

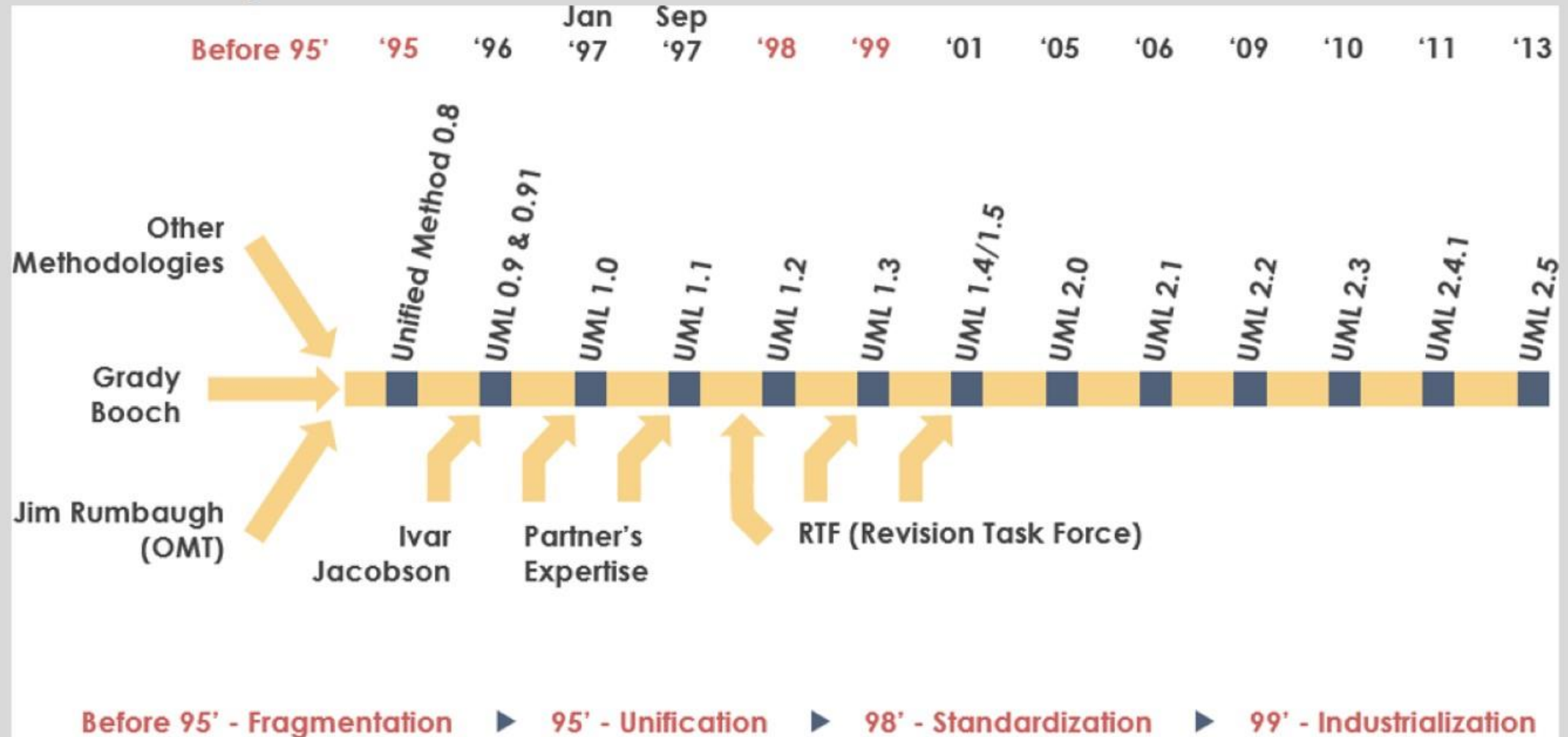
# Narzędzia typu CASE

- CASE – Computer-Aided Software Engineering = Wspomagana komputerowo inżynieria oprogramowania
- Oprogramowanie, które umożliwia wspieranie realizacji kolejnych zadań w ramach procesu tworzenia systemów informatycznych.
- Wspomaga w opisie wycinka rzeczywistości w wersji elektronicznej
- Mogą być freeware i shareware – w zależności od wersji, różna ilość dostępnych funkcji

# Podstawowe elementy narzędzi CASE

- Słownik danych
- Moduł inżynierii odwrotnej
- Narzędzia RAD (rapid application development)
- Moduł import/eksport
- Moduł kontroli poprawności
- Moduł kontroli jakości
- Edytory diagramów
- Moduł projektowania interfejsów
- Moduł pracy sieciowej
- Moduł zarządzania pracą grupową
- Moduł zarządzania wersjami
- Generator raportów
- Generator dokumentacji technicznej
- Generatory kodów

# Rozwój UML

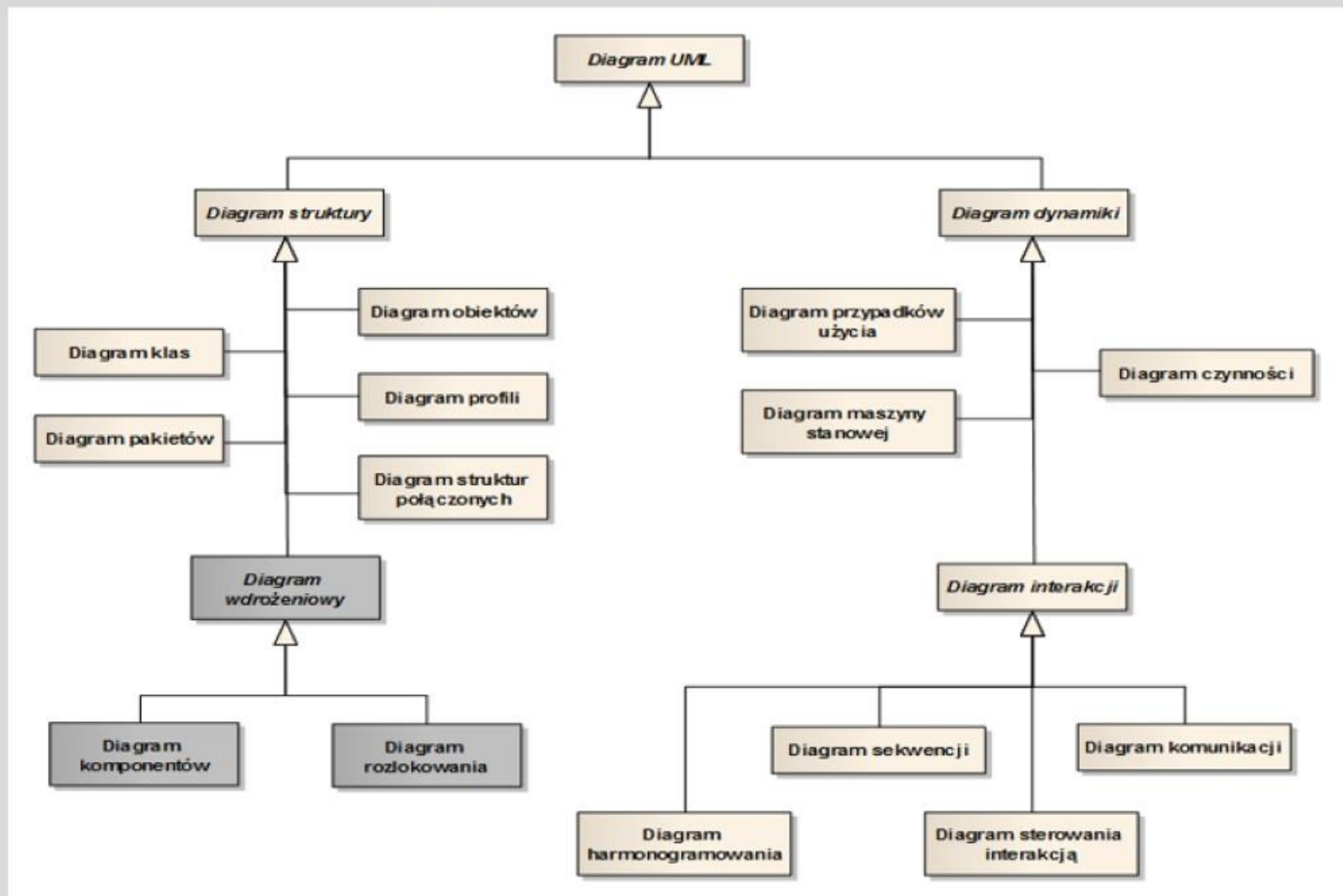


## HISTORY

### FORMAL VERSIONS

VERSION	ADOPTION DATE	URL
2.5.1	grudnia 2017	<a href="https://www.omg.org/spec/UML/2.5.1/">https://www.omg.org/spec/UML/2.5.1/</a>
2.4.1	lipca 2011	<a href="https://www.omg.org/spec/UML/2.4.1/">https://www.omg.org/spec/UML/2.4.1/</a>
2.3	maja 2010	<a href="https://www.omg.org/spec/UML/2.3/">https://www.omg.org/spec/UML/2.3/</a>
2.2	stycznia 2009	<a href="https://www.omg.org/spec/UML/2.2/">https://www.omg.org/spec/UML/2.2/</a>
2.1.2	października 2007	<a href="https://www.omg.org/spec/UML/2.1.2/">https://www.omg.org/spec/UML/2.1.2/</a>
2.0	lipca 2005	<a href="https://www.omg.org/spec/UML/2.0/">https://www.omg.org/spec/UML/2.0/</a>
1.5	marca 2003	<a href="https://www.omg.org/spec/UML/1.5/">https://www.omg.org/spec/UML/1.5/</a>
1.4	września 2001	<a href="https://www.omg.org/spec/UML/1.4/">https://www.omg.org/spec/UML/1.4/</a>
1.3	lutego 2000	<a href="https://www.omg.org/spec/UML/1.3/">https://www.omg.org/spec/UML/1.3/</a>
1.2	lipca 1999	<a href="https://www.omg.org/spec/UML/1.2/">https://www.omg.org/spec/UML/1.2/</a>
1.1	grudnia 1997	<a href="https://www.omg.org/spec/UML/1.1/">https://www.omg.org/spec/UML/1.1/</a>

# Struktura diagramów UML



# Język SysML

- Język modelowania ogólnego przeznaczenia
- Służy do:
  - Specyfikowania
  - Analizy
  - Projektowania
  - Weryfikacji złożonych systemów
- Jest graficznym językiem modelowania, oparty ma semantyce, umożliwiające reprezentowanie wymagań, dynamiki, struktury oraz cech systemu
- Wykorzystywany jest w inżynierii systemów – wielodyscyplinarne podejście do przekształcania zestawu potrzeb i wymagań interesariuszy w zharmonizowane rozwiązanie systemowe, zaspokajające te wymagania.
- Wywodzi się z języka modelowania UML
- Obecna wersja języka w wersji 1.6 jest z grudnia 2019 r.



# DIAGRAM WYMAGAŃ SYSTEMOWYCH

Wymagania są wyrażonymi w sposób formalny potrzebami klienta – funkcjonalnościami lub cechami, które system winien spełniać.

Pozyskiwanie wymagań stanowi podstawę całego procesu budowy systemów, a od rezultatów tego etapu uzależniony jest dalszy sposób realizacji projektu; dobrze określone wymagania zapewniają lepszą jakość przyszłego oprogramowania i, w konsekwencji, wyższy poziom satysfakcji zamawiającego.

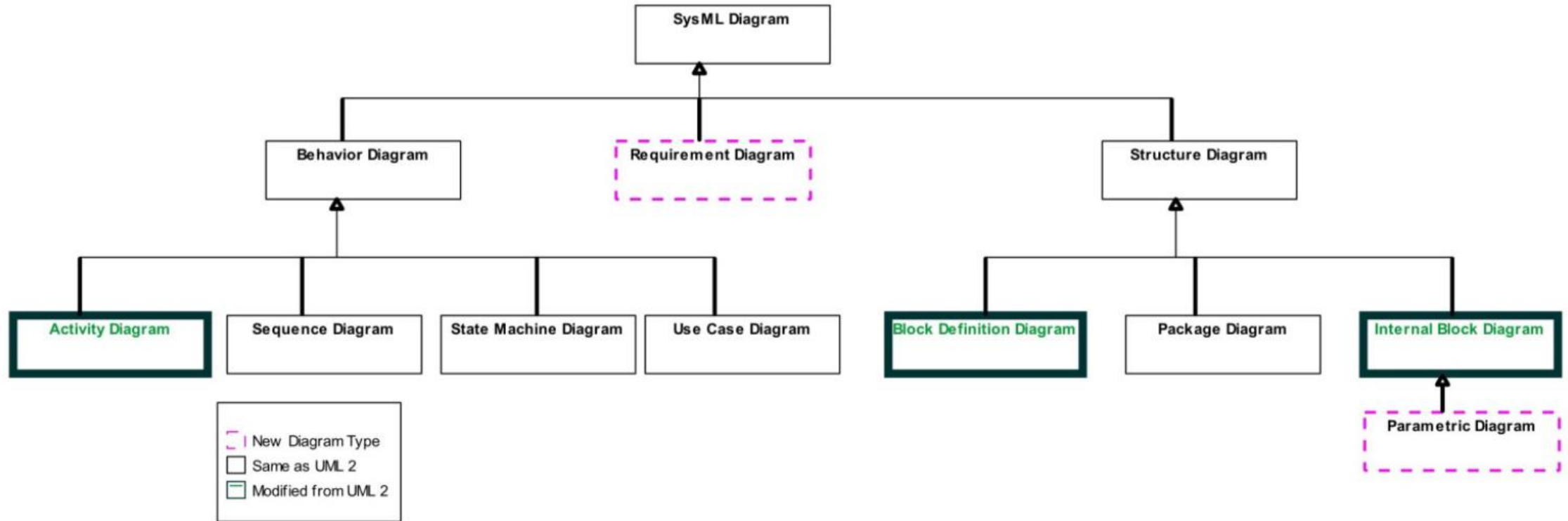


Figure A.1: SysML Diagram Taxonomy



# Klasyfikacja wymagań – cz. 1

DOCELOWO, WYMAGANIA POWINNY BYĆ OPISANE NA 3 POZIOMACH:

1. **(WB) WYMAGANIA BIZNESOWE / STRATEGICZNE** - SĄ TO WYMAGANIA OPISUJĄCE CELE BIZNESOWE LUB STRATEGICZNE, KTÓRE SYSTEM POMOŻE OSIĄGNĄĆ. NIE SĄ NA TYM ETAPIE OPISYWANE OCZEKIWANE FUNKCJONALNOŚCI ROZWIĄZANIA, A RACZEJ OCZEKIWANIA CO DO ROZWIĄZANIA OGÓŁEM – CO APLIKACJA MI DA, CO DZIĘKI NIEJ OTRZYMAM, JAK BĘDĘ MÓGŁ OSIĄGNĄĆ SWOJE CELE PRZY POMOCY ROZWIĄZANIA.
2. **(WU) WYMAGANIA UŻYTKOWNIKA** – BARDZIEJ SZCZEGÓŁOWY OPIS TEGO, JAK POTENCJALNI UŻYTKOWNICY WYOBRAŻAJĄ SOBIE KORZYSTANIE Z APLIKACJI – CO IM UŁATWI, CO IM PRZYSPIESZY, CZEGO DZIĘKI NIEJ NIE BĘDĄ MUSIELI ROBIĆ. NIE PISZEMY JESZCZE O KONKRETNÝCH FUNKCJONALNOŚCIACH (WYSZUKIWARKA, PROCES, WYDRUKI), A RACZEJ O KORZYŚCIACH I WARTOŚCI DOSTARCZONEJ UŻYTKOWNIKOM.
3. **WYMAGANIA SYSTEMOWE** - KONKRETNE WŁAŚCIWOŚCI, KTÓRE ROZWIĄZANIE POWINNO POSIADAĆ. W PODZIALE NA FUNKCJONALNOŚCI (WYMAGANIA SYSTEMOWE) ORAZ SPEŁNIANE OGRANICZENIA POZAFUNKCJONALNE (WYMAGANIA NIEFUNKCJONALNE).
  - a. **(WF) WYMAGANIA FUNKCJONALNE** - FUNKCJONALNOŚCI (FUNKCJE, MOŻLIWOŚCI) KTÓRE POWINNO POSIADAĆ ROZWIĄZANIE.
  - b. **(WNF) WYMAGANIA NIEFUNKCJONALNE** – OGRANICZENIA ZWIĄZANE Z WYDAJNOŚCIĄ, BEZPIECZEŃSTWEM, STOSEM ARCHITEKTONICZNYM, DOSTĘPNOŚCIĄ, POLITYKĄ BACKUPÓW, AWARIAMI, SLA ITP.

KAŻDE Z WYMAGAŃ POWINNO ZAWIERAĆ CO NAJMNIEJ:

1. Kod (WB/WU/WF/WNF\_001, \_002...),
2. Opis (WB i WU - ZGODNY Z FORMUŁĄ JAKO [KTO], CHCĘ MÓC [COŚ], ABY OSIĄGNĄĆ [COŚ]. WF i WNF - FORMUŁA TO „SYSTEM POWINIEN BYĆ W STANIE/UMOŻLIWIAĆ... [OPIS FUNKCJONALNOŚCI LUB OGRANICZENIA])
3. POWIĄZANI UŻYTKOWNICY – DLA WYMAGAŃ UŻYTKOWNIKA
4. (OPCJONALNIE) DODATKOWE NOTATKI I KOMENTARZE

DO WYBORU FORMA PRZEDSTAWIENIA WYMAGAŃ:

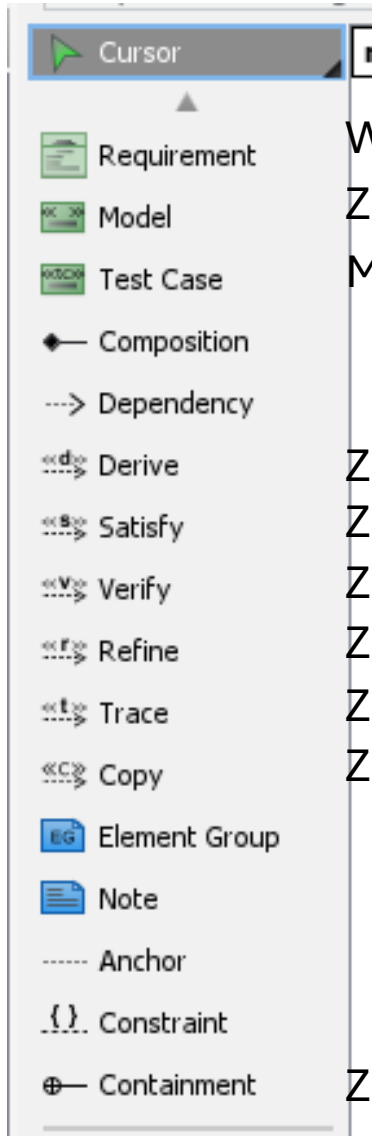
1. **TABELA**
2. **DIAGRAM WYMAGAŃ**

Rodzaj informacji	Definicja
Wymagania biznesowe	“Informacja, która opisuje dlaczego organizacja podjęła się projektu, określa cele biznesowe, definiuje wizję produktu i zawiera inne informacje dotyczące ustalania kierunku.” (Wiegers i Hokanson, 2024, p. 22)
Reguła biznesowa	„Dyrektywa, która definiuje lub ogranicza działania w ramach operacji organizacji. Polityka, regulacja, prawo lub standard, które prowadzą do wynikających z nich wymagań rozwiązania, które go egzekwują lub z nim są zgodne”
Ograniczenia	„Ograniczenie nałożone na wymagania, projekt lub implementację”
Wymagania danych	„Definicja obiektu danych lub elementu, którym system musi manipulować, jego budowy, atrybutów, relacji między obiektami danych oraz ich formatu wejściowego i wyjściowego”
Wymagania dla interfejsów zewnętrznych	„Opis połączenia między tworzonym rozwiązaniem a innymi elementami otaczającego go świata, w tym użytkownikami, innymi systemami oprogramowania, urządzeniami sprzętowymi i sieciovymi”
Wymagania funkcjonalne	„Opis pewnego zachowania, jakie produkt będzie wykazywał w określonych okolicznościach”
Wymagania нефunkcjonalne	„Najczęściej odnosi się do tego, co jest również znane jako „wymaganie dotyczące cech jakościowych. Cechy jakościowe opisują różne jakości, obsługę lub parametry wydajności rozwiązania”
Wymagania dotyczące rozwiązania	„Opis możliwości lub cechy, które produkt w trakcie tworzenia musi posiadać, aby zaspokoić określone wymagania użytkowników i przyczynić się do osiągnięcia celów biznesowych projektu. Wymagania dotyczące rozwiązania obejmują wymagania funkcjonalne, нефunkcjonalne, dotyczące danych oraz operacje manualne”
Wymagania systemowe	„Opis głównej możliwości lub cechy złożonego systemu, który składa się z wielu podsystemów, często obejmujących zarówno elementy sprzętowe, jak i oprogramowanie. Wymagania systemowe stanowią podstawę pochodnych wymagań dotyczących rozwiązania oprogramowania”
Wymagania użytkownika	„Opis zadania lub celu, który użytkownik chce osiągnąć za pomocą rozwiązania. Międzynarodowy Instytut Analizy Biznesowej (IIBA) uogólnia tę kategorię jako <<wymagania klientów>>, ale w rzeczywistości wszystkie wymagania pochodzą od pewnych klientów. Tutaj szczególnie odnosimy się do rzeczy, których użytkownik musi dokonać, i oczekiwań specyficznych dla użytkownika, które rozwiązanie musi spełniać”

# Klasyfikacja wymagań – cz. 2

- **Wymagania funkcjonalne**
- **Wymagania pozafunkcjonalne**
  - **Użyteczność** (usability) – spełnienie tych wymagań skutkuje zwiększeniem stopnia przyswajalności obsługi systemu przez użytkowników dzięki estetycznemu i ergonomicznemu interfejsowi użytkownika, zapewniającemu intuicyjną nawigację w systemie;
  - **Niezawodność** (reliability) – czyli własność systemu, określająca, czy pracuje on poprawnie; jej miernikami są między innymi: średni czas międzyawaryjnymi, średni czas wdrożenia obejścia (bypass), średni czas naprawy, liczba błędów na 1k linii kodu;
  - **Wydajność** (performance) – wolumen pracy wykonanej przez system w określonym czasie i przy zaangażowaniu określonych zasobów; miernikami wydajności mogą być między innymi: czas odpowiedzi systemu, liczba transakcji w jednostce czasu, liczba jednocześnie obsługiwanych klientów zdalnych;
  - **Przystosowalność** (supportability) – czyli łatwość konfigurowania, aktualizowania, serwisowania systemu, rejestrowania zdarzeń systemowych w logach i przystosowania oprogramowania do specyficznych potrzeb użytkownika przez help desk i personel wsparcia technicznego.

# Elementy diagramu wymagań



Wymaganie, które będziemy określać

Zachowuje się jak referencja do określonego obiektu w projekcie

Metoda weryfikacji wymagania przy spełnieniu określonego X

Zależność wyprowadzania

Zależność realizacji

Zależność weryfikowania

Zależność precyzowania

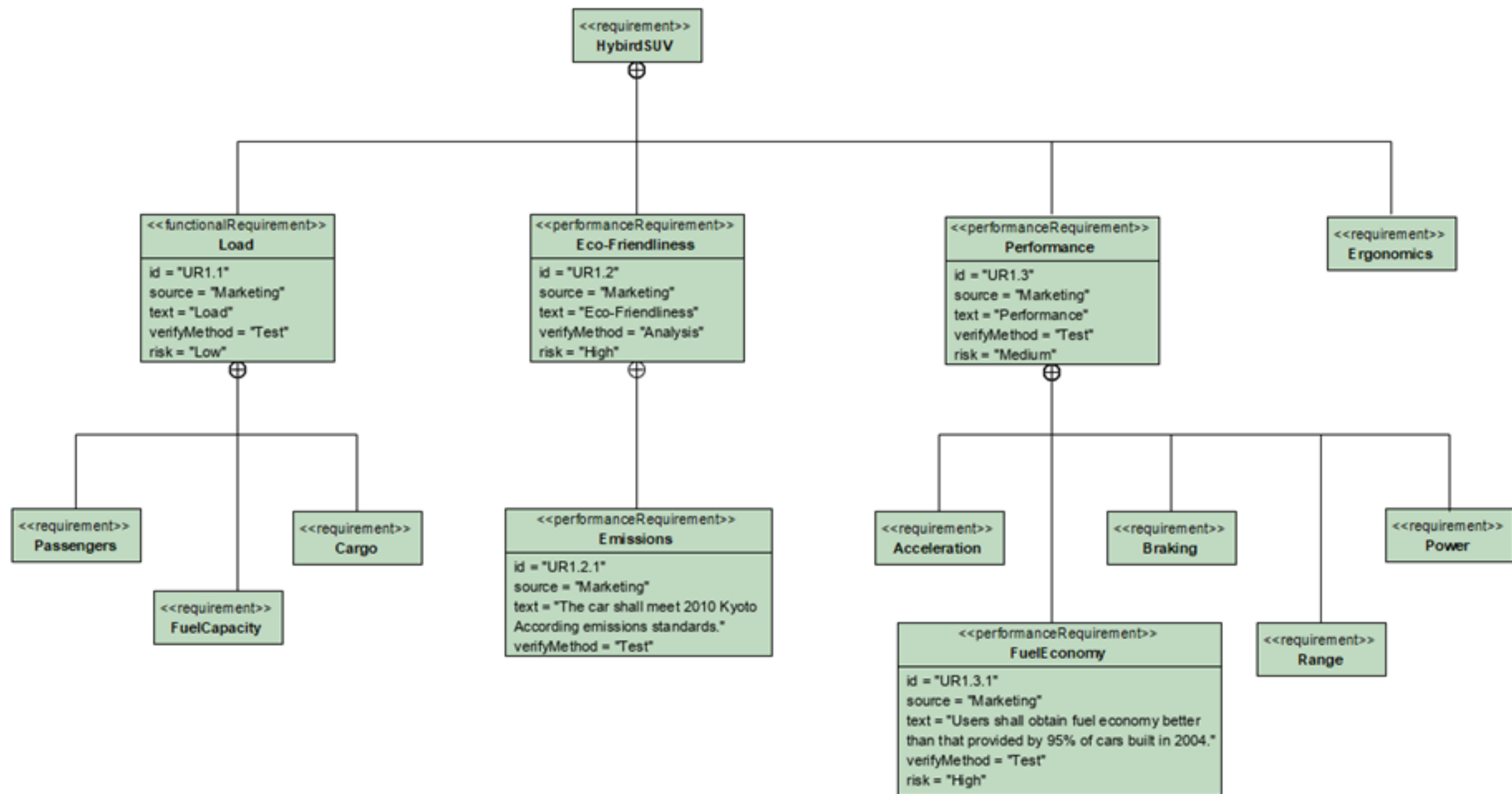
Zależność śledzenia

Zależność powielania

Zagnieżdżenie

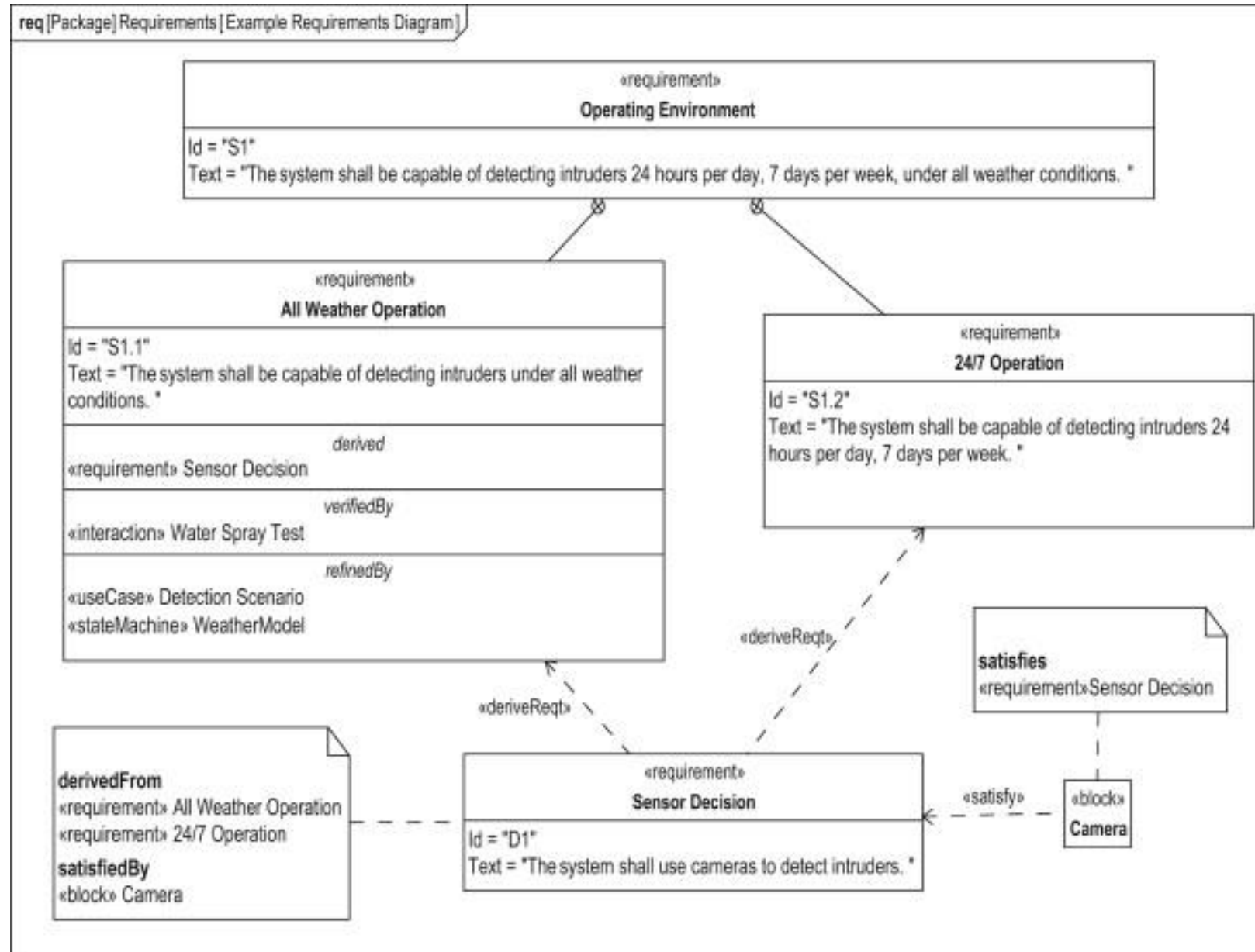
# Zagnieżdżenie

- Najpowszechniej stosowany
- Umożliwia ono połączenie wymagań nadrzędnych z podrzędnymi, przez co tworzona jest wielopoziomowa, hierarchiczna struktura wymagań systemowych.
- Wymaganie na danym poziomie hierarchii może mieć tylko jeden element nadrzędny. Nie dotyczy wymagania będącego na szczycie hierarchii, które naturalnie nie posiada wymagań nadrzędnych.
- Wymaganie nadrzędne zostaje spełnione, gdy wszystkie podrzędne.
- Wielokrotne użycie danego wymagania pomiędzy różnymi gałęziami hierarchii wymagań jest możliwe dzięki zależności powielania <<copy>>



# Zależność wyprowadzania

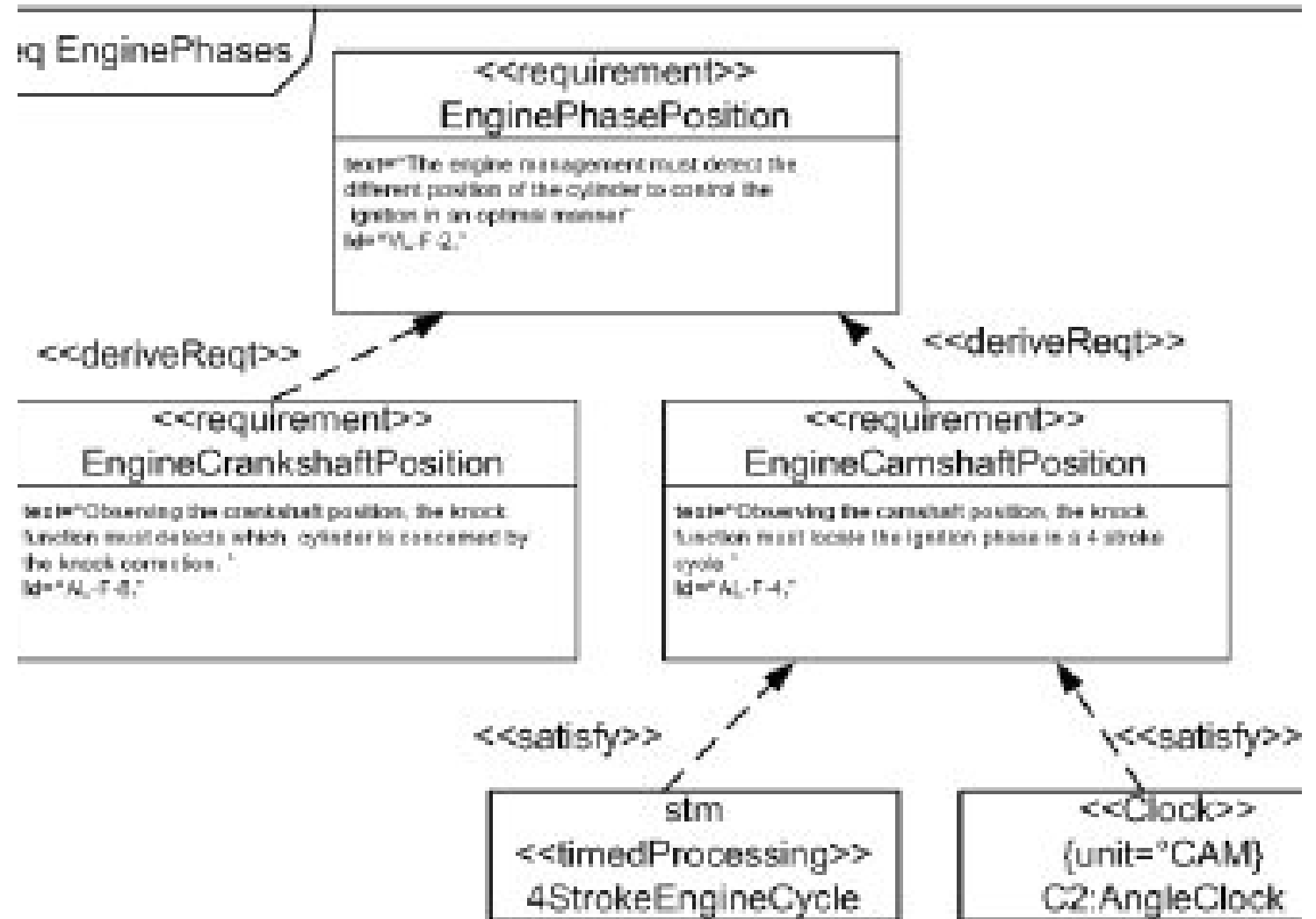
- Wykorzystuje stereotyp <<derive>>
- Pozwala na wyprowadzenie wymagania docelowego z wymagania źródłowego.
- Zazwyczaj pojedyncze wymaganie źródłowe wspierane jest przez szereg wymagań docelowych, powiązanych osobnymi zależnościami wyprowadzania.

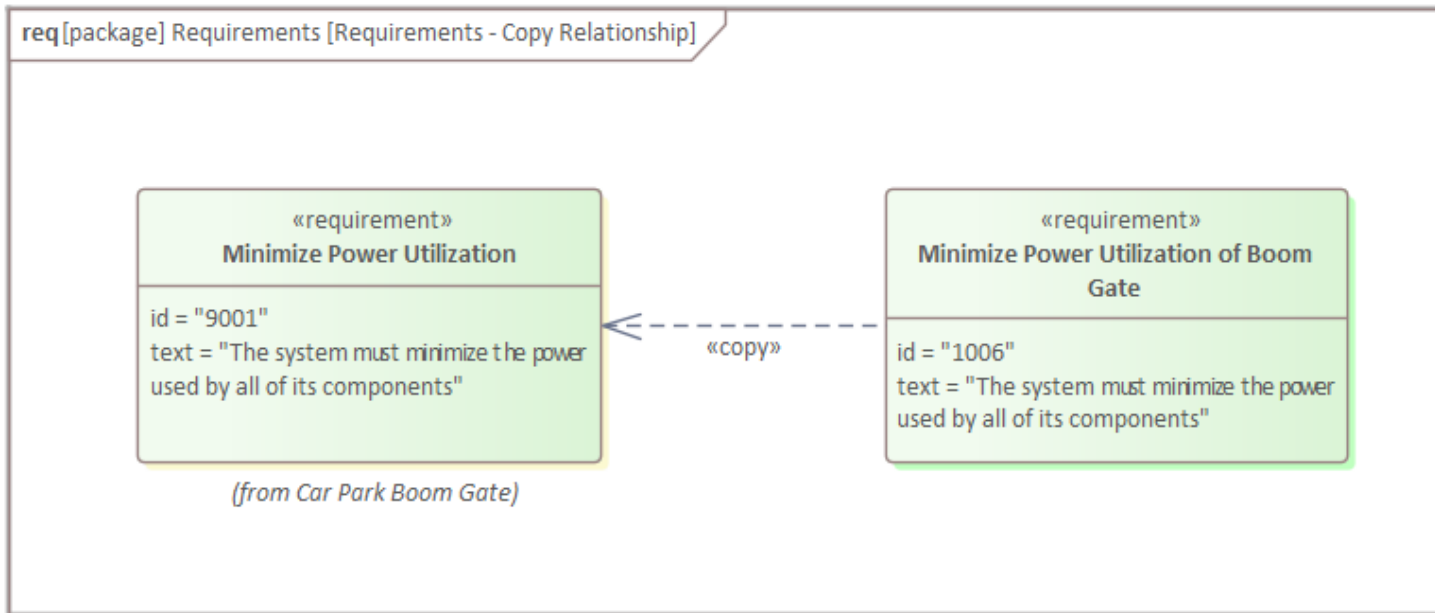




# Zależność realizacji

- Wykorzystuje stereotyp `<<satisfy>>`
- Pozwala określać skutki zmian w obrębie wymagania wobec elementów od niego zależnych i zarazem wskazywać, które z kluczowych elementów projektu i implementacji systemu są podatne na zmiany w obrębie danego wymagania i jakie ma to implikacje dla tego wymagania.



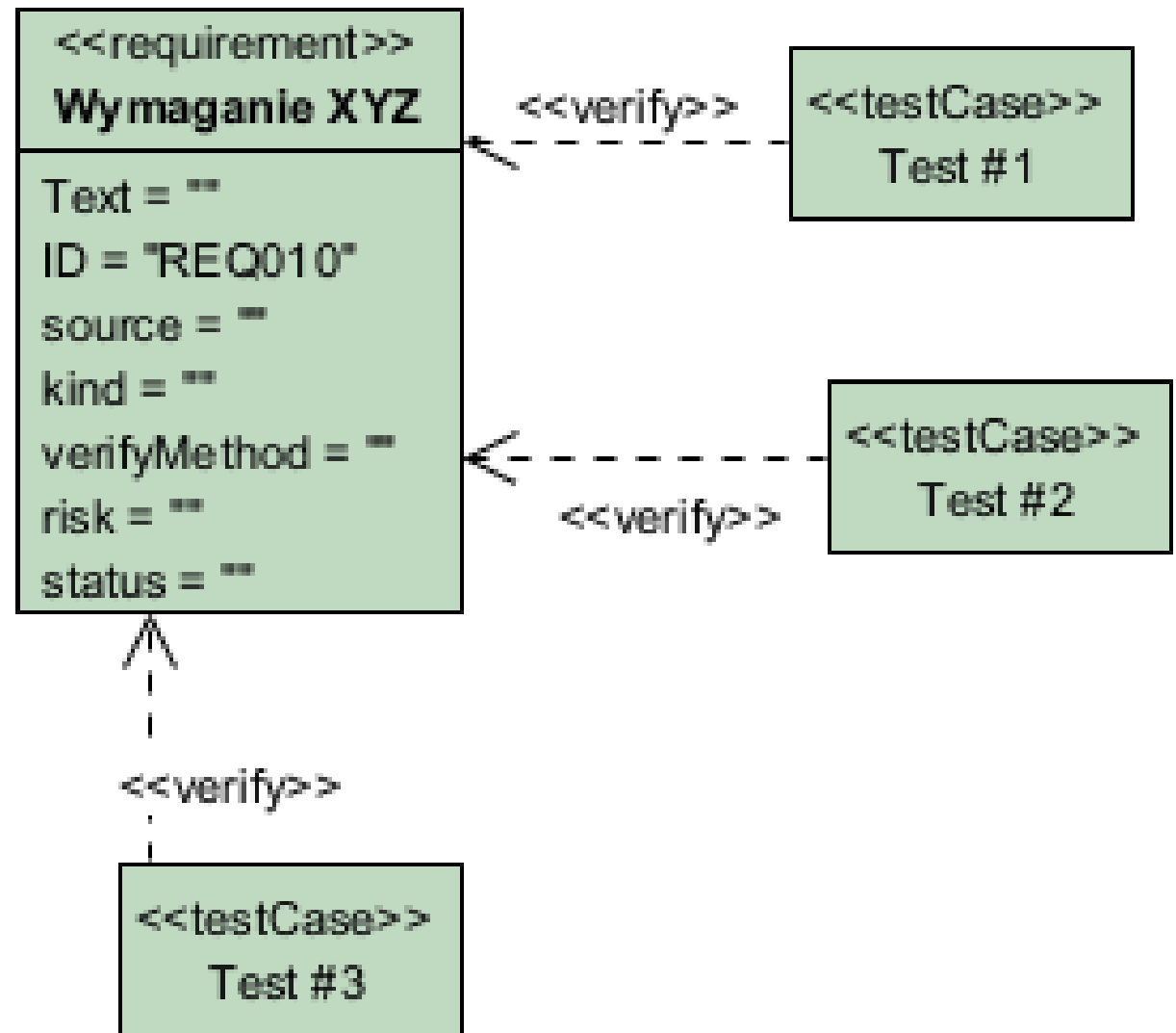


## Zależność powielania

- Używa stereotypu <<copy>>
- Jest sprzeczna z uniwersalną zasadą ponownego wykorzystania (reuse), konsekwentnie stosowaną w systemach obiektowych. Powoduje to niepotrzebny chaos w projekcie, jak również utrudnia śledzenie zmian.
- W momencie poprowadzenia zależności pomiędzy dwoma wymaganiami przyjmuje się, że wymaganie docelowe ma charakter tylko do odczytu, tj. przyjmuje ono tożsamą treść w stosunku do wymagania źródłowego.
- Zastosowania tego związku eliminuje zatem konieczność wyszukania wszystkich wymagań o analogicznej treści i stosownego wprowadzania korekt. Z kolei numery porządkowe oraz nazwy wymagań źródłowych i docelowych mogą się różnić.

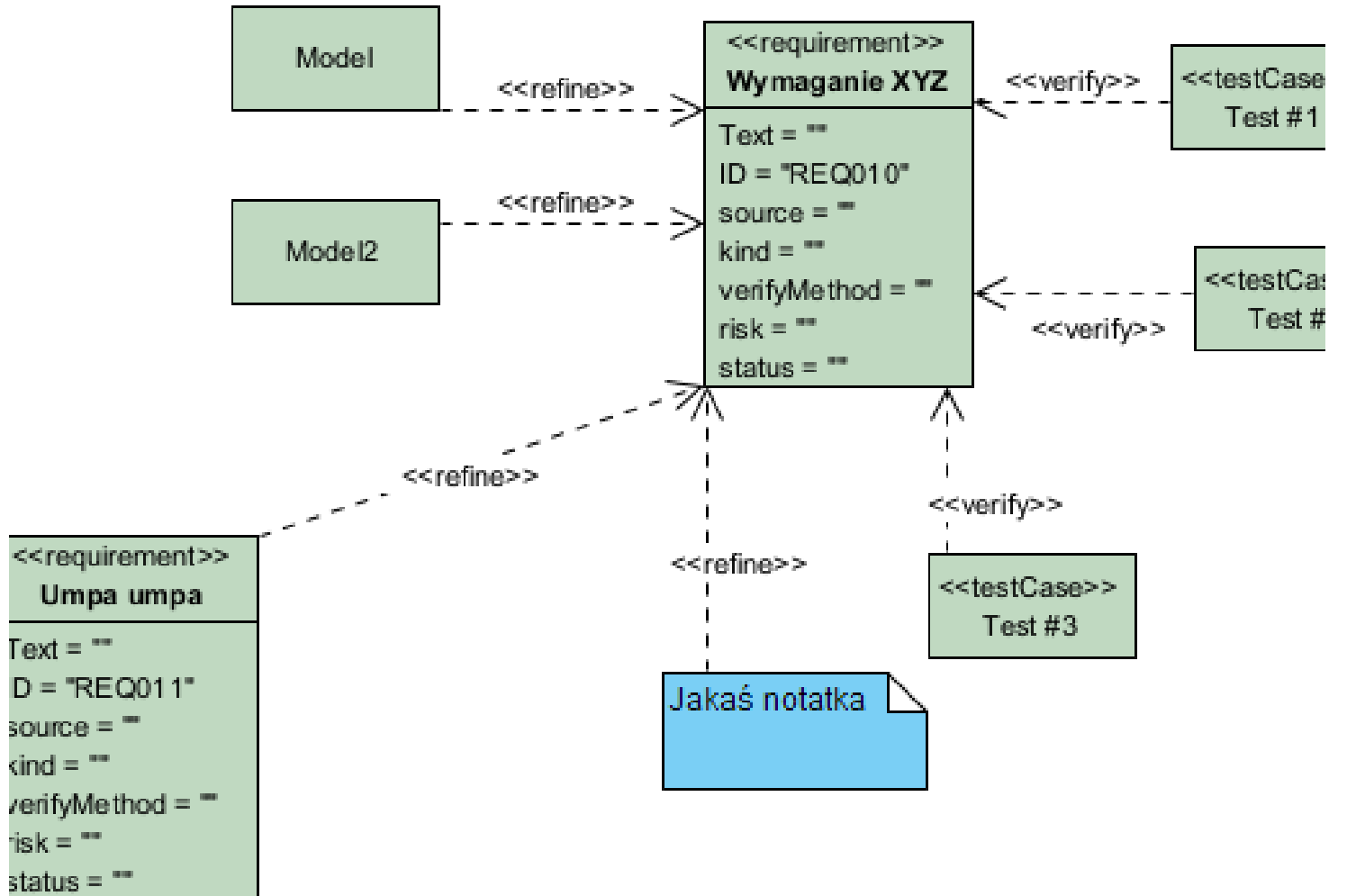
# Zależność weryfikowania

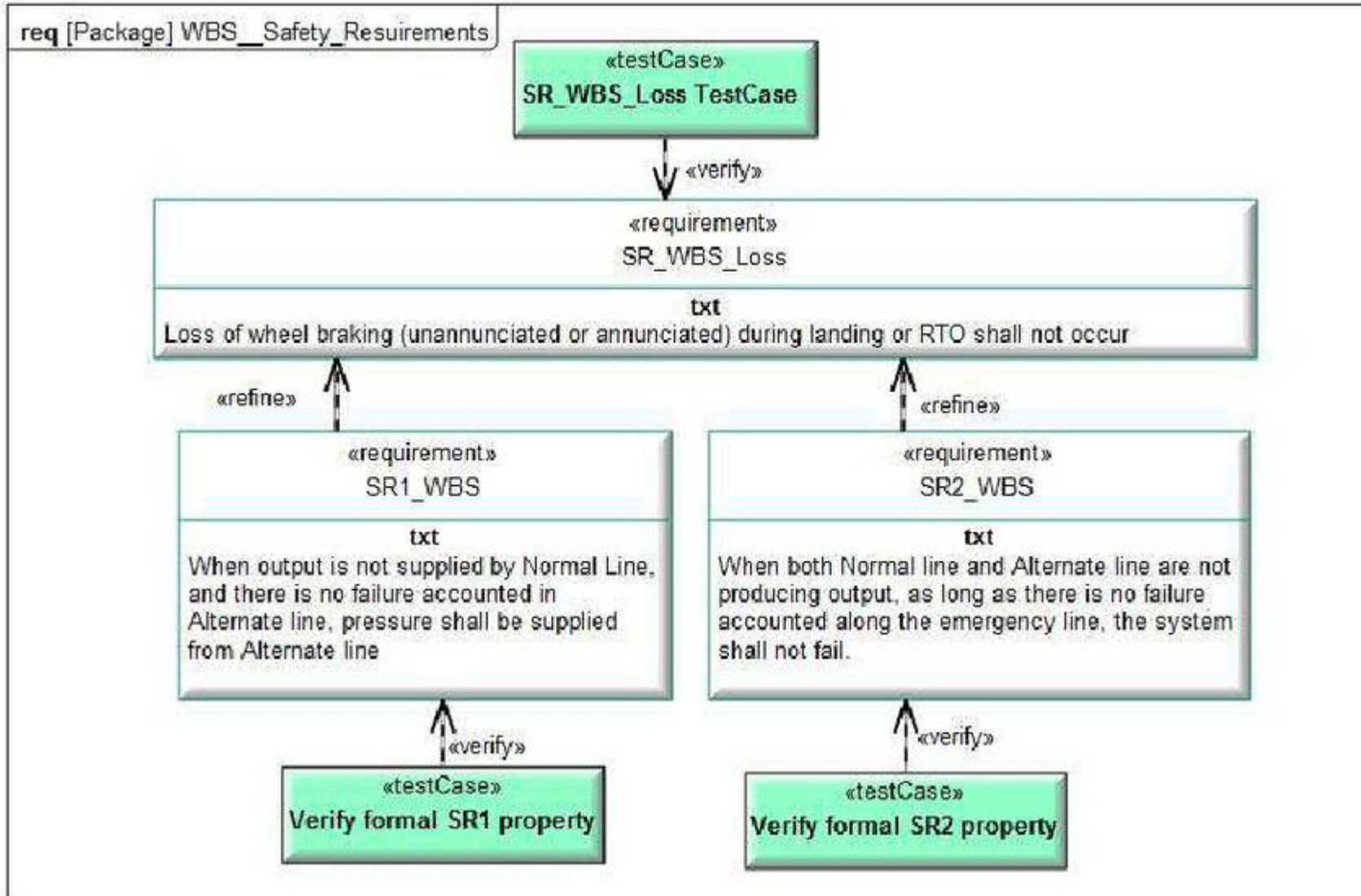
- Używa stereotypu <<verify>>
- Zaliczana jest do dobrych praktyk – pozwala poprzez testy sprawdzić czy poszczególne wymagania zostały zrealizowane, poprzez pisanie testów.
- Zależność weryfikowania wyprowadzona jest od elementu docelowego, którym jest tzw. testowy przypadek użycia.
- Wymaganie może mieć przypisanych kilka testów
- Testowemu przypadkowi użycia można także przypisać dodatkowe stereotypy testowania, odpowiadające różnym metodom testowania, np.:
  - Analyse
  - Inspect
  - Demonstrate
  - Test
- Ma szczególne znaczenie w iteracyjno-przyrostowym cyklu życia systemu

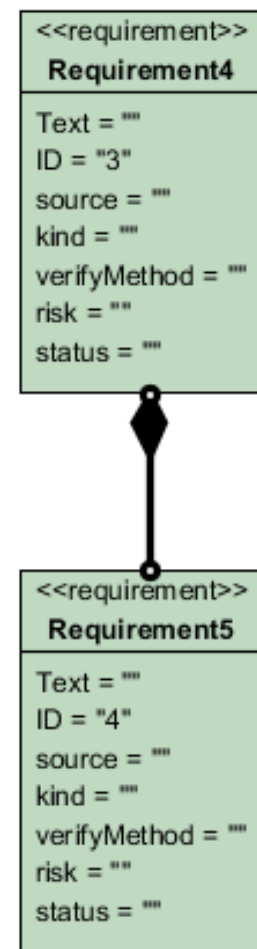
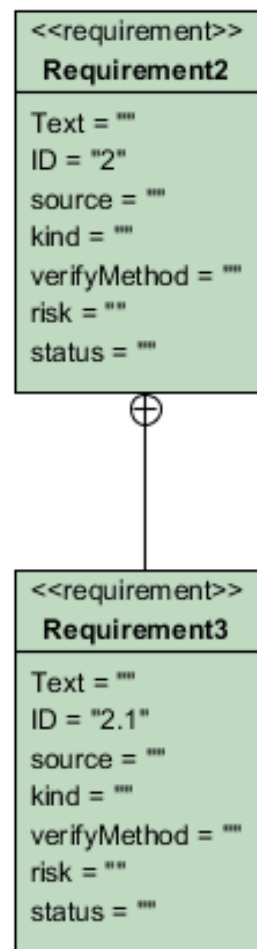
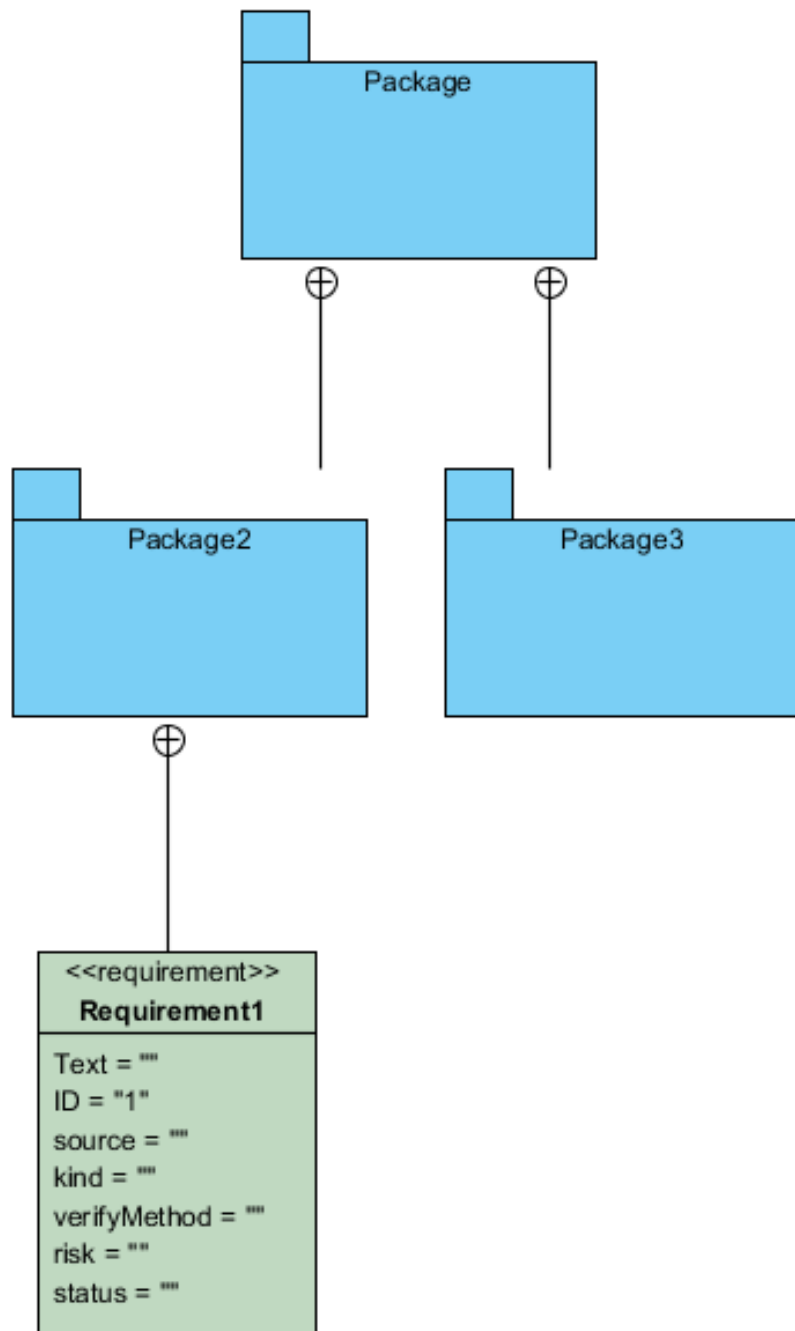


# Zależność precyzowania

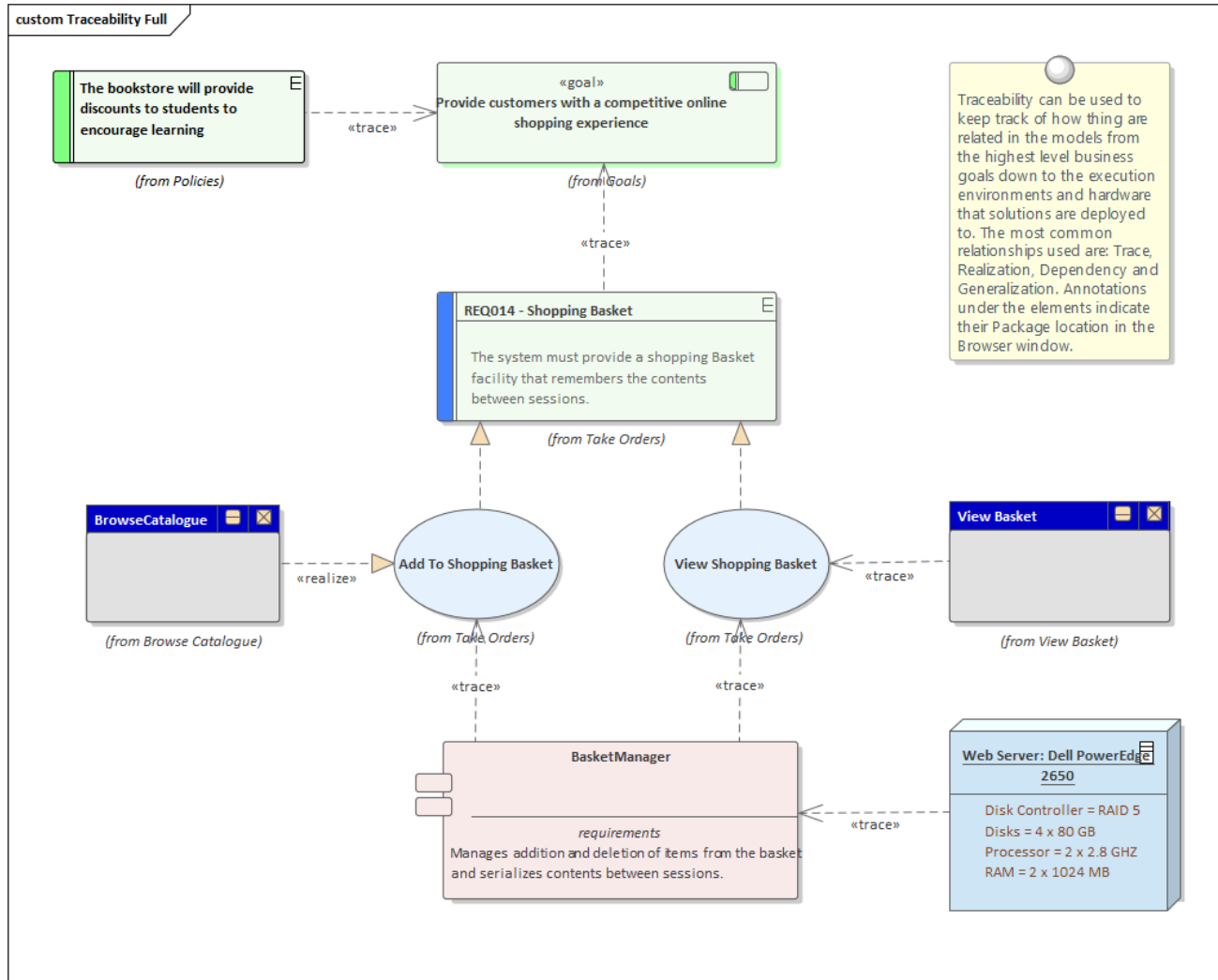
- Używa stereotypu <<refine>>
- Pozwala na wprowadzenie do diagramu wymagań systemowych licznych detali, reprezentowanych poprzez docelowe elementy związku.
- Ułatwiają zrozumienie sensu wymagania







# Zależność śledzenia



- Używa stereotypu <<trace>>
- Umożliwia na zaprezentowanie nieformalnego związku pomiędzy wymaganiem a dowolnym elementem modelu systemu, w tym innym wymaganiem
- Specyfikacja SysML pozwala na swobodę stosowania tego typu związków



# Mapowanie Wymagań na przypadki użycia

- Każde wymaganie powinno być realizowane przynajmniej przez jeden przypadek użycia.
- Jeden przypadek użycia może realizować więcej niż 1 wymaganie.
- Jeżeli zostanie(-ą) puste komórki, oznacza że proces wydobywania wymagań był błędny lub mapowanie nie zostało przeprowadzone poprawnie
- Tabela mapowania

	Wymaganie 1	Wymaganie 2	Wymaganie 3	Wymaganie 4	Wymaganie X
Przypadek użycia 1	R				
Przypadek użycia 2		R		R	
Przypadek użycia 3					R
Przypadek użycia X			R		