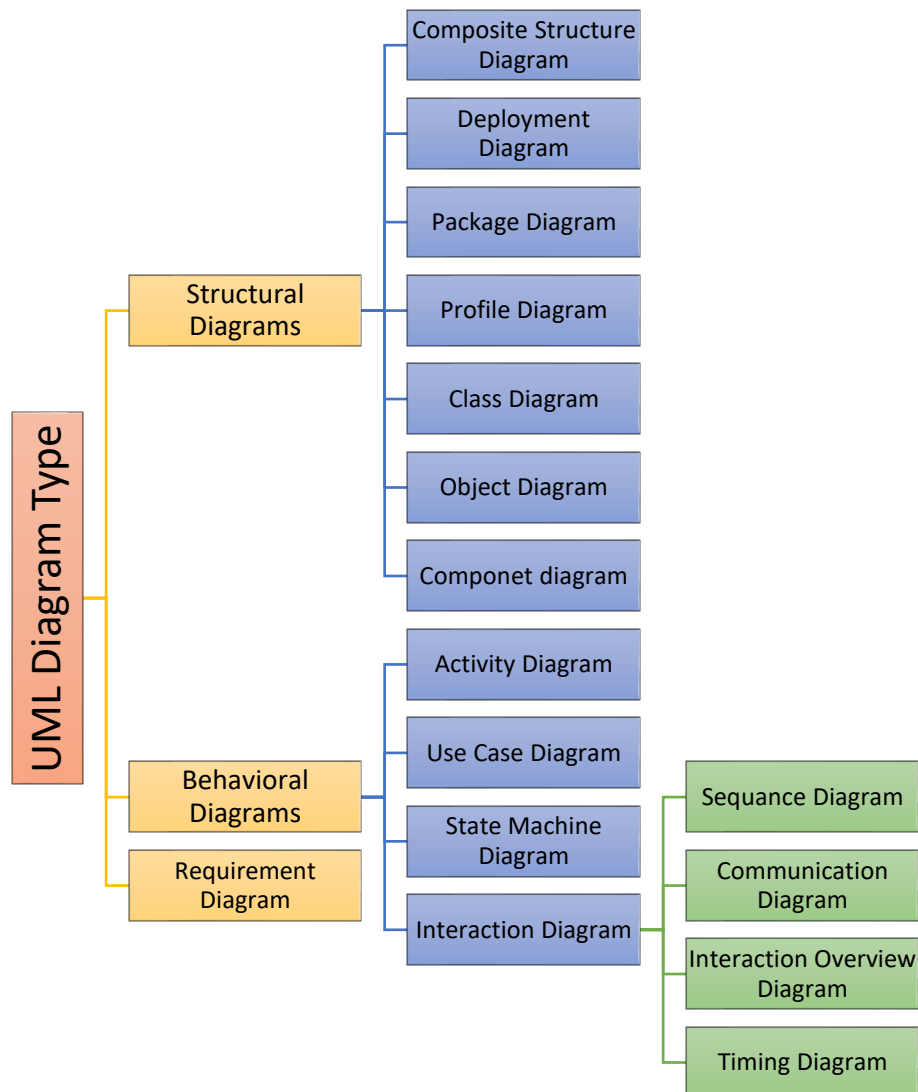


Laboratorium 2a

Diagram wymagań

Unified Modeling Language (UML) to uniwersalny, rozwojowy język modelowania w dziedzinie inżynierii oprogramowania, którego celem jest zapewnienie standardowego sposobu wizualizacji projektu systemu. Reprezentuje on zbiór najlepszych praktyk inżynieryjnych, które okazały się skuteczne w modelowaniu dużych i złożonych systemów. UML wykorzystuje głównie notacje graficzne do wyrażania projektów oprogramowania. Korzystanie z UML pomaga zespołom projektowym komunikować się, badać potencjalne projekty i weryfikować projekt architektoniczny oprogramowania. Jak wspomniano wcześniej, UML jest bardzo ważną częścią rozwoju oprogramowania obiektowego i procesu tworzenia oprogramowania, ale sam w sobie nie jest metodą rozwoju; został jednak zaprojektowany tak, aby był kompatybilny z wiodącymi metodami tworzenia oprogramowania obiektowego swoich czasów. Diagramy UML można podzielić na kategorie hierarchiczne zgodnie z podziałem przedstawionym na Rysunku 1.

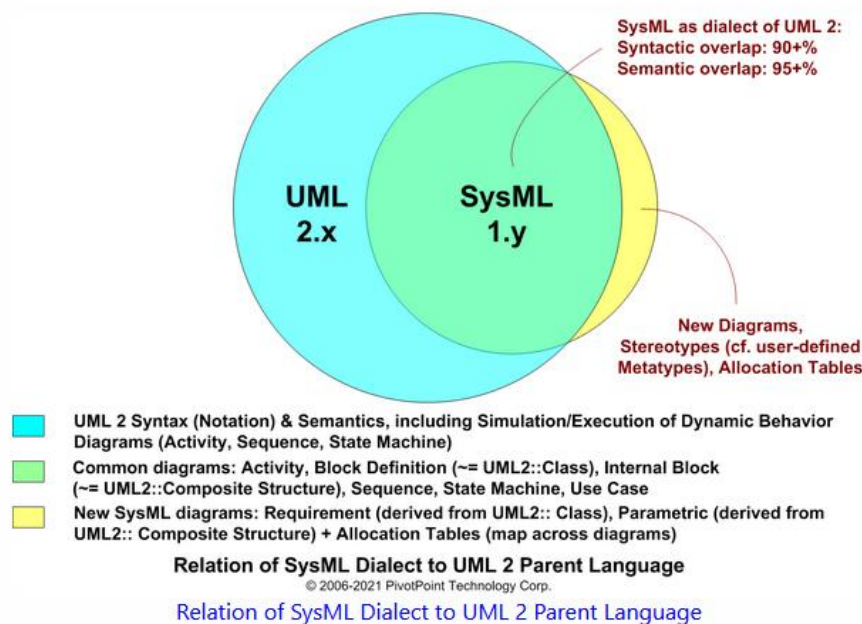


Rysunek. 1. Typy diagramów UML

Requirement Diagram

Wymaganie określa zdolność lub warunek, który musi (lub powinien) być spełniony. Wymaganie może określać funkcję, którą system musi wykonać lub warunek wydajności, który system musi osiągnąć. Przypadki użycia są skuteczne do przechwytywania wymagań funkcjonalnych, ale nie nadają się do wyrażania wymagań niefunkcjonalnych. Włączenie wymagań tekstowych do SysML (Systems modeling language) skutecznie uwzględnia szeroki zakres wymagań.

Diagram wymagań jest nowym typem diagramu specjalnie używanym w SysML, w którym wymagania i relacje między nimi mają być określone. Standardowe wymaganie SysML zawiera właściwości określające jego unikalny identyfikator i samo wymaganie tekstowe, dodatkowe właściwości, takie jak status weryfikacji, priorytet itp. mogą być również określone przez użytkownika.



Rysunek. 1. Relacje pomiędzy SysML a UML

Projektowanie diagramu wymagań:

Wymaganie: Wymaganie (notacja: prostokąt ze słowem kluczowym "requirement") to zdolność lub warunek, który system musi ("shall") spełnić. Wymaganie funkcjonalne (słowo kluczowe "functionalRequirement") określa funkcję, którą system musi wykonać, podczas gdy wymaganie niefunkcjonalne (NFR) określa kryteria jakości, które można wykorzystać do przetestowania skuteczności funkcji systemu.

Diagram wymagań (req): jest statycznym diagramem strukturalnym, który pokazuje relacje między konstrukcjami wymagań ("requirement"), elementami modelu, które je spełniają ("satisfy" Dependency) oraz przypadkami testowymi, które je weryfikują ("verify" Dependency).

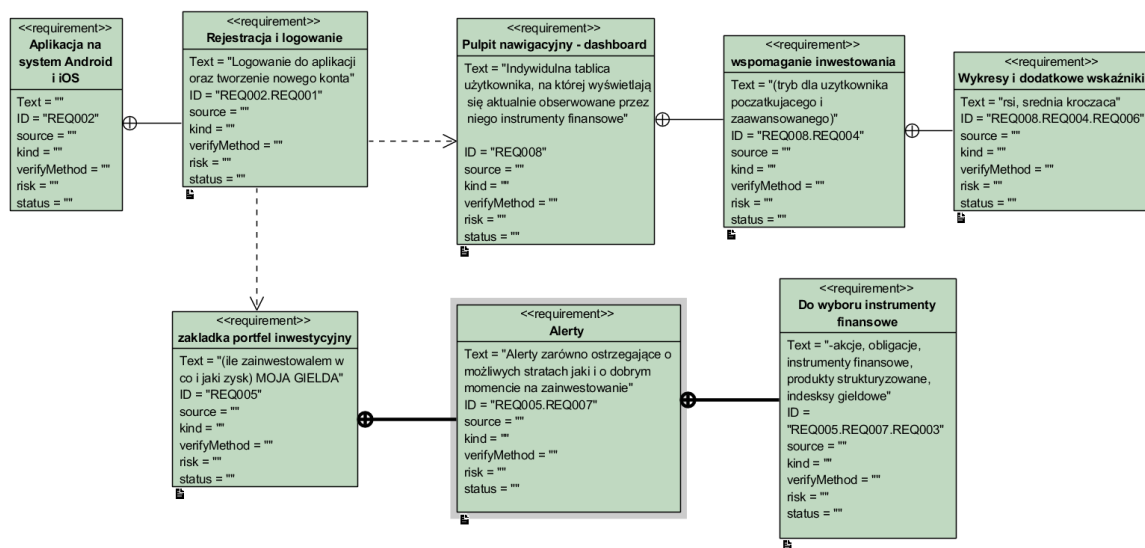
Celem diagramów wymagań jest określenie zarówno wymagań funkcjonalnych, jak i niefunkcjonalnych w modelu, tak aby można je było prześledzić do innych elementów modelu, które je spełniają i przypadków testowych, które je weryfikują.

Inżynieria Oprogramowania – Laboratorium

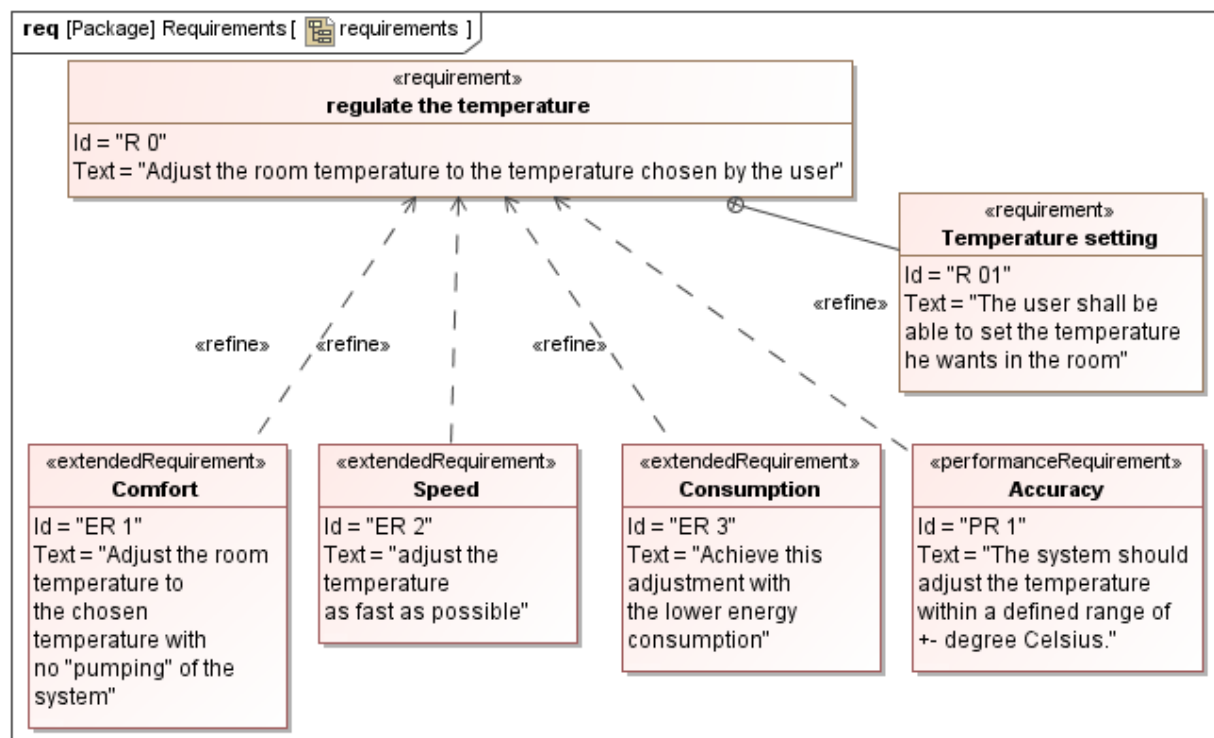
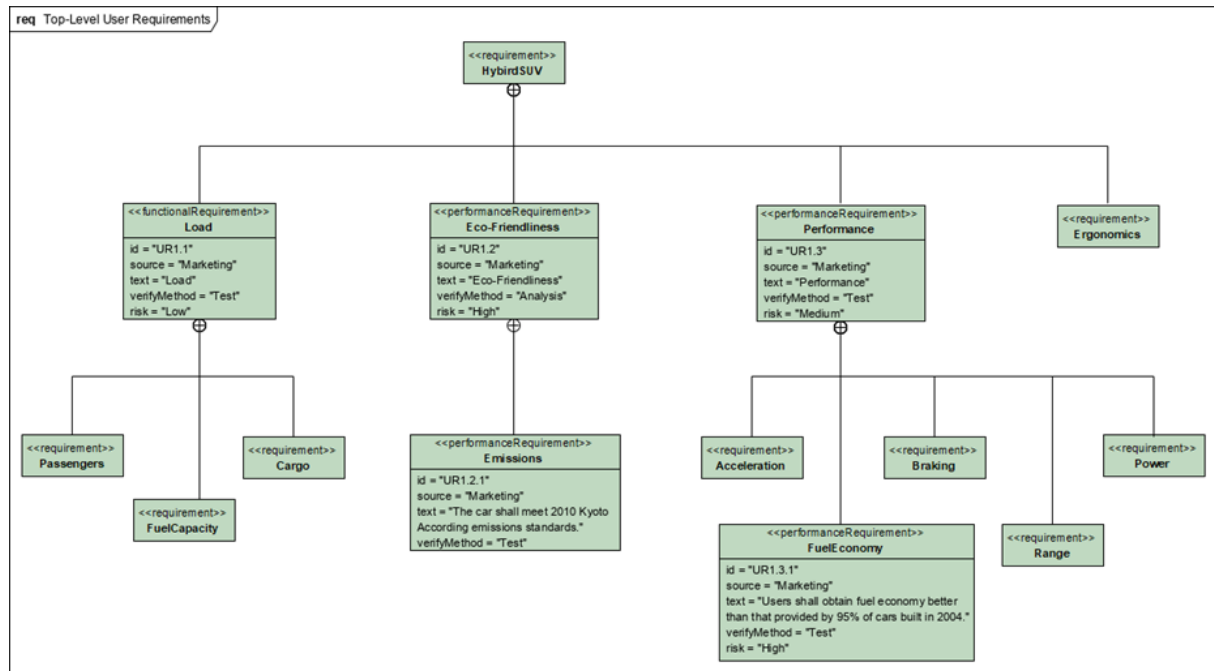
Przykłady diagramu wymagań przedstawiono na Rysunku 2.

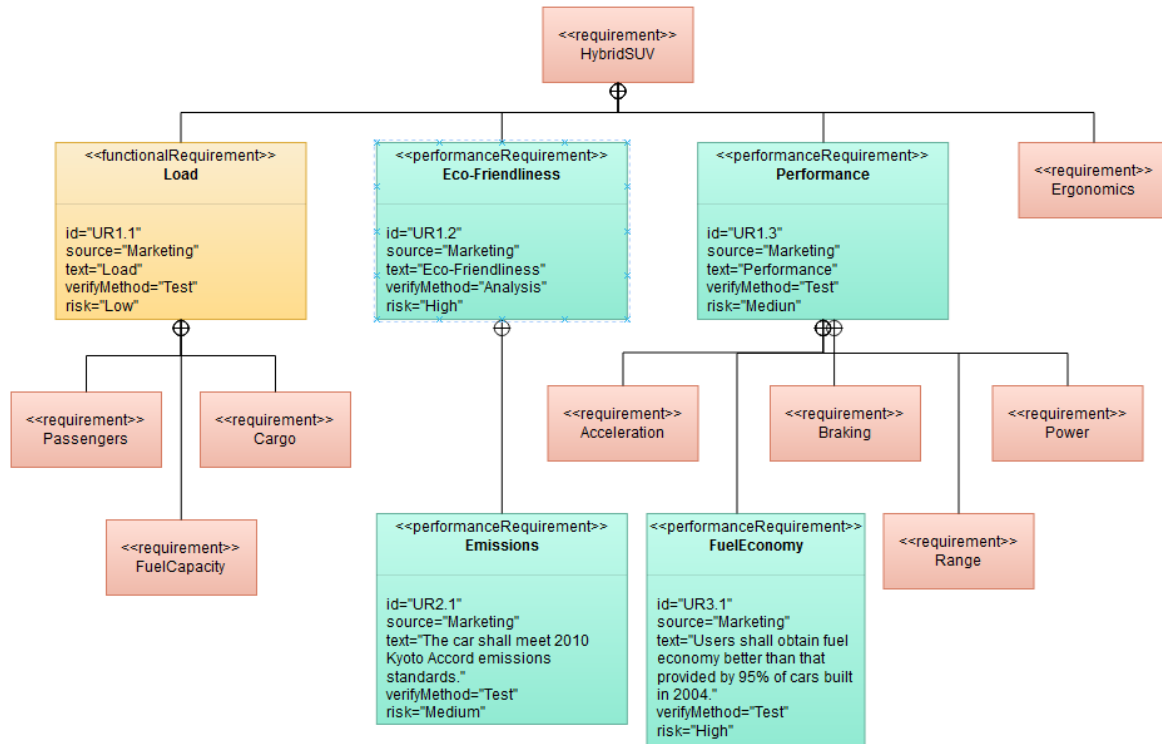
ID	Name	Priority	Description
1.	Secure Login	10	The system must have a secure login mechanism to ensure that only authorized users can access the payroll data.
1.1.	Multi-factor Authentication	10	The system must have multi-factor authentication to ensure that login is secure and user identity is verified
1.2.	User Role Management	10	The system must have a way to assign roles to users and define what actions they can perform in the system.
2.	Employee Management	10	The system must have a way to store and manage employee records, including employee personal information, such as name, address, contact information, etc.
2.1.	Employee Personal Information	10	The system must have a way to store and manage employee personal information, such as name, address, contact information, etc.
2.2.	Employee Salary and Benefits	10	The system must have a way to store and manage employee salary and benefits information, such as hourly rate, vacation days, etc.
3.	Payroll Processing	10	The system must have the ability to process payroll, including calculating taxes, deductions, and net pay.

req Diagram wymagań SIAG



Inżynieria Oprogramowania – Laboratorium





Rysunek. 2. Przykładowe diagramy wymagań

Diagramy UML można projektować w środowisku Software Ideas Modeler (darmowe środowisko w wersji edukacyjnej) Szczegółowe informacje odnośnie środowiska dostępne pod adresem: <https://www.softwareideas.net/>

Zadanie 1. Na podstawie opisu świata rzeczywistego należy zaprojektować diagram wymagań.

Literatura:

1. Software Ideas Modeler <https://www.softwareideas.net> (data dostępu 29.10.2023)
2. Diagram wymagań <https://www.softwareideas.net/requirement-diagram> (data dostępu 29.10.2023)
3. Sprawdzone wskazówki dotyczące definiowania wymagań dla projektów oprogramowania <https://www.softwareideas.net/a/1834/7-proven-tips-to-define-requirements-for-software-projects> (data dostępu 29.10.2023)
4. Tworzenie diagramu wymagań SysML przy użyciu Open API <https://knowhow.visual-paradigm.com/openapi/requirement-diagram/> (data dostępu 29.10.2023)