umożliwia czytelnikowi sprawdzenie danych swojego konta (przypadek „Sprawdź dane konta"). Przede wszystkim może to obejmować sprawdzenie danych: osobowych, o dokumentach wypożyczonych, o dokumentach udostępnionych, o rezerwacjach dokumentów, dotyczących historii operacji czytelnika,

a także wielkości kar i opłat. Czytelnik dla dokumentu przez niego wypożyczonego może opcjonalnie wykonać operację jego prolongaty, co na diagramie prezentuje przypadek użycia „Prolonguj dokumenty" będący rozszerzeniem przypadku użycia „Sprawdź dokumenty wypożyczone".



Rys. 2. Diagram przypadków

Źródło: opracowanie własne.

Z perspektywy pracownika biblioteki moduł powinien umożliwiać: przyjmowanie zwrotów dokumentów, naliczanie oraz rozliczanie wysokości kar. Przypadek „Przyjmij zwrot dokumentu" może być rozszerzony o wyznaczenie wielkości kary, w przypadku gdy dozwolony termin zwrotu dokumentu został przekroczony. Przypadek „Przyjmij zwrot dokumentu" zawiera przypadek „Modyfikuj stan konta",

ponieważ przyjęcie zwrotu pomniejsza liczbę wypożyczonych dokumentów przez czytelnika. Pracownik biblioteki powinien również mieć możliwość rozliczania kar czytelnika (przypadek użycia „Rozlicz kary").

Każdy z przypadków użycia przedstawionych na rys. 2 może zostać uszczegółowiony za pomocą: słownego jego opisu, przedstawienia scenariuszy realizacji przypadku użycia, diagramu aktywności lub diagramu sekwencji.

6. Zakończenie

Duża część niepowodzeń przedsięwzięć informatycznych ma swoje źródła w fazie analizy biznesowej. Niezrozumienia wymagań użytkownika oraz niejednoznaczności ich opisu wykryte po zakończeniu fazy analizy pociągają za sobą dodatkowe koszty związane z koniecznością ponownej analizy wymagań oraz powtórzenia prac dalszych faz procesu tworzenia oprogramowania.

Przygotowanie precyzyjnych i łatwych do zrozumienia opisów wymagań może zostać osiągnięte przez zastosowanie metod analizy, których głównym celem jest tworzenie formalnego modelu. Można je opracować według własnego, autorskiego podejścia. Zastosowanie metod standardowych jest warte rozważenia z następujących powodów: oszczędności czasu koniecznego do opracowania własnych metod, mniejszych wymagań względem kompetencji osób modelujących oraz lepszego udokumentowania specyfikacji i będącego tego konsekwencją łatwiejszego ich zrozumienia [Beynon-Davies 2004, s. 227]. Do tworzenia modeli wymagań można zastosować metody obiektowe. Są one uznawane za bardziej naturalne, ponieważ rzeczywistość jest w nich przedstawiana za pomocą współpracujących ze sobą obiektów. W modelowaniu obiektowym można korzystać z języka UML. W modelowaniu wymagań użytkownika szczególne znaczenie mają diagramy przypadków użycia, prezentujące wymagania funkcjonalne poszczególnych aktorów wobec systemu. Opisu poszczególnych przypadków użycia można dokonać za pomocą strukturalizowanej tekstowej formy lub przy wykorzystaniu diagramów aktywności lub sekwencji.

Umiejętne zastosowanie zaprezentowanego podejścia do modelowania wymagań może przyczynić się do: zwiększenia szansy zapanowania nad złożonością systemu, zmniejszenia ryzyka niejednoznaczności opisu wymagań i zależności pomiędzy nimi oraz poprawy jakości oprogramowania przez zastosowanie powstałych modeli na etapie tworzenia i testów oprogramowania.

Literatura

Beynon-Davies P., *Inżynieria systemów informacyjnych*, WNT, Warszawa 2004.
Booch G., Raumbaugh J., Jacobson I., *UML przewodnik użytkownika*, WNT, Warszawa 2002.
Flasiński M., *Wstęp do analitycznych metod projektowania systemów informatycznych*, WNT, Warszawa 1997.
Graham I., *Metody obiektowe w teorii i praktyce*, WNT, Warszawa 2004.
Kruchten P., *Rational Unified Process od strony teoretycznej*, WNT, Warszawa 2007.
Nowicki A. (red.), *Komputerowe wspomaganie biznesu*, Placet, Warszawa 2006.
Pawlak M., *Zarządzanie projektami*, PWN, Warszawa 2007.
Płodzień J., Stemposz E., *Analiza i projektowanie systemów informatycznych*, Wydawnictwo PJWSTK, Warszawa 2003.
Sommerville I., *Software Engineering*, Addison Wesley, Edinburgh, 2001.
Zmitrowicz K., *Analiza biznesowa*, „Software Developer's Jurnal" nr 8(176), sierpień 2009.

REQUIREMENTS MODELING WITH THE USE OF OBJECT METHODS

Summary: The article describes how to use object-oriented methods, particularly methods of analysis for creating a model of functional requirements. It describes the process of business analysis. Next, the author characterizes the basic concepts of object-oriented methods: object, class, abstraction, encapsulation, and inheritance. To document functional requirements, use case diagrams, activity diagrams, and sequence diagrams could be applied. Finally, the author describes the application of use case diagram.