



MODELOWANIE Z WYKORZYSTANIEM UML

Anna Mroczek

Kraków 2013

PLAN

- ❑ Wprowadzenie do UML
- ❑ Modelowanie struktury statycznej
- ❑ Modelowanie zachowań i podziału
- ❑ Zaawansowane elementy UML

CZYM JEST UML?



UML oznacza *Zunifikowany Język Modelowania* (Unified Modelling Language)

UML jest językiem *wizualizacji, specyfikacji, konstrukcji i dokumentacji* artefaktów związanych z tworzeniem oprogramowania

Model UML jest wypadkową wielu *widoków* różnych aspektów systemu.

UML abstrahuje od *obiektu modelowania i metodologii modelowania*

UNIFIKOWAĆ, CZYLI ŁĄCZYĆ

UML łączy najlepsze cechy:

- ❑ modelowania **danych (EER)**
czyli - jak przedstawić informację?
- ❑ modelowania **czynności (DFD)**
czyli - co się dzieje w systemie?
- ❑ modelowania **obiekowego (OOA)**
czyli - jak zrozumiale przedstawiać świat?
- ❑ zarządzania złożonością (komponenty)
czyli - divide et impera!

A W SKRÓCIE...

- ❑ 9 typów diagramów - perspektyw
 - modelowanie wymagań
 - modelowanie struktury statycznej koncepcji
 - modelowanie zależności dynamicznych i zachowań
 - modelowanie struktury fizycznej
- ❑ mechanizm rozszerzeń
 - stereotypy (*ang. stereotypes*)
 - metki (*ang. tags*)
 - ograniczenia (*ang. constraints*)

TYPY DIAGRAMÓW

- przypadków użycia (*use-case diagram*)
- klas i obiektów (*class diagram*)
- stanu obiektów (*statechart diagram*)
- współpracy (*collaboration diagram*)
- sekwencji (*sequence diagram*)
- czynności (*activity diagram*)
- komponentów (*component diagram*)
- rozmieszczenia (*deployment diagram*)

HISTORIA UML

Dawno temu...

- ❖ różnorodne, niespójne metodyki obiektowe
(*OMT, OOSE, Fusion, OOA/OOD*)

TRZEJ AMIGOS



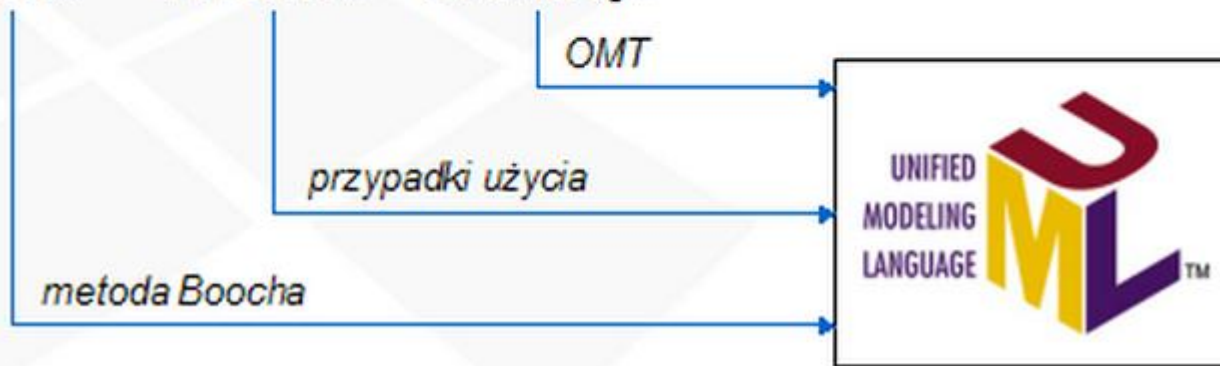
G. Booch



I. Jacobson



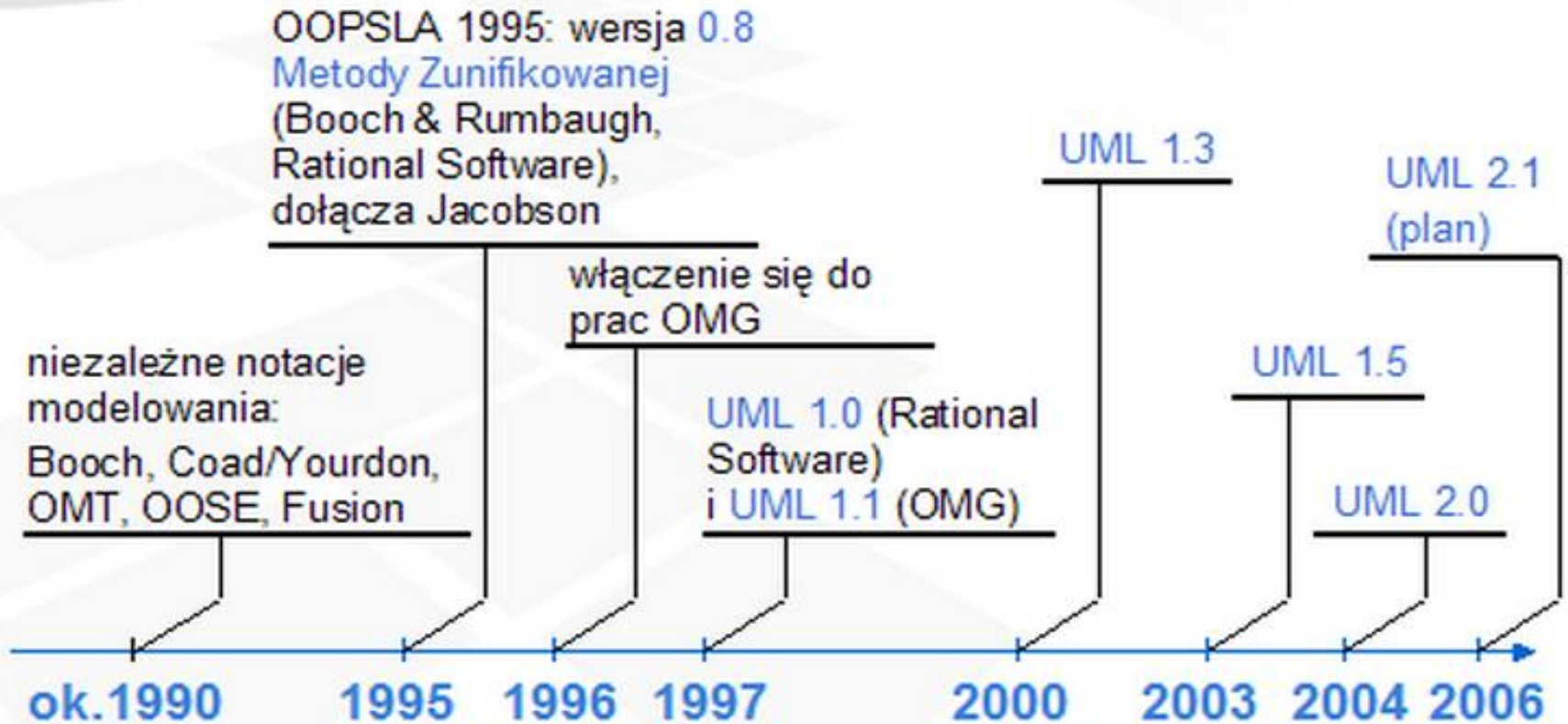
J. Rumbaugh



Trzej autorzy UMLa: Booch, Jacobson i Rumbaugh zunifikowali swoje notacje, tworząc jedną metodę.

Z poszczególnych notacji przejęto najlepsze rozwiązania.

HISTORIA UML



HISTORIA UML

Obecnie

– wersja 2.4.1 UML (znormalizowana w kwietniu 2012 roku)

<http://www.uml.org>

UNIFIED MODELING LANGUAGE

- ❑ graficzny język obiektowego modelowania systemów (oprogramowania), pozwalający na reprezentację różnych perspektyw (aspektów) modelowania
- ❑ rodzina notacji oparta na jednym metamodelu
- ❑ graficzna notacja- wspomagająca komunikację i dokumentację
- ❑ odwołanie do wspólnego metamodelu pozwala na specyfikację oprogramowania
- ❑ rozbudowany aparat formalny jest efektem kompromisów związanych ze standaryzacją- w praktyce wykorzystujemy 20% języka, które pozwala nam wykonać 80% pracy

ZASTOSOWANIA UML

- ❑ w projektowaniu systemów (forward/reverse engineering) jako opis koncepcji (szkic) – mniej formalny, obejmuje wybrane aspekty, ma charakter poznawczy, dynamicznie się zmienia, nie wymaga zaawansowanych narzędzi
- ❑ jako specyfikacja (projekt) systemu – sformalizowany, kompletny (ze szczegółami realizacji), specjalizowane narzędzia CASE
- ❑ do konstrukcji systemów
- ❑ MDA = Model-Driven Architecture
- ❑ xUML = executable UML

WSPARCIE

UML Partners Consortium

udział największych firm produkujących oprogramowanie:

- DEC, HP, Microsoft, IBM, Oracle, Texas Instruments

oraz producenci CASE

- Rational Software
- Visual Paradgim

APLIKACJE WSPOMAGAJĄCE

□ Darmowe:

- **ArgoUML** - napisany w Javie, zaawansowane generowanie kodu i odpowiedzi, ciągle tworzony,
- **Dia** - ogólne narzędzie do rysowania diagramów,
- **UML Sculptor** - prosty, łatwy w użyciu program do tworzenia diagramów klas,
- **Umbrello** - program dla Linuksa, część KDE,
- **StarUML**.

□ Komercyjne:

- **Borland Together** - rodzina programów integrujących się z różnymi IDE, jest wersja darmowa,
- **Poseidon for UML** - zaawansowane narzędzie bazujące na ArgoUML, darmowa edycja Community,
- **Enterprise Architect** - Profesjonalne narzędzie w przystępnej cenie o wygodnym interfejsie działające na platformach Windows i Linux. Wspiera UML 2.0,
- **Visual Paradigm for UML**,
- **IBM Rational Rose**.

CYKL TWORZENIA OPROGRAMOWANIA

UML jest **niezależny od procesu** ale twórcy sugerują proces;

- ukierunkowany na **przypadki użycia**
- zorientowany na **architekturę**
- **iteracyjny i przyrostowy**

PRZYKŁAD

- Biblioteka prowadzi wypożyczalnię wydawnictw: książek i czasopism. Korzystają z niej czytelnicy.
- Wszystkie wydawnictwa mogą występować w wielu egzemplarzach.
- Czytelnicy mogą rezerwować i odwoływać rezerwacje na wydawnictwa.
- Książka może być dostępna, wypożyczona, zaginiona, lub zniszczona.

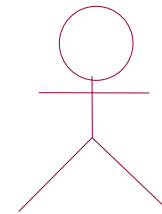
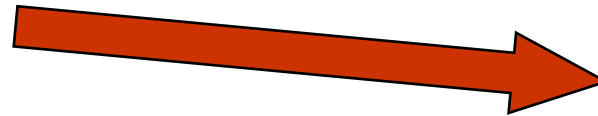
STATYCZNE ZACHOWANIE SYSTEMU

Jak modelować funkcjonalność

czyli o przypadkach użycia

AKTOR, CZYLI DZIAŁACZ

Aktor to ktoś (coś), kto (co) musi współdziałać z modelowanym systemem.



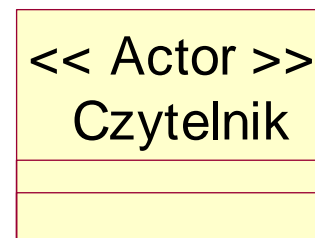
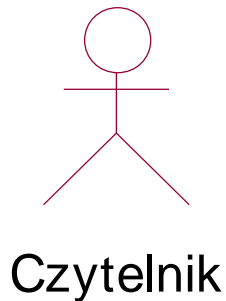
Aktor

AKTOR W UML

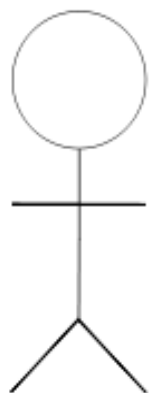
**Aktor w UML jest klasą (nie obiektem!)
o nadanym stereotypie *Actor*.**

Można go oznaczać poprzez

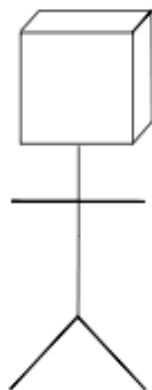
- ikonę
- klasę ze stereotypem



Aktor osobowy



System zewnętrzny



Aktor nieosobowy



Urządzenie



Czas

Przykłady nazw

Konsultant
Dział sprzedaży

System rezerwacji pokoi
System sporządzania zestawień

Nagrywarka DVD-RAM
Kiosk multimedialny

Termin płatności
Ostatni dzień miesiąca

PRZYPADEK UŻYCIA

Przypadek użycia (*use-case*)

- jest sposobem, w jaki aktorzy używają (chcą używać) systemu
- jest podstawową jednostką funkcjonalności
- definiuje wymagania

Czego potrzebują użytkownicy?

- Bibliotekarz...
- Czytelnik...

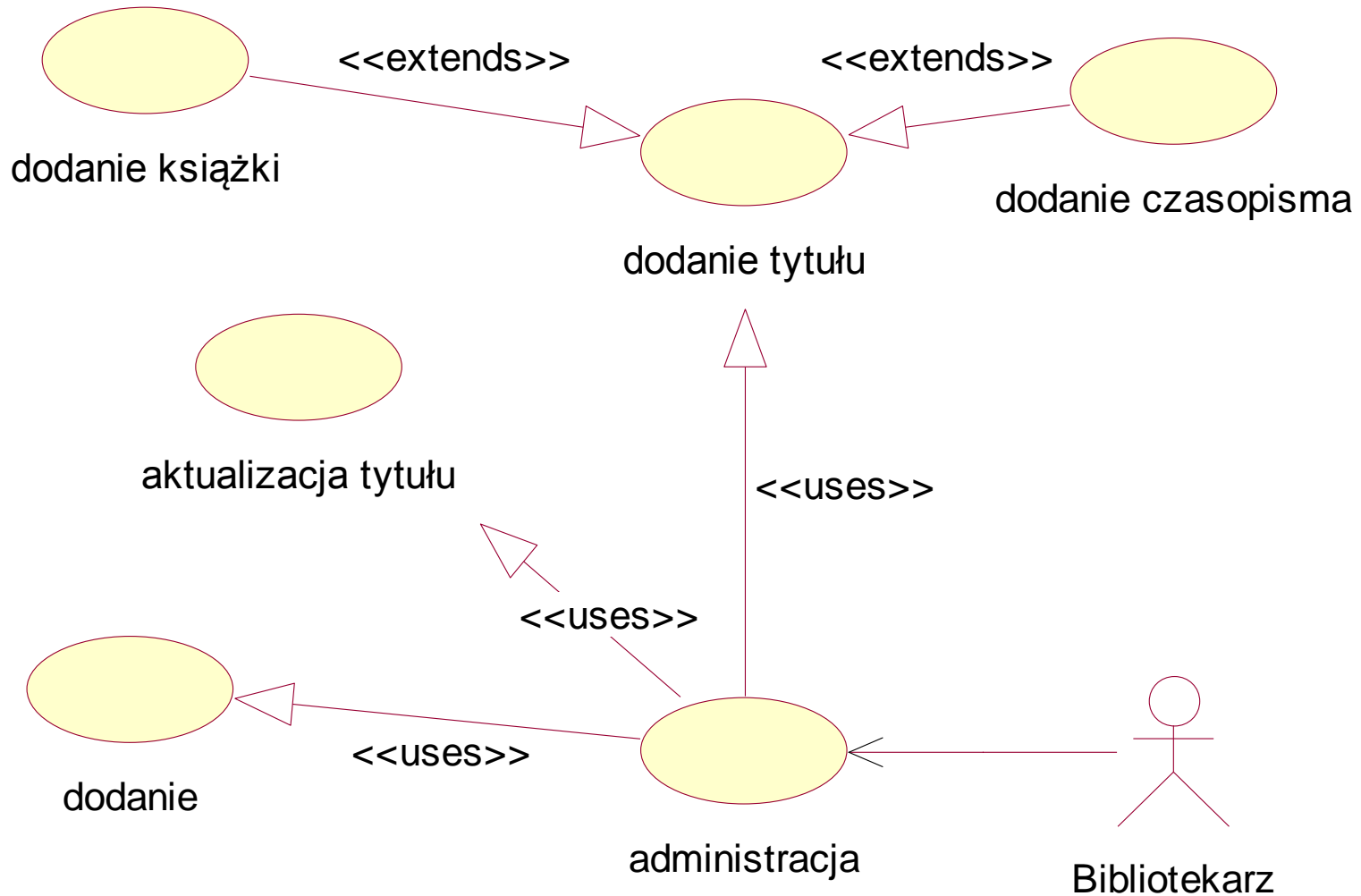
DIAGRAM *USE-CASE*

Definiuje

- **granice** systemu, *czyli jak daleko sięga model*
- jego **kontekst**, *czyli co pozostaje na zewnątrz*
- **użytkowników** systemu, *czyli aktorów*
- **funkcje** systemu
- **zależności** między użytkownikami i funkcjami

... i jest czytelny dla odbiorcy!

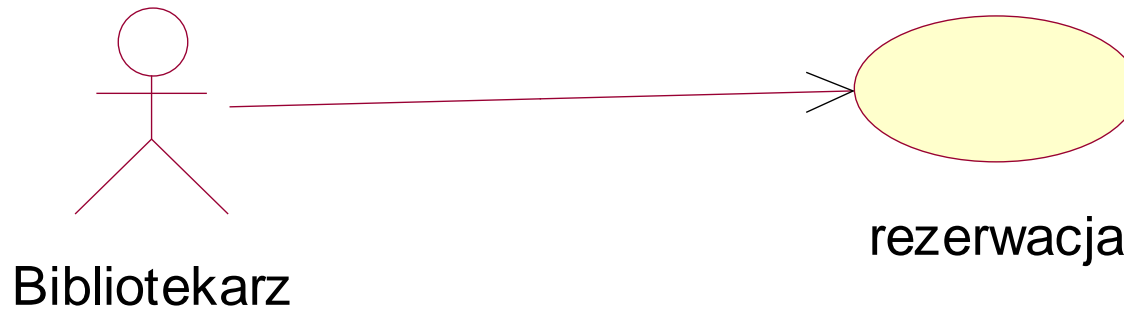
PRZYKŁAD DIAGRAMU *USE-CASE*



UŻYCIE FUNKCJI

Aktor używa funkcji (wykonuje funkcję)

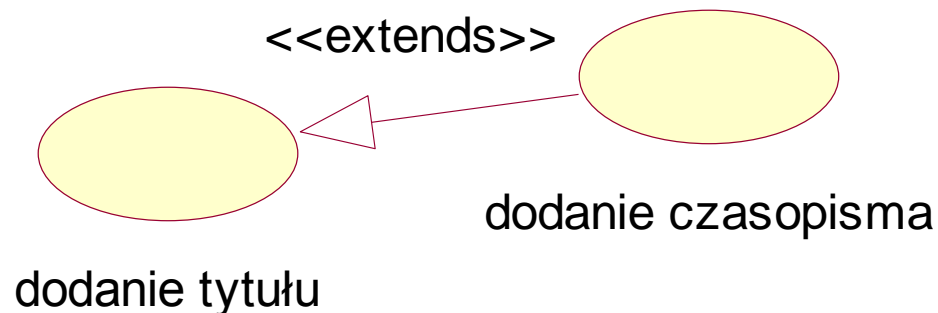
- domyślny stereotyp <<communicates>>
- od użytkownika do funkcji



ZALEŻNOŚCI MIĘDZY FUNKCJAMI (CD.)

Funkcja uszczegóławia funkcję

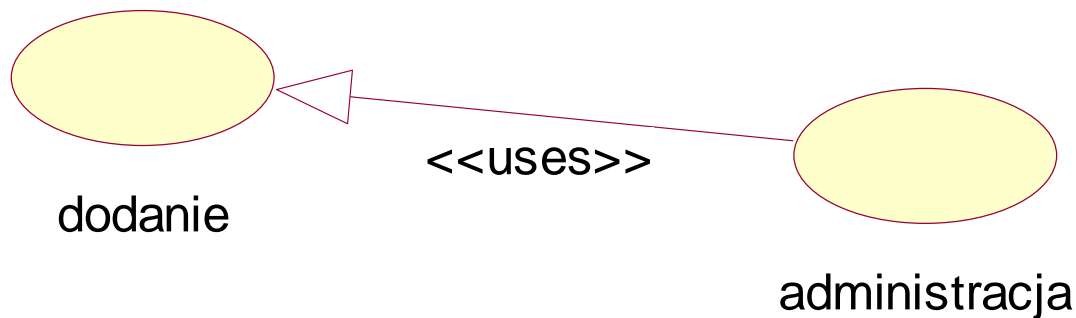
- relacja dziedziczenia
- stereotyp <<extends>>
- funkcje abstrakcyjna
- od funkcji szczegółowej do funkcji ogólnej

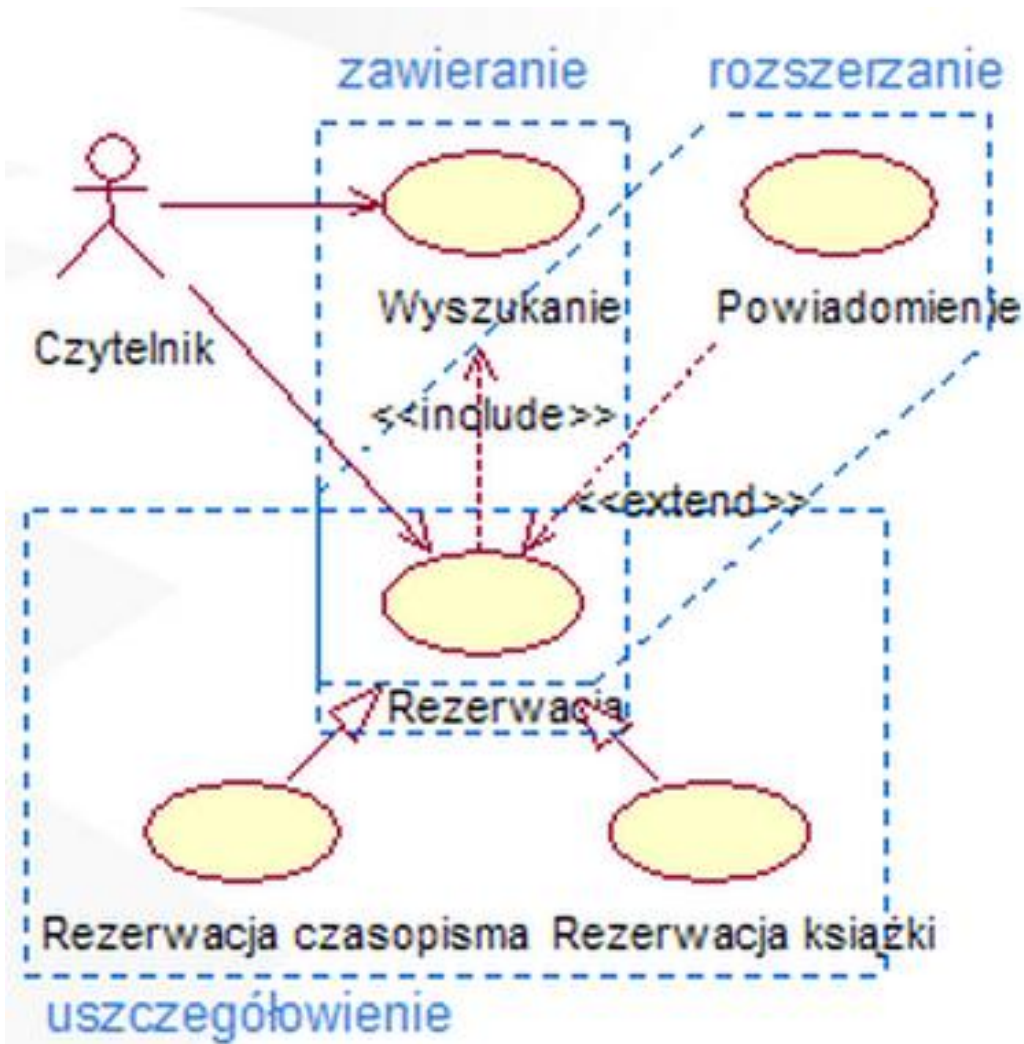


ZALEŻNOŚCI MIĘDZY FUNKCJAMI (CD.)

Funkcja wywołuje inną funkcję

- relacja zależności funkcji
- ponowne użycie funkcji/komponentu
- stereotyp `<<uses>>`
- od funkcjiwołającej do funkcjiwołanej





ZASTOSOWANIE

- ❑ Analiza systemów (podsystemów, klas, interfejsów)
- ❑ Podstawa do opracowania przypadków testowych
- ❑ Źródło testów regresyjnych (dla podsystemów) oraz testów integracyjnych i systemowych dla systemu
- ❑ Pomagają w skrótnym łączeniu informacji z profilu użytkownika, reguł gospodarczych i wymagań wobec formatu danych
- ❑ Pomagają w opracowaniu dokumentacji wymagań a także planu przedsięwzięcia (daty wersji, priorytety, stan wytwarzania etc.)
- ❑ Pomagają w projektowaniu interfejsu użytkownika

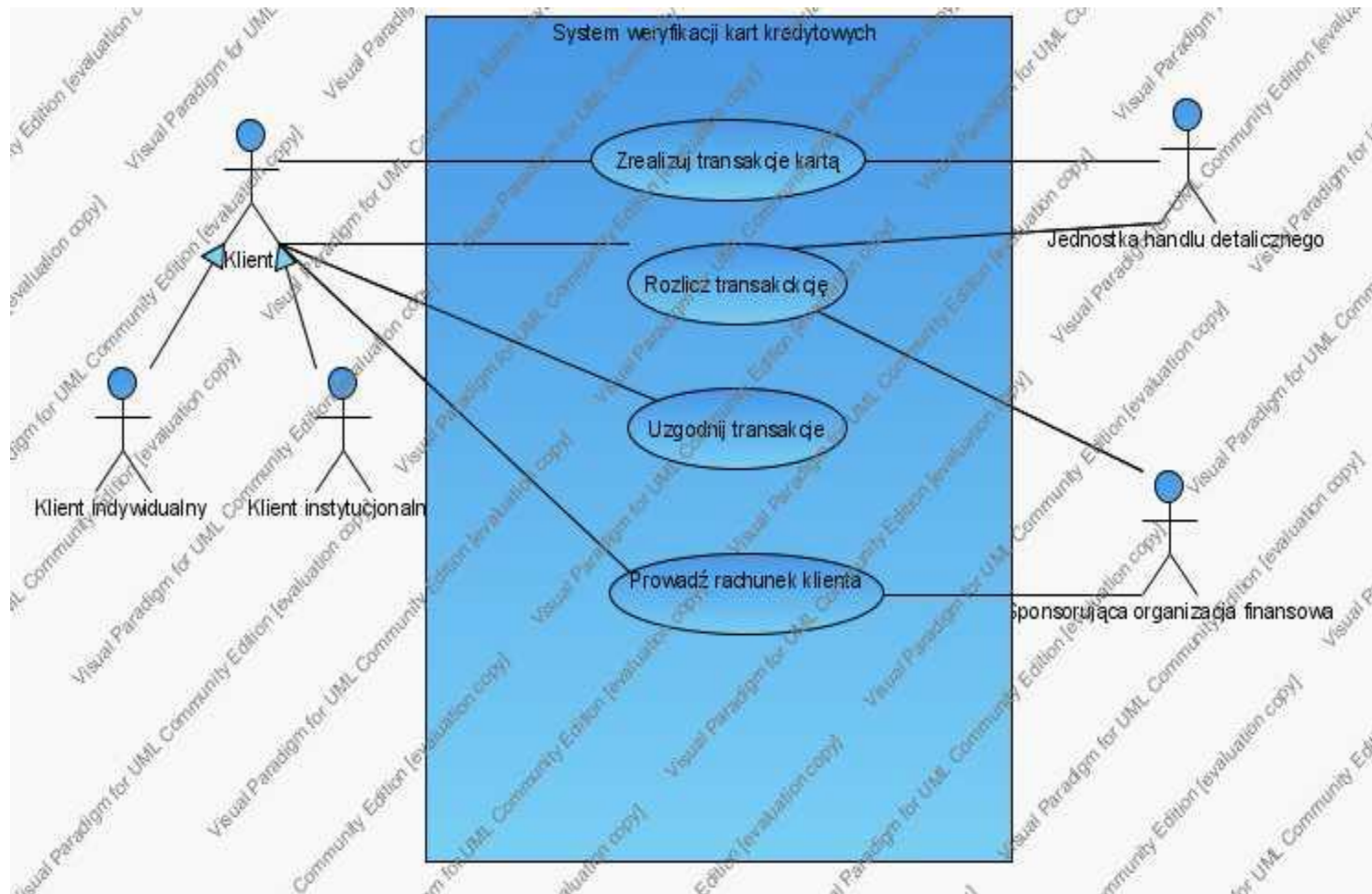
MODELOWANIE OTOCZENIA

Otoczenie systemu to wszystkie byty istniejące poza systemem a będące z nim w interakcji—uwzględniamy tylko tych aktorów, którzy są niezbędni do właściwego działania systemu

Postępowanie:

- Identyfikacja aktorów działających wokół systemu
- Uporządkowanie aktorów za pomocą generalizacji
- W razie potrzeby dodanie stereotypów

MODELOWANIE OTOCZENIA – PRZYKŁAD



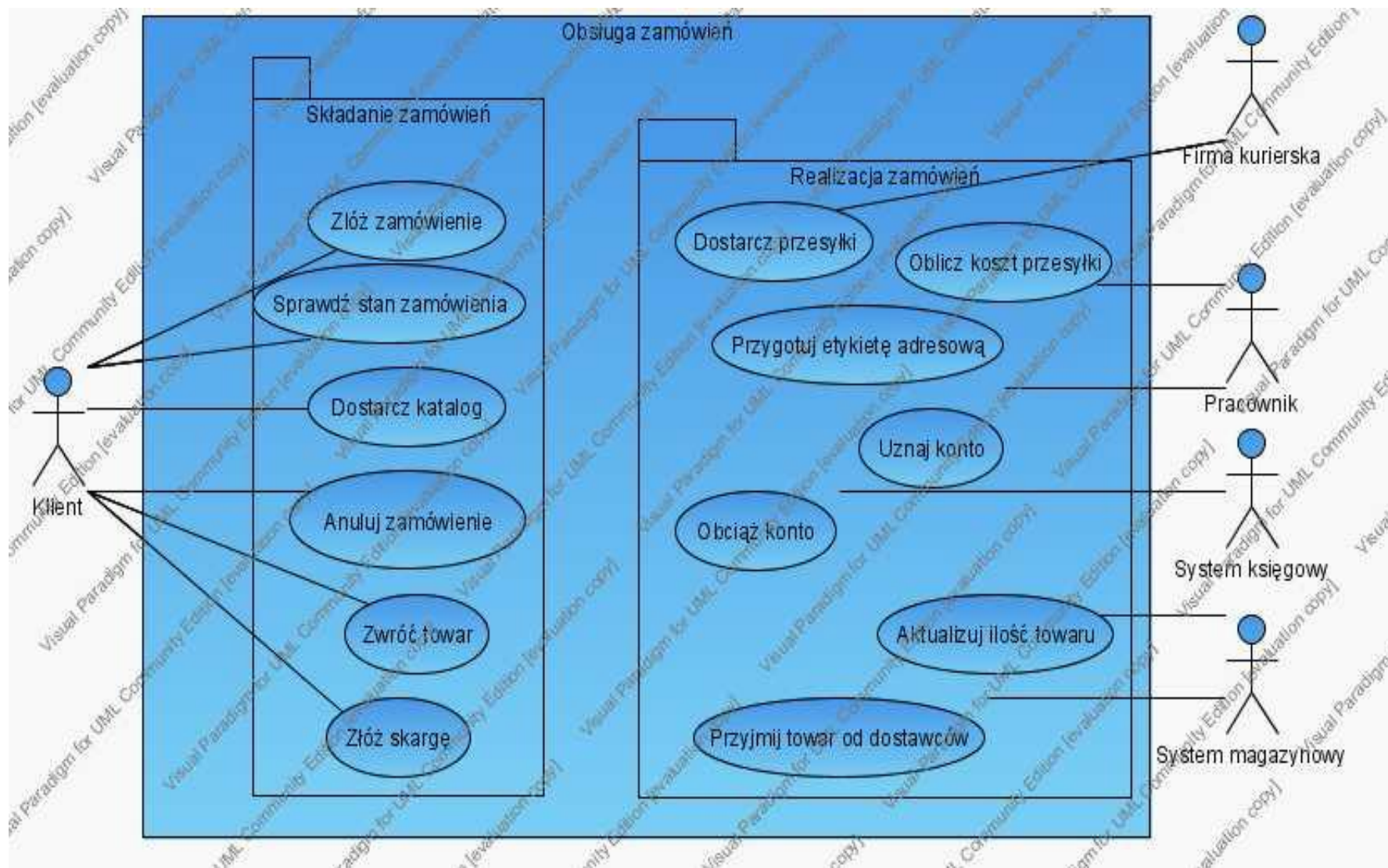
MODELOWANIE WYMAGAŃ

Wymaganie to element projektu. To właściwość lub zachowanie systemu.

To swoisty kontrakt między bytami z otoczenia a samym systemem określający co system ma robić (bez zwracania uwagi jak system ma to robić)

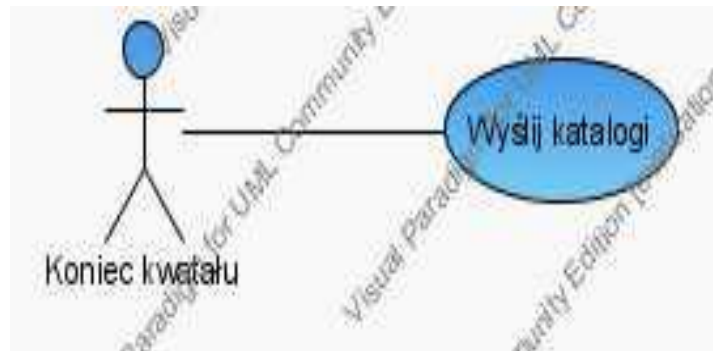
ZASADY MODELOWANIA

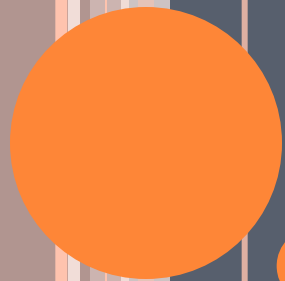
- 1) Określ otoczenie systemu dla każdego aktora rozważ działania których on oczekuje lub wymaga od systemu – zapisz je w postaci przypadków użycia.
- 2) Wyłącz powtarzające się fragmenty działań i utwórz z nich nowe przypadki które będą „włączane” przez inne przypadki.
- 3) Wydziel warianty działań i umieść je w nowych przypadkach użycia, które będą „rozszerzać” ciągi zdarzeń innych przypadków użycia.
- 4) Uwzględnij na diagramie związki pomiędzy aktorami a przypadkami użycia.
- 5) Dodaj notatki uwzględniające wymagania niefunkcjonalne.



ZALEŻNOŚCI CZASOWE

- System obsługi zamówień co kwartał automatycznie wysyła katalogi do klientów
- Czas jako aktor
- Czas jako część systemu





PRZYKŁAD

35

WYPOŻYCZALNIA/ BIBLIOTEKA

- Aktorzy
- Przypadki użycia
- Relacje

RYSUNKI TO NIE WSZYSTKO

- ❑ **Aktor główny:** Nabywca
- ❑ **Zakres:** pakiet Osobisty doradca finansowy
- ❑ **Poziom Celu:** użytkownika
- ❑ **Uczestnicy i interesy:**
 - **Nabywca** – chce kupić akcje i mieć je automatycznie w elektronicznym portfelu
 - **Biuro maklerskie** – chce otrzymać pełną informację o zakupie
- ❑ **Warunek początkowy:** Użytkownik uruchomił już ODF
- ❑ **Minimalna gwarancja:** W dziennikach będzie istnieć informacja wystarczająca aby ODF mógł wykryć, że coś jest nie w porządku i poprosić użytkownika o szczegóły
- ❑ **Gwarancja powodzenia:** Zdalny serwis WWW potwierdził zakup; zaktualizowano dzienniki i portfel użytkownika

❑ Główny scenariusz powodzenia:

1. Nabywca postanawia kupić akcje przez WWW
2. ODF pobiera od użytkownika informacje o serwisie WWW z którego ma skorzystać
3. ODF uruchamia połączenie WWW z tym serwisem i zachowuje sterowanie
4. Nabywca przegląda i kupuje akcje z serwisu WWW
5. ODF przechwytuje reakcje serwisu WWW i aktualizuje portfel nabywcy
6. ODF wyświetla użytkownikowi nowy stan portfela

- 2a. Nabywca chce skorzystać z serwisu WWW, którego ODF nie obsługuje
 - 2a1 System pyta Użytkownika o nową sugestię z możliwością przerwania przypadku Użycia

- 3a Dowolna awaria WWW w czasie uruchamiania
 - 3a1 System informuje nabywcę o awarii i wyświetla podpowiedź, po czym cofa się do poprzedniego kroku
 - 3a2 Nabywca rezygnuje - kończy przypadek Użycia albo próbuje ponownie

- 4a Komputer ulega awarii lub jest wyłączany w trakcie transakcji
 - 4a1 ???
 - 4b Serwis WWW nie potwierdza zakupu, ale odkłada go na później
 - 4b1 ODF zapisuje opóźnienie do dziennika, ustawia zegar aby zapytać nabywcę o wynik

- 5a Serwis WWW nie przekazuje potrzebnych informacji o zakupie
 - 5a1 ODF umieszcza w dzienniku zapis o braku informacji, klient musi zaktualizować wątpliwy zakup

- ❑ **Aktor główny:** Zgłaszający szkodę
- ❑ **Zakres:** Firma ubezpieczeniowa
- ❑ **Poziom:** streszczenie
- ❑ **Uczestnicy i interesy:**
 - **Zgłaszający szkodę** – otrzymać jak najwyższe odszkodowanie
 - **FU** – wypłacić jak najniższe odszkodowanie
 - **Urząd nadzoru** – dopilnować aby wszystkie przepisy były przestrzegane
- ❑ **Warunek początkowy** – brak
- ❑ **Minimalna gwarancja:** FU rejestruje zgłoszenie szkody i wszystkie czynności
- ❑ **Gwarancja powodzenia:** Zgłaszający szkodę i FU uzgadniają kwotę do wypłaty. Zgłaszający przedstawia szkodę

- Główny scenariusz powodzenia
- 1. Zgłaszający szkodę przedstawia roszczenie i podaje wszystkie istotne dane
- 2. FU stwierdza że zgłaszający ma ważną polisę
- 3. FU wyznacza likwidatora, który zajmie się szkodą
- 4. FU stwierdza, że wszystko jest zgodne z postanowieniami polisy
- 5. FU wypłaca odszkodowanie i zamyka sprawę

1a. Przekazane dane są niepełne:

1a1 FU prosi o brakujące informacje

1a2 Zgłaszający dostarcza brakujących informacji

2a Zgłaszający szkodę nie ma ważnej polisy:

2a1 FU odrzuca zgłoszenie, informuje o tym zgłaszającego, wszystko to rejestruje i kończy przypadek użycia

3a W tej chwili nie ma wolnych likwidatorów

3a1 ???

4a Okoliczności wypadku naruszają podstawowe postanowienia polisy:

4a1 FU odrzuca zgłoszenie, informuje o tym zgłaszającego, wszystko to rejestruje i kończy przypadek użycia

4b Okoliczności wypadku naruszają pewne mniej istotne postanowienia polisy:

4b1 FU rozpoczyna negocjacje ze zgłaszającym szkodę w sprawie ustalenia kwoty wypłaty

OZNACZENIA I DEFINICJE

- ❑ **Zakres** – wskazuje system który analizujemy
- ❑ **Warunki początkowe i gwarancje:** co musi być prawdą przed i po wykonaniu przypadku użycia
- ❑ **Główny scenariusz powodzenia** – ciąg zdarzeń w sytuacji gdy wszystko idzie dobrze
- ❑ **Rozszerzenia** – co może pójść inaczej w trakcie tego scenariusza
 - Liczby w rozszerzeniach odnoszą się do numerów kroków w głównym scenariuszu powodzenia w których wykrywane są odmienne warunki
 - Gdy przypadek użycia obejmuje (włącza) inny przypadek – przywoływany przypadek użycia jest podkreślany

KUP COŚ – WERSJA NIEFORMALNA

Zamawiający przygotowuje zlecenie i wysyła je do swojego Akceptującego. Akceptujący sprawdza czy w budżecie są jeszcze środki, sprawdza cenę towarów, wypełnia Ządanie dostawy i wysyła je do Kupującego. Kupujący sprawdza stan magazynu i znajduje najlepszego dostawcę towarów. Kontroler weryfikuje podpis Akceptującego. Kupujący wypełnia Ządanie zamówienia i rozpoczyna PO z Dostawcą. Dostawca dostarcza towary do Odbioru, odbiera pokwitowanie dostawy (poza zakresem projektowanego systemu). Odbiorca rejestruje dostawę i wysyła towary do Zamawiającego. Zamawiający oznacza zlecenie jako zrealizowane.

Przed odbiorem towarów Zamawiający w każdej chwili może zmienić lub anulować zlecenie. Anulowanie zlecenia powoduje zaprzestanie jakiegokolwiek aktywnego przetwarzania (usunięcie z systemu??). Zmniejszenie ceny nie wpływa na przetwarzanie. Wzrost ceny powoduje wysłanie zlecenia z powrotem do Akceptującego

KUP COŚ – WERSJA FORMALNA

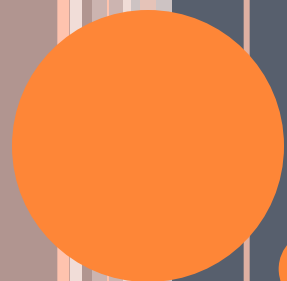
- ❑ **Aktor główny** – zamawiający
- ❑ **Zakres** – Przedsiębiorstwo – całościowy system zakupów, elektroniczny i nie elektroniczny postrzegany przez pracowników
- ❑ **Poziom:** streszczenie
- ❑ **Uczestnicy i interesy:**
 - **Zamawiający:** Chce tego co zamówił, łatwego otrzymania tego co zamówił
 - **Firma:** Chce kontrolować wydatki, ale chce umożliwić potrzebne zakupy
 - **Dostawca:** Chce otrzymać zapłatę za dostarczone towary
- ❑ **Warunek początkowy:** Brak
- ❑ **Minimalna gwarancja:** Każde wysłane na zewnątrz zamówienie jest zaakceptowane przez uprawnionego Akceptującego. Zamówienie jest tak kontrolowane aby firma mogła być obciążona jedynie za towary dostarczone zgodnie z zamówieniem

- ❑ **Gwarancja powodzenia:** Zamawiający ma towary i prawidłowy budżet, który można obciążyć
- ❑ **Wyzwalacz:** zamawiający postanawia coś kupić
- ❑ **Główny scenariusz powodzenia**
 1. Zamawiający przygotowuje zlecenie
 2. Akceptujący sprawdza środki w budżecie, sprawdza cenę towarów, wypełnia żądanie dostawy
 3. Kupujący sprawdza stan magazynu i znajduje najlepszego dostawcę towarów
 4. Kontroler weryfikuje podpis Akceptującego
 - Kupujący wypełnia żądanie zamówienia, rozpoczyna PO z dostawcą
 - Dostawca dostarcza towar do odbiorcy, odbiera pokwitowanie dostawy (poza zakresem projektowanego systemu)
 - Odbiorca rejestruje dostawę, wysyła towary do zamawiającego
 - Zamawiający oznacza zlecenie jako zrealizowane

- 1a** Zamawiający nie zna dostawcy lub ceny: pozostaw te części puste i kontynuuj
- 1b** Przed odbiorem towarów zamawiający w każdej chwili może zmienić lub anulować zlecenie
- Anulowanie zlecenia powoduje zaprzestanie jakiegokolwiek aktywnego przetwarzania
 - Zmniejszenie ceny nie ma wpływu na przetwarzanie
 - Zwiększenie ceny powoduje wysłanie zlecenia z powrotem do Akceptującego
- 2a** Akceptujący nie zna dostawcy lub ceny: pozostaw te części puste i kontynuuj
- 2b** Akceptujący nie jest szefem zamawiającego: w porządku, pod warunkiem, że podpisze to kontroler
- 2c** Akceptujący odmawia: wyślij zlecenie z powrotem do Zamawiającego, by je zmienił lub cofnął

- 3a Kupujący znajduje towary w magazynie: wyślij je, zmniejsz zlecenie o ich liczbę i kontynuuj
- 4a Kontroler odmawia Akceptującemu: wyślij z powrotem do zamawiającego i wycofaj to z aktywnego przetwarzania
- 5a Zlecenie obejmuje kilku Dostawców: Kupujący generuje kilka PO
- 5b Kupujący łączy kilka zleceń: zastosuj taki sam proces, ale oznacz PO numerami połączonych zleceń
- 6a Dostawca nie dostarcza na czas: System alarmuje o Niedostarczeniu
- 7a Częściowa dostawa: Odbiorca zaznacza na PO częściową dostawę i kontynuuje
- 7b Częściowa dostawa z wielu PO: Odbiorca przyporządkowuje wielkości do zleceń i kontynuuje
- 8a Nie te towary lub nie tej jakości: Zamawiający odrzuca dostarczone towary
- 8b Zamawiający zwolnił się z firmy: Kupujący kontaktuje się z szefem Zamawiającego: zmień Zamawiającego, zwróć towary lub anuluj zlecenie

- ❑ **Lista wariantów technologii i danych:** Brak
- ❑ **Priorytet:** Zmienny
- ❑ **Wersje:** Kilka
- ❑ **Czas reakcji:** Zmienny
- ❑ **Częstotliwość Użycia:** 3 razy dziennie
- ❑ **Kanał aktora głównego:** przeglądarka WWW, system pocztowy lub inny równoważny
- ❑ **Aktorzy drugoplanowi:** Dostawca
- ❑ **Otwarte zagadnienia:**
 - Kiedy usunąć z systemu anulowane zamówienie
 - Jakie uprawnienia są potrzebne do anulowania zlecenia
 - Kto może zmienić treść zlecenia
 - Jaka historia zmian musi być przechowywana i dostępna na żądanie
 - Co się dzieje gdy Zamawiający odrzuca dostarczone towary
 - W jaki sposób przy zamawianiu spytać o stan i skorzystać z wewnętrznego magazynu



TO JUŻ JEST KONIEC...



...na dziś 😊