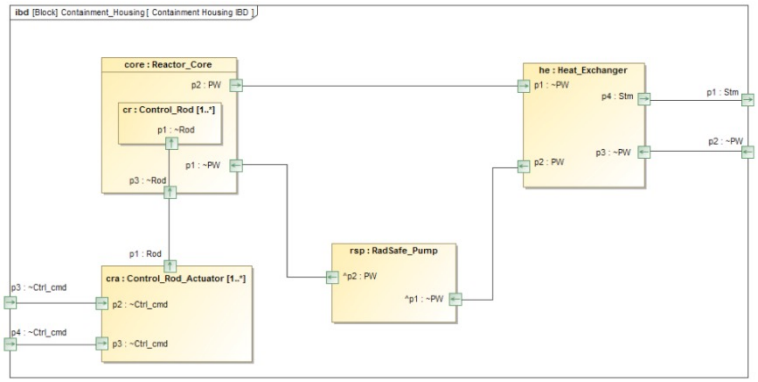
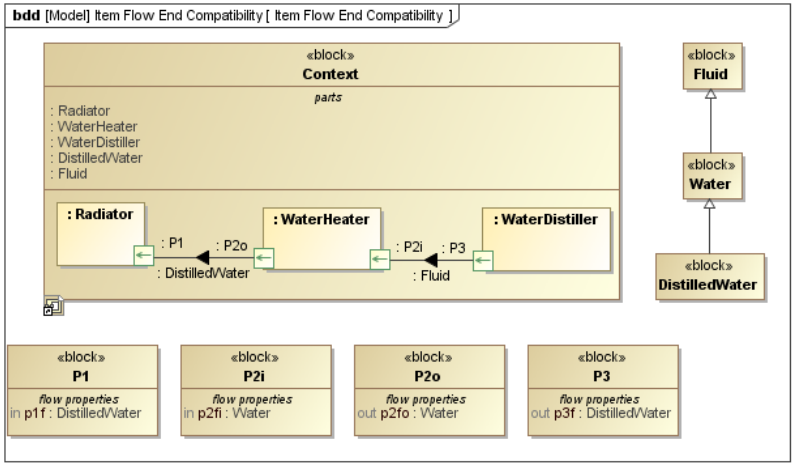


Modelando estrutura com Blocos em SysML



Curso Ford

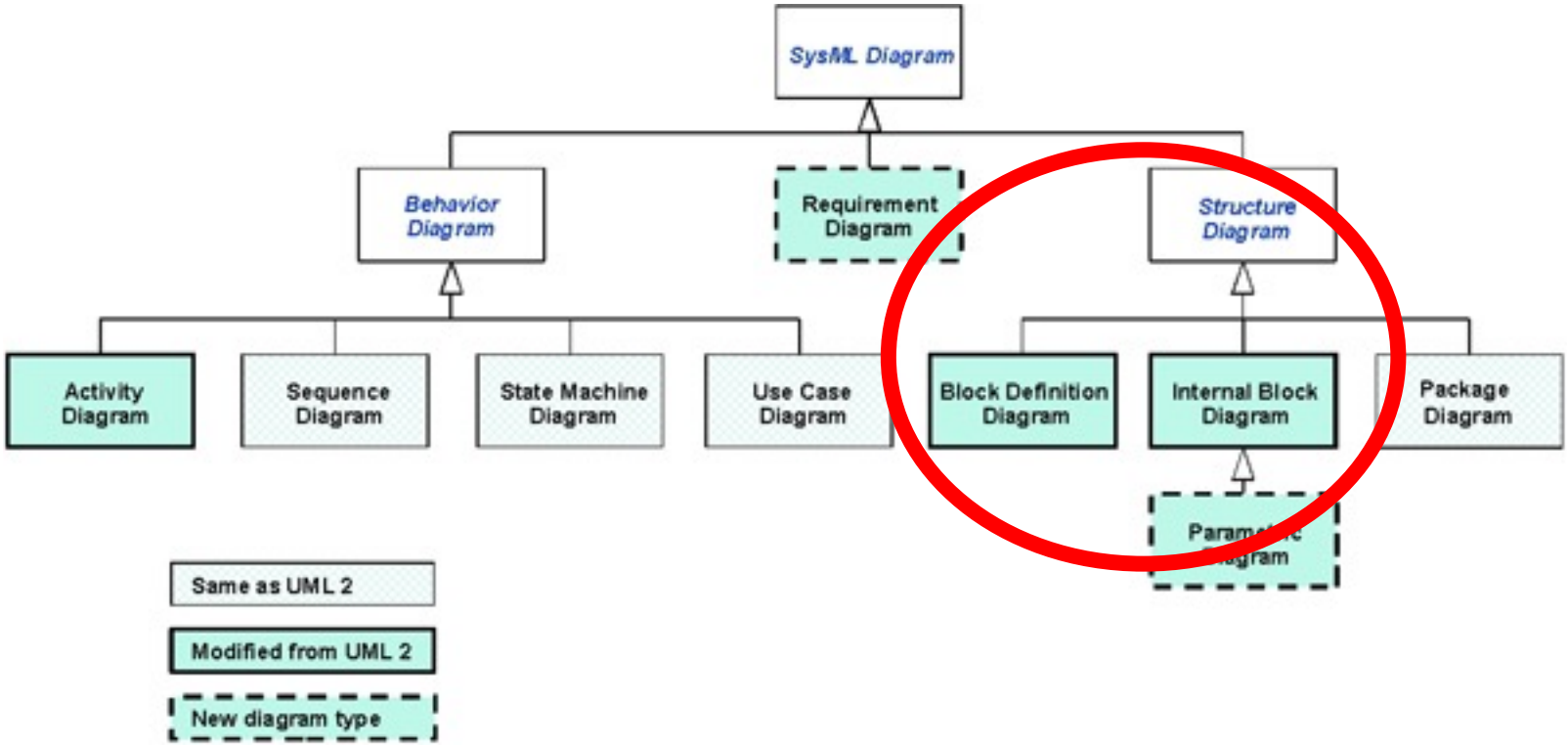
Prof. Lucas Albertins

Agenda

- Blocos
- Diagramas de Blocos
- Relacionamentos entre Blocos
- Variantes e Configurações
- Modelando Fluxos
- Comportamento de Blocos
- Utilizando Portas
- Conclusões

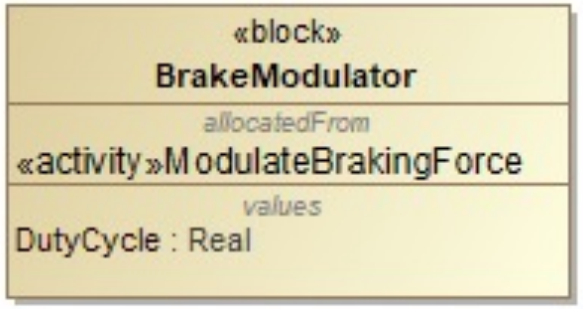
SysML

- **UML Profile for System Engineering**

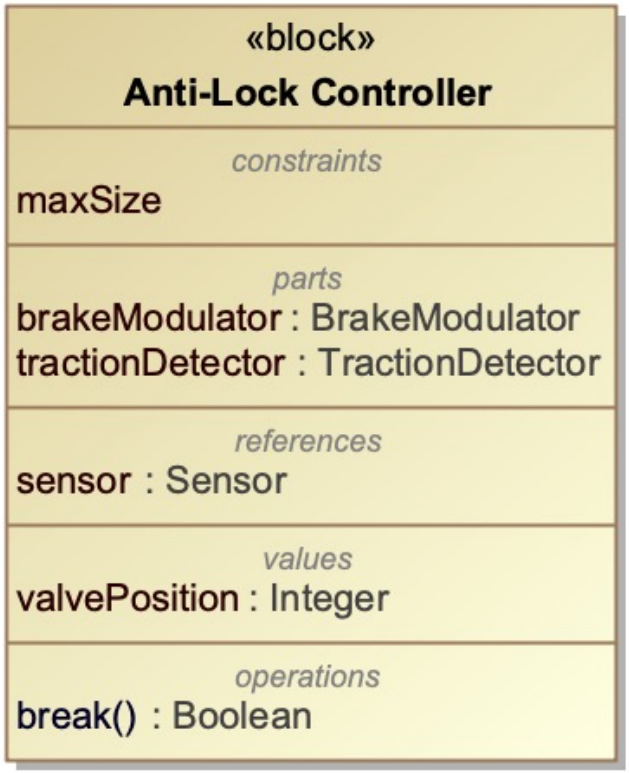


Blocos

- Elementos estruturais básicos
- Fornecem um conceito universal para descrever a estrutura de um elemento ou sistema (lógico, conceitual ou físico)
  - Sistema
  - Hardware
  - Software
  - Dados
  - Procedimentos
  - Pessoa



Blocos



- Define um tipo que descreve um conjunto de instâncias (objetos)
- Três aspectos:
  - Estrutural
  - Comportamental
  - Restrições

Blocos

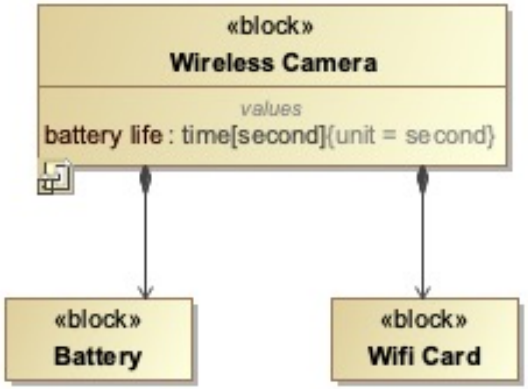
- Múltiplos compartimentos podem
- descrever as características do bloco
  - Properties (parts, references, values, ports)
  - Operations
  - Constraints
  - Allocations from/to other model elements (e.g. atividades)
  - Requisitos que o bloco satisfaz
  - Compartimentos definidos pelo usuário

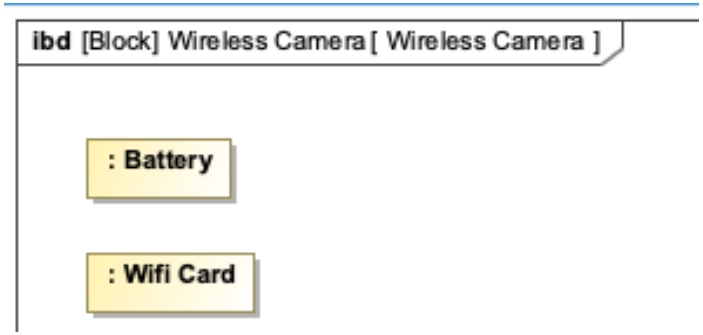
Propriedades de um bloco

- Elementos estruturais primários de um bloco
- Tem um tipo que define sua característica
- Possui três categorias
  - **Part Property**
    - Define a composição do bloco em termos dos seus elementos constituintes
  - **Reference Property**
    - Trata das partes referenciadas que são possuídas por outros blocos
  - **Value Property**
    - Descreve características quantificáveis como velocidade, peso, etc.

Parte (Part Property)

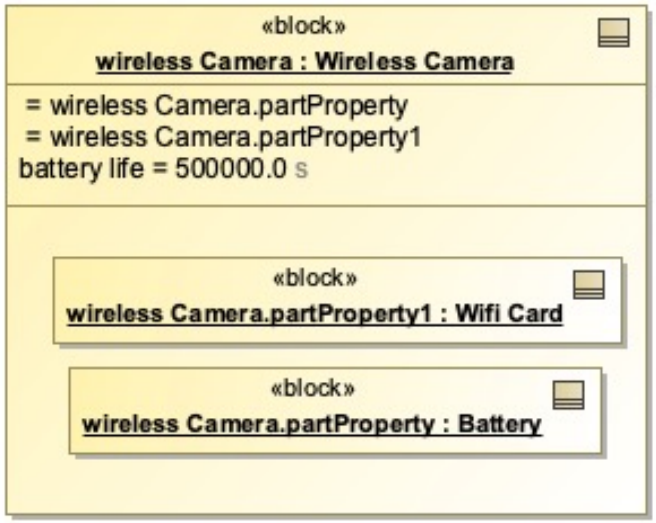
- Uso de um bloco no contexto de composição
- Ilustra o **uso** de um bloco dentro de outro
- IBD para ilustrar conexões entre diferentes partes
- Não é uma instância!



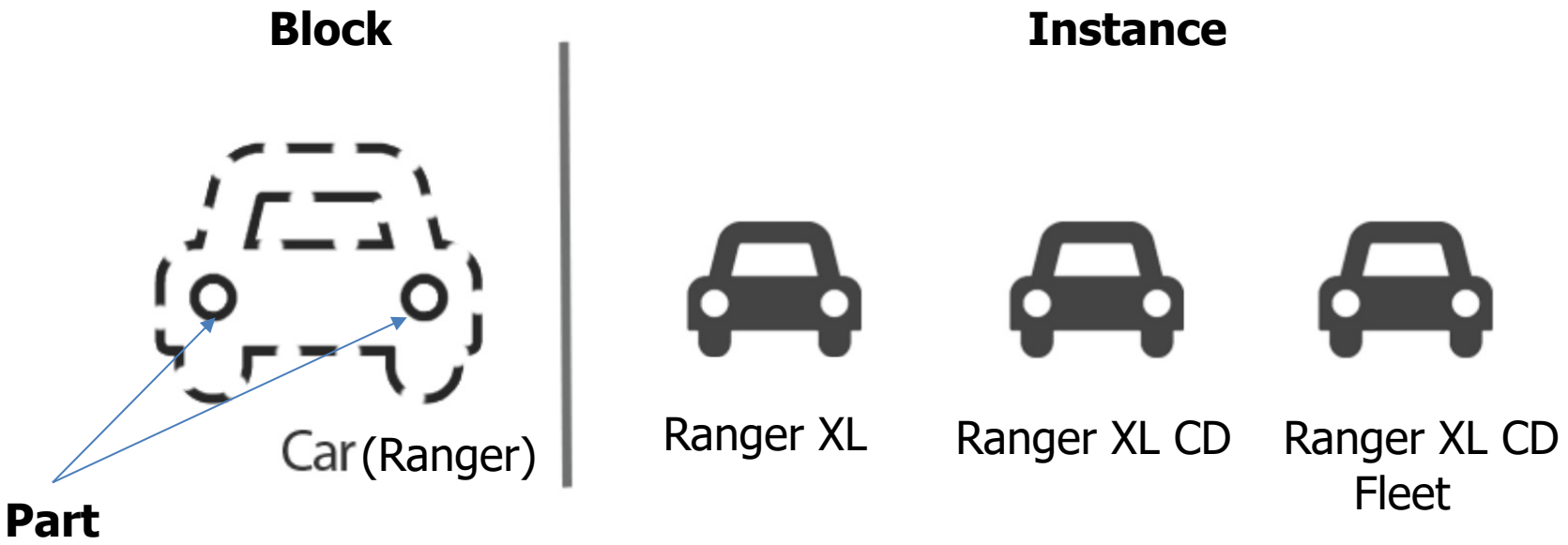


Instância

- Elemento para representar um uso real de um bloco
- Os valores das propriedades devem ser concretizados
- Elemento que representa o bloco em tempo de execução
- Podem ser utilizadas no contexto de simulação

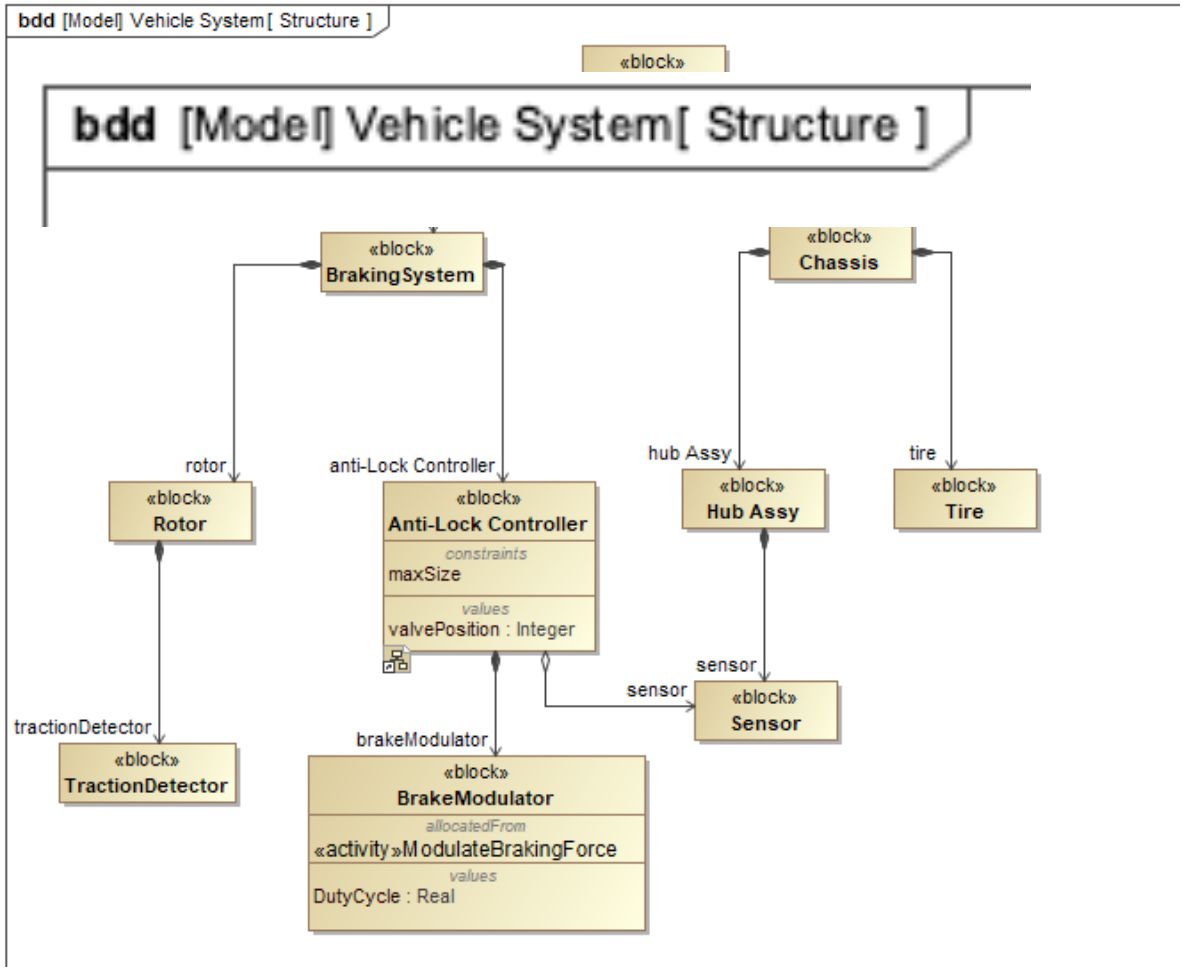


Block / Part / Instance



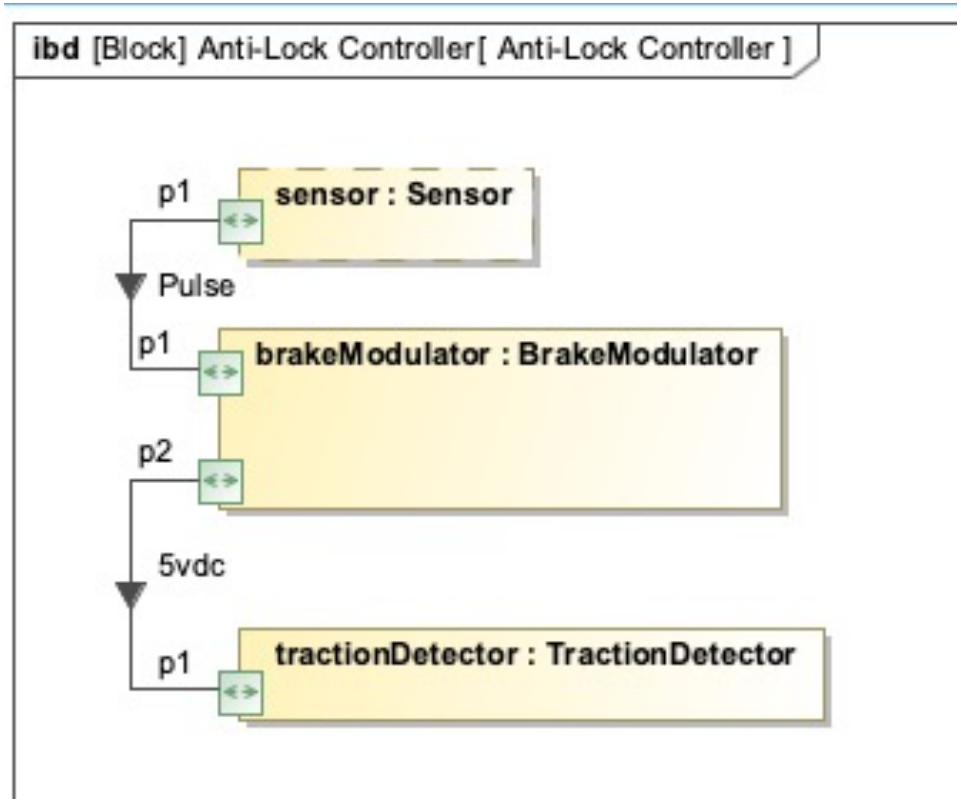
Diagramas de Blocos

- BDD – Block Definition Diagram



Diagramas de Blocos

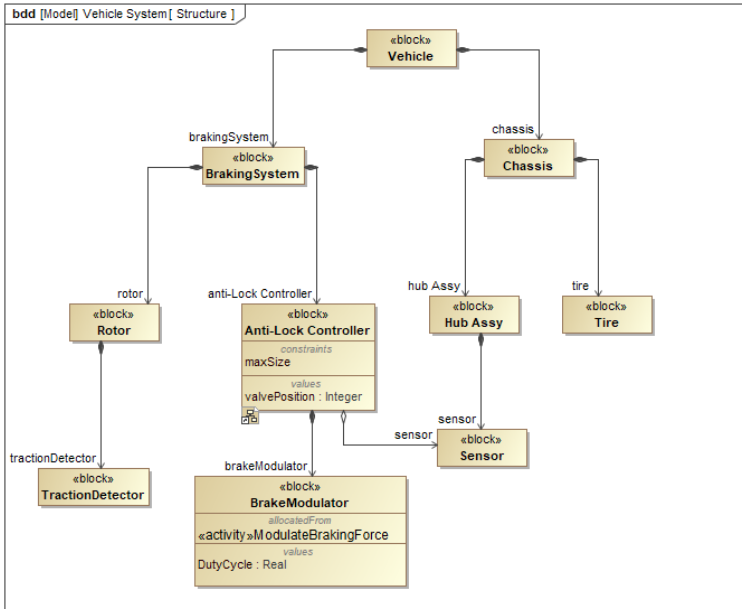
- IBD – Internal Block Diagram



BDD vs IBD

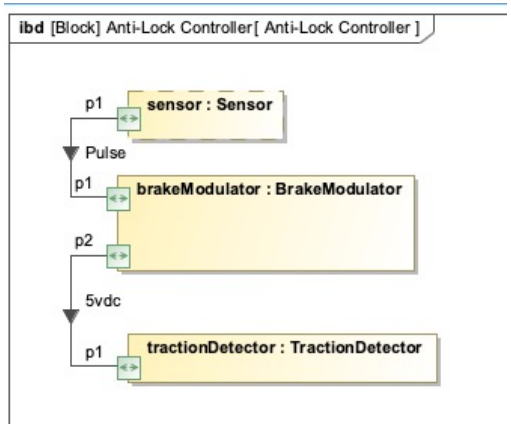
BDD

- Blocos são definições/tipos
- Capturam propriedades
- Reusados em vários contextos



IBD

- Trata da conexão das partes que compõem um bloco
- Parte é o uso de um bloco no contexto de uma composição

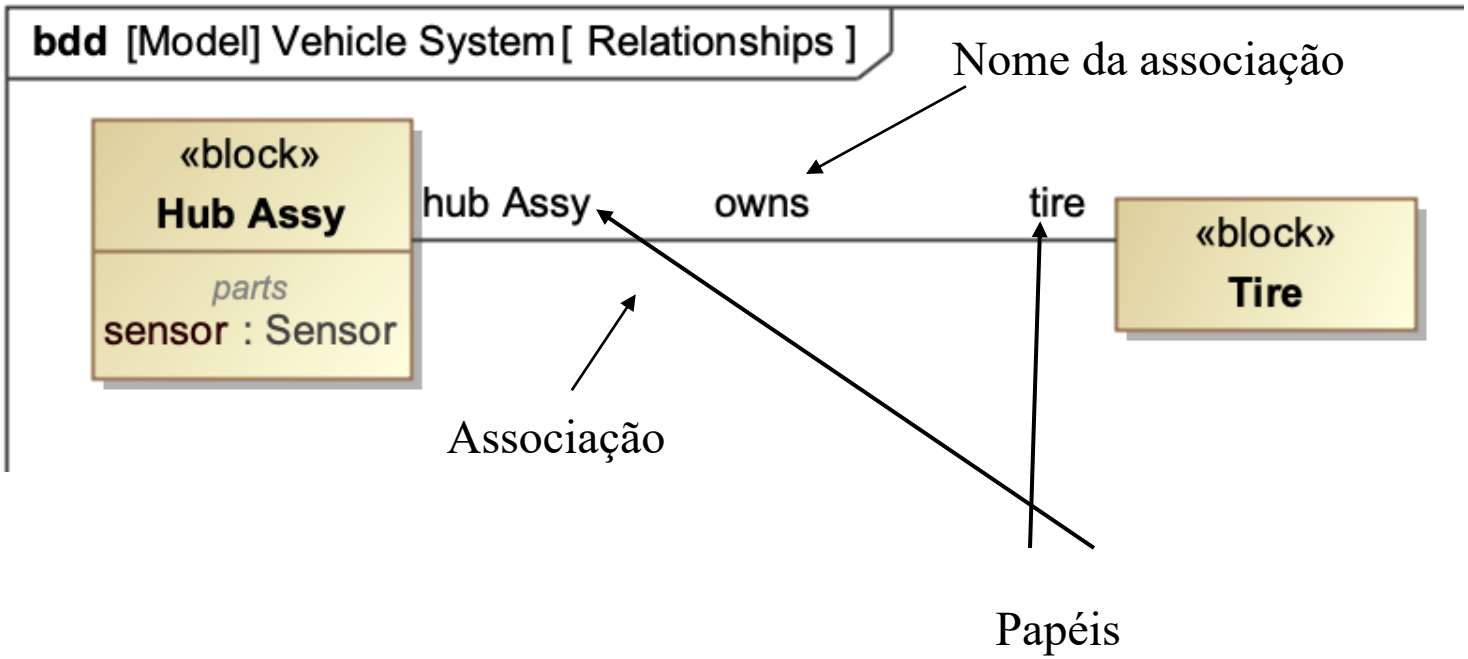


Relacionamentos entre Blocos

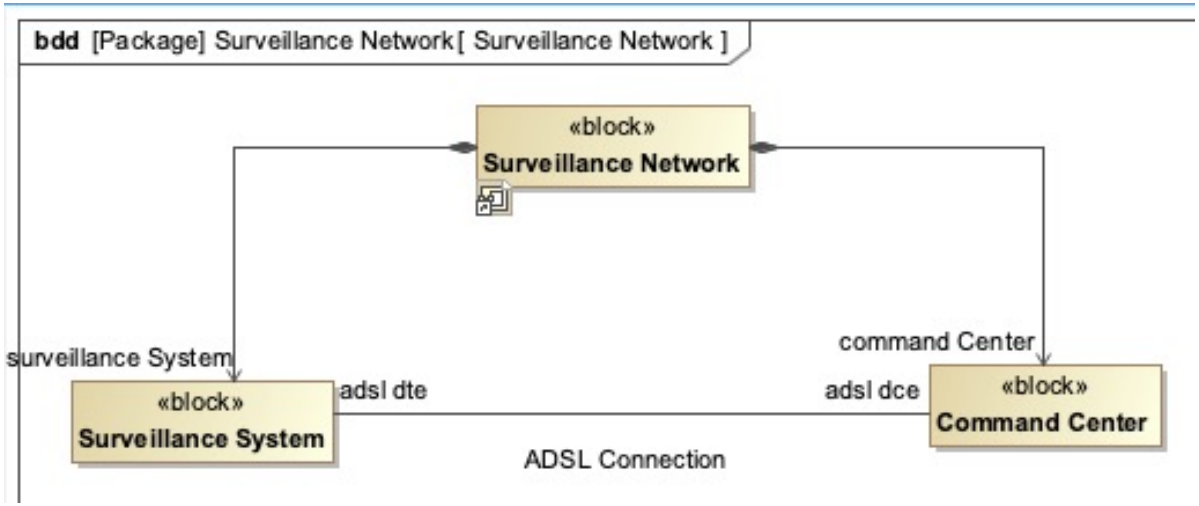
- BDD
- Associação
  - Composição
  - Agregação
  - Simples
- Generalização
- Realização
- Uso ou Dependência (*Usage)*

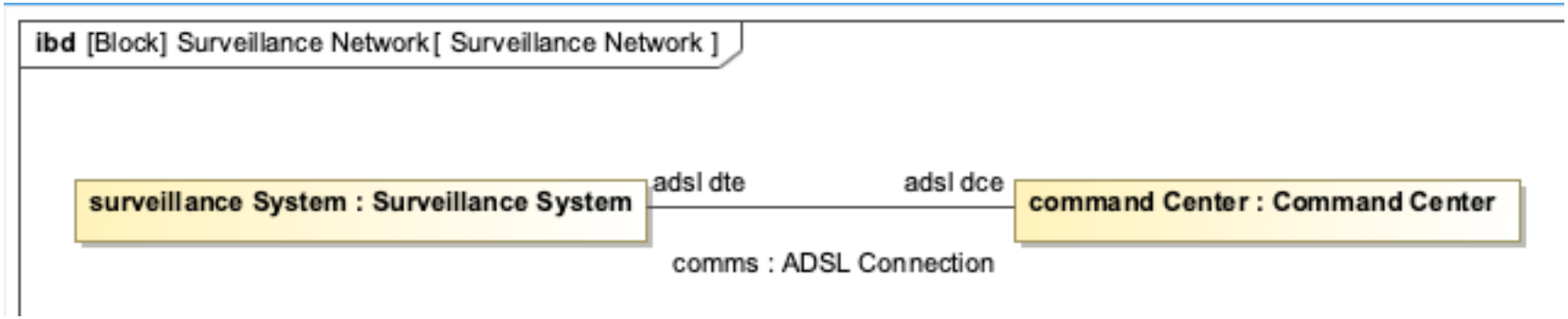
Associação

- Relação estrutural entre blocos

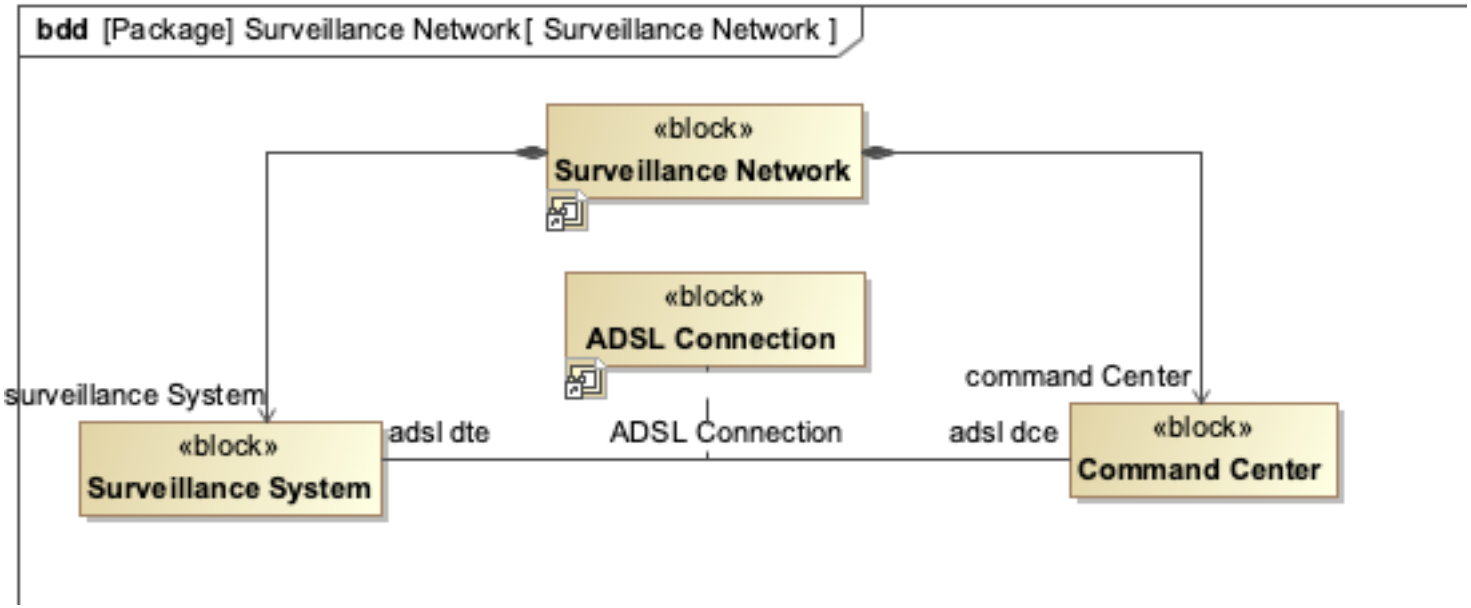


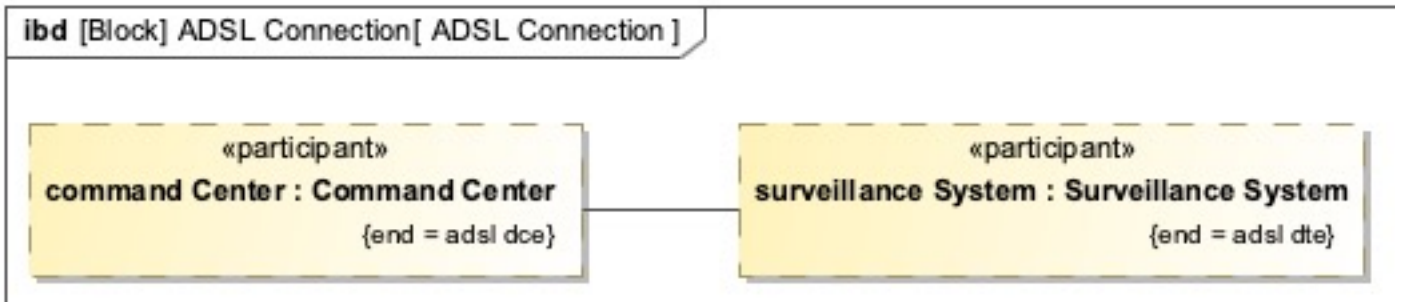
Associação – Tipificando associações





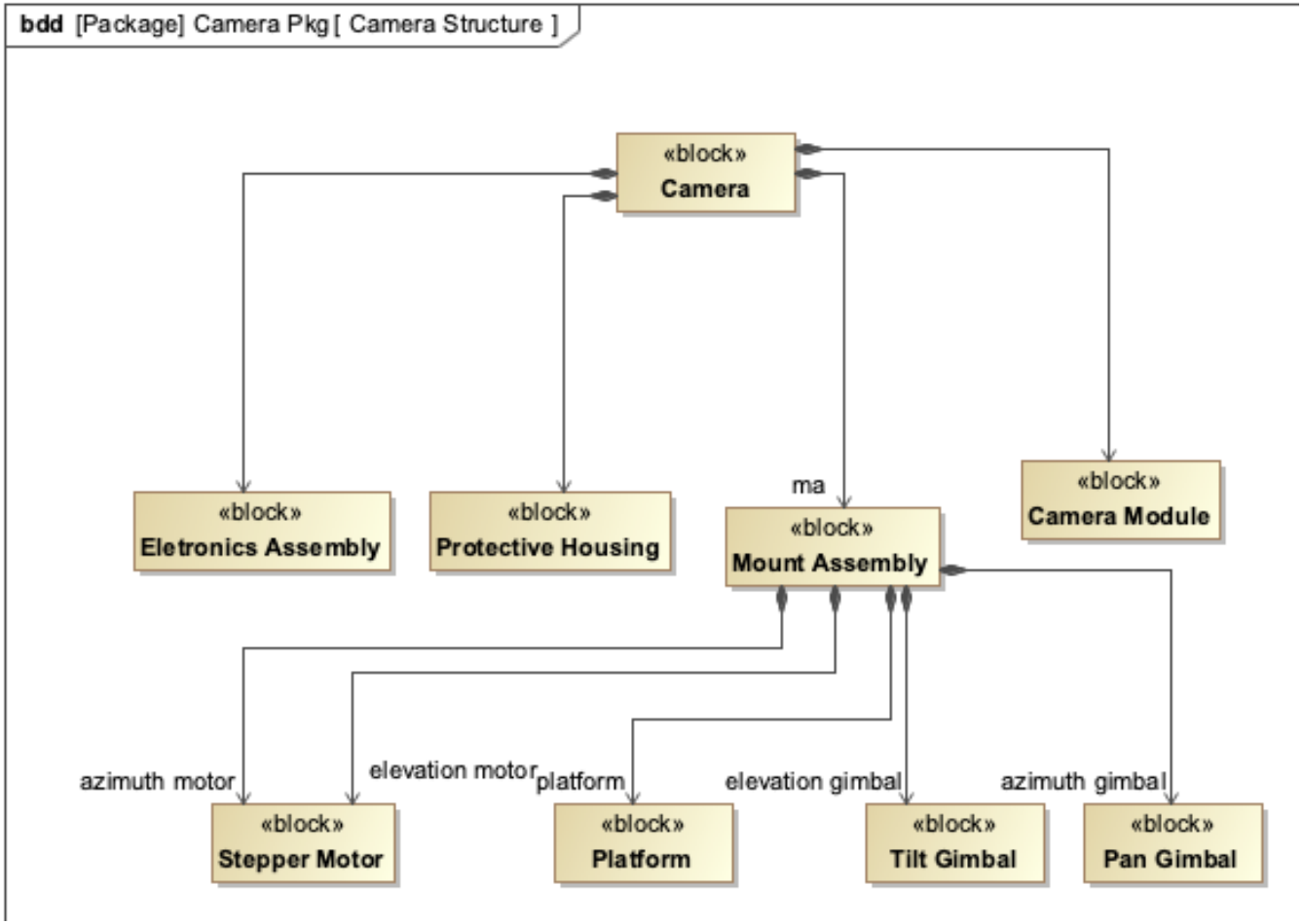
Associação – Tipificando associações





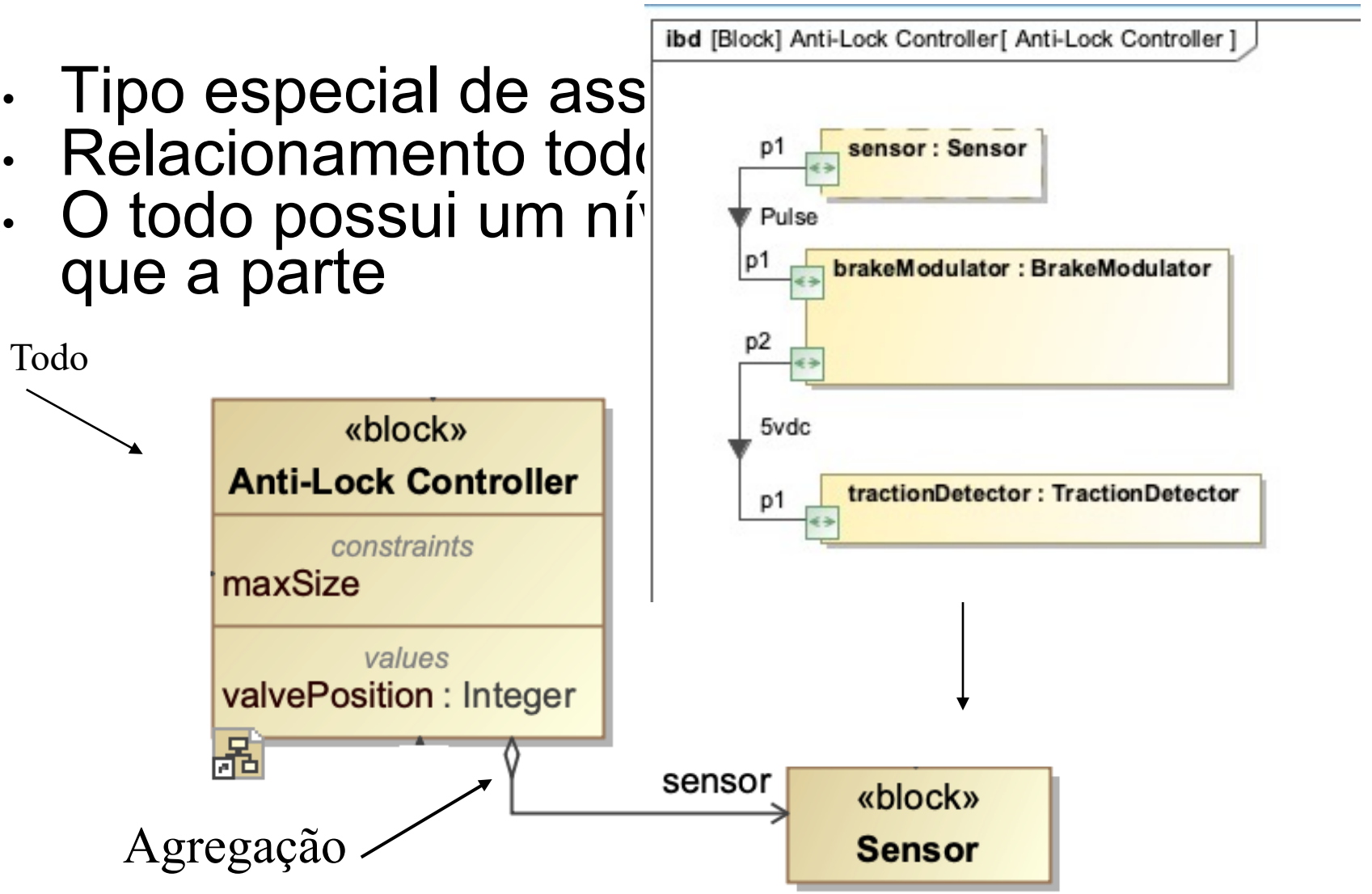
Papéis

- BDD – Block Definition Diagram - Nomeclatura



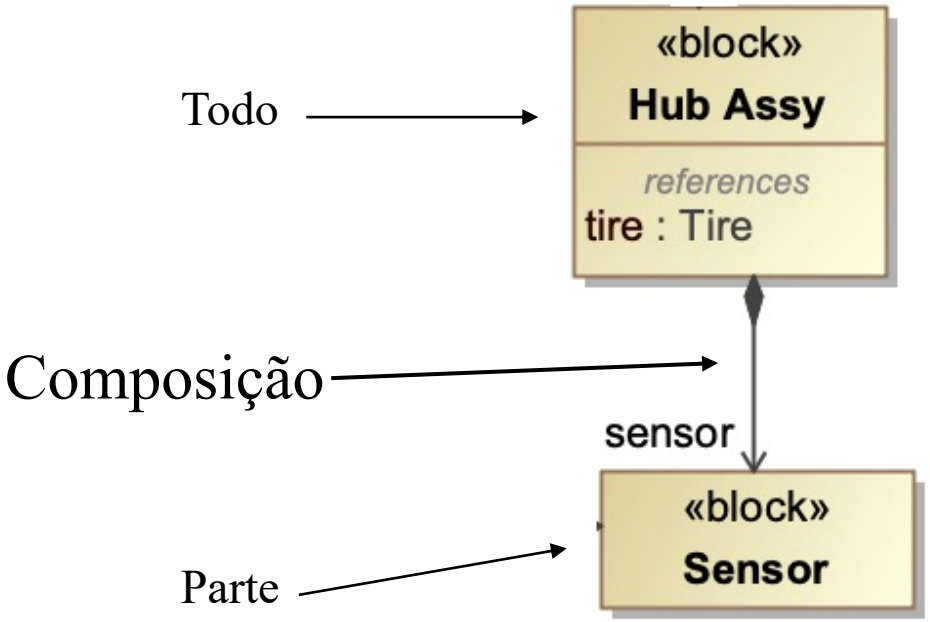
Agregação

- Tipo especial de ass
- Relacionamento tod
- O todo possui um ní
  que a parte



Composição

- Tipo especial de agregação
- Relação de posse mais forte
- O todo é responsável pela criação da parte
- A parte não vive sem o todo

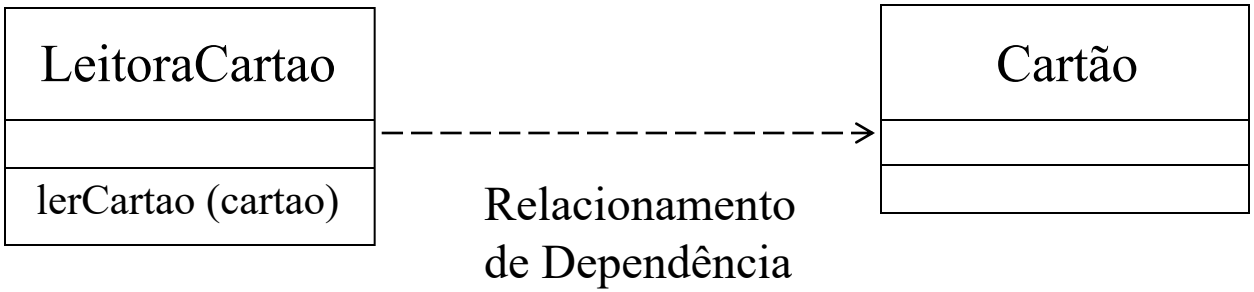


Exercício

- Crie um BDD que modela o domínio automobilístico onde devemos considerar os elementos internos como também externos. Seu modelo deve considerar elementos para veículo, passageiros, motorista, e ambiente externo com Estrada (com inclinação e fricção), Atmosfera (com temperatura e densidade do ar), e EntidadeExterna (elemento que pode ser visualizado pelo motorista). O detalhamento da estrutura interna do veículo não precisa ser realizada neste momento, pois o foco são nos elementos citados. Use somente os relacionamentos abordados até o momento.

Dependência ou Uso

- Relacionamento não estrutural
  - mais fraco que associação
- Uma dependência entre dois elementos indica que mudança em um elemento pode causar mudanças no outro

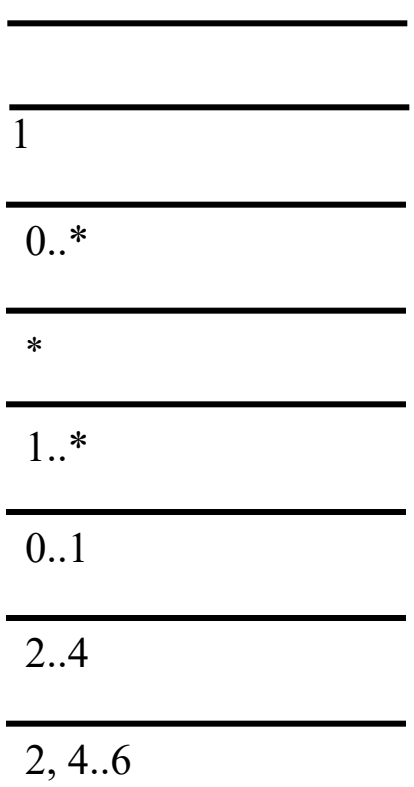


Multiplicidade

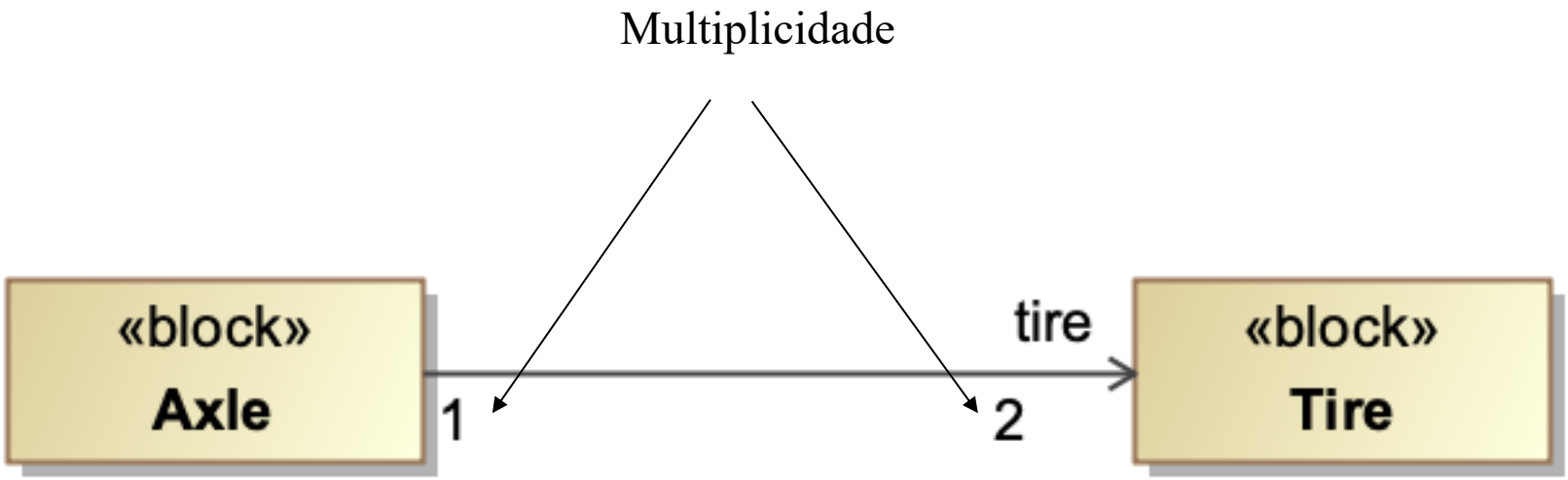
- Multiplicidade define quantas partes participam do relacionamento
  - O número de partes de um bloco relacionadas a uma parte de outro bloco
  - Especificado em cada uma das pontas do relacionamento

Tipos de Multiplicidade

- Não especificada
- Exatamente um
- Zero ou mais
- Muitos (mesmo que 0..*)
- Um ou mais
- Zero ou um
- Intervalo determinado
- Valores múltiplos

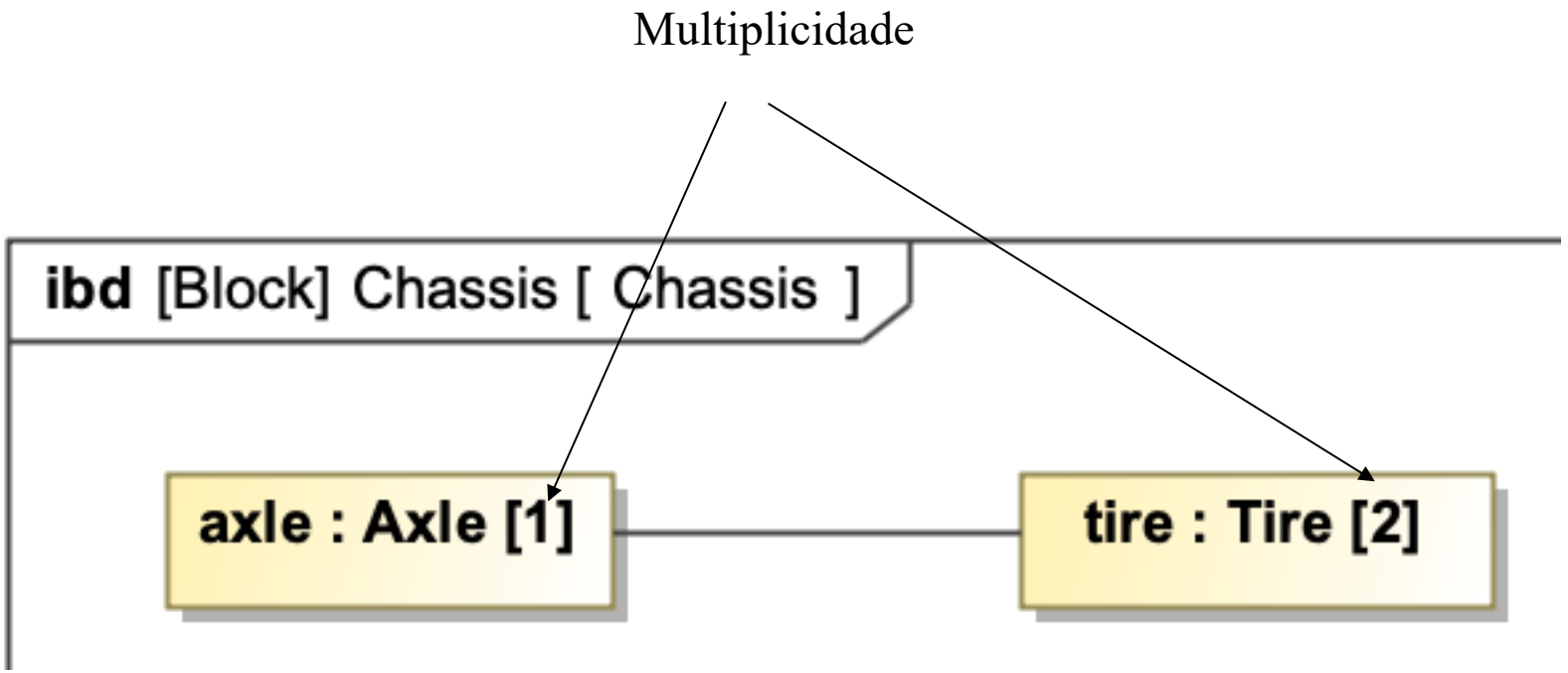


Exemplo: Multiplicidade BDD



É uma boa prática especificar as multiplicidades no BDD e manter sua consistência no IBD!

Exemplo: Multiplicidade IBD



É uma boa prática especificar as multiplicidades no BDD e manter sua consistência no IBD!

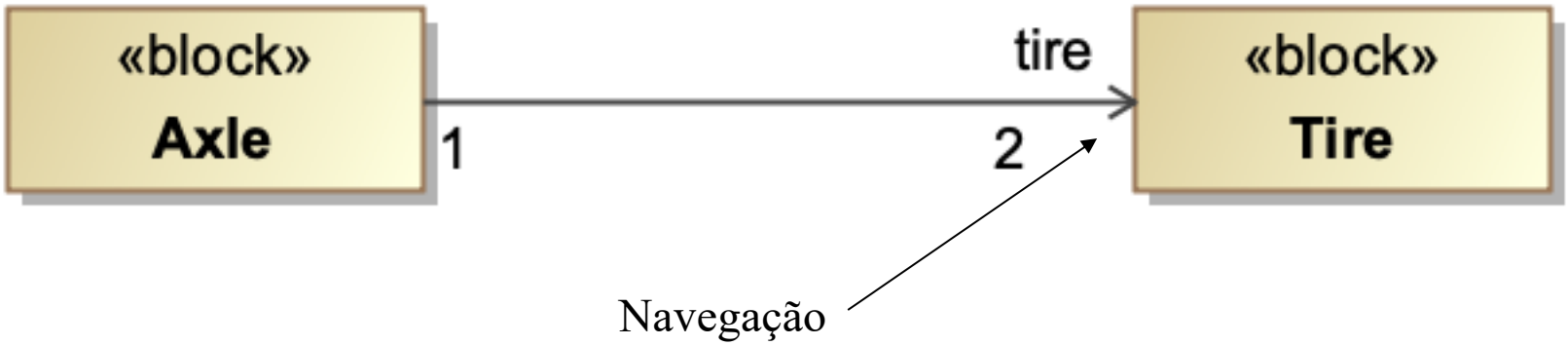
Exercício

- Incremente seu modelo colocando multiplicidade nos relacionamentos que você acha relevante.

Navegação

- Especifica a direção da associação
- Associações são bidirecionais por *default*, mas é desejável que a navegação seja restringida a apenas uma direção
- Associações bidirecionais são mais difíceis de implementar e acoplam o modelo

Exemplo: Navegação



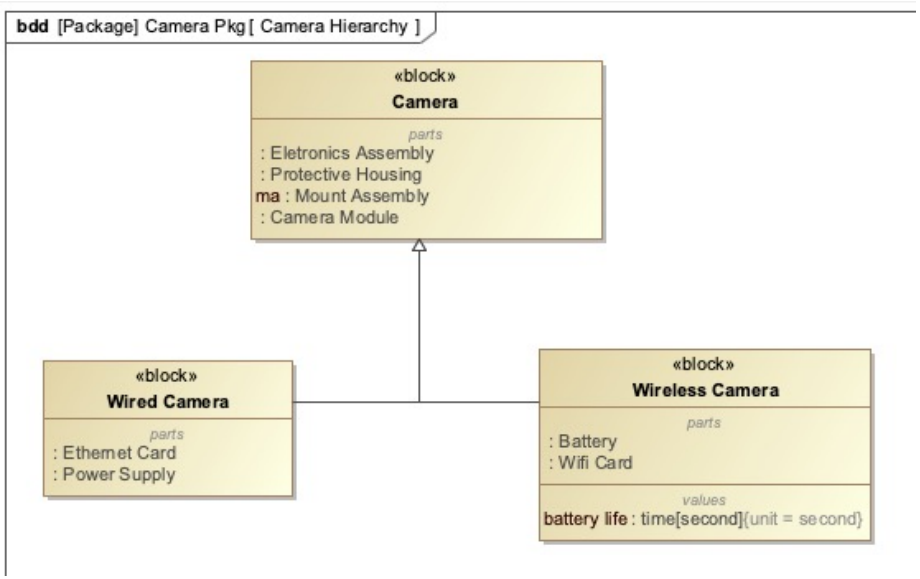
É uma boa prática sempre que possível especificar a direção da navegação para deixar o modelo consistente em termos de acoplamento.

Exercício

- Cheque o seu modelo e veja se existe locais onde a navegação é interessante de ser adicionada.

Generalização/Especialização

- Relacionamento entre blocos onde um bloco compartilha a estrutura (propriedades e relacionamentos) e comportamento (operações) com outros blocos
- Define uma hierarquia de abstrações



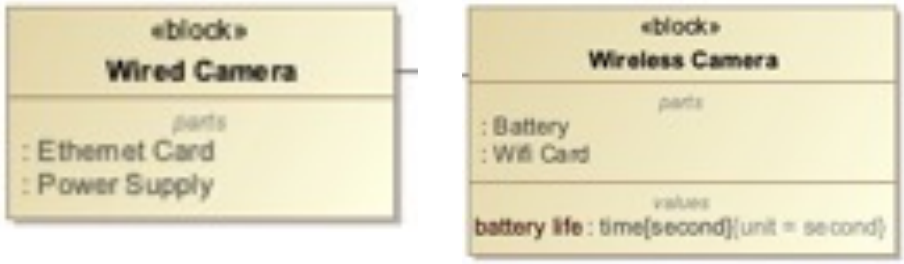
Generalização/Especialização

Superclasse

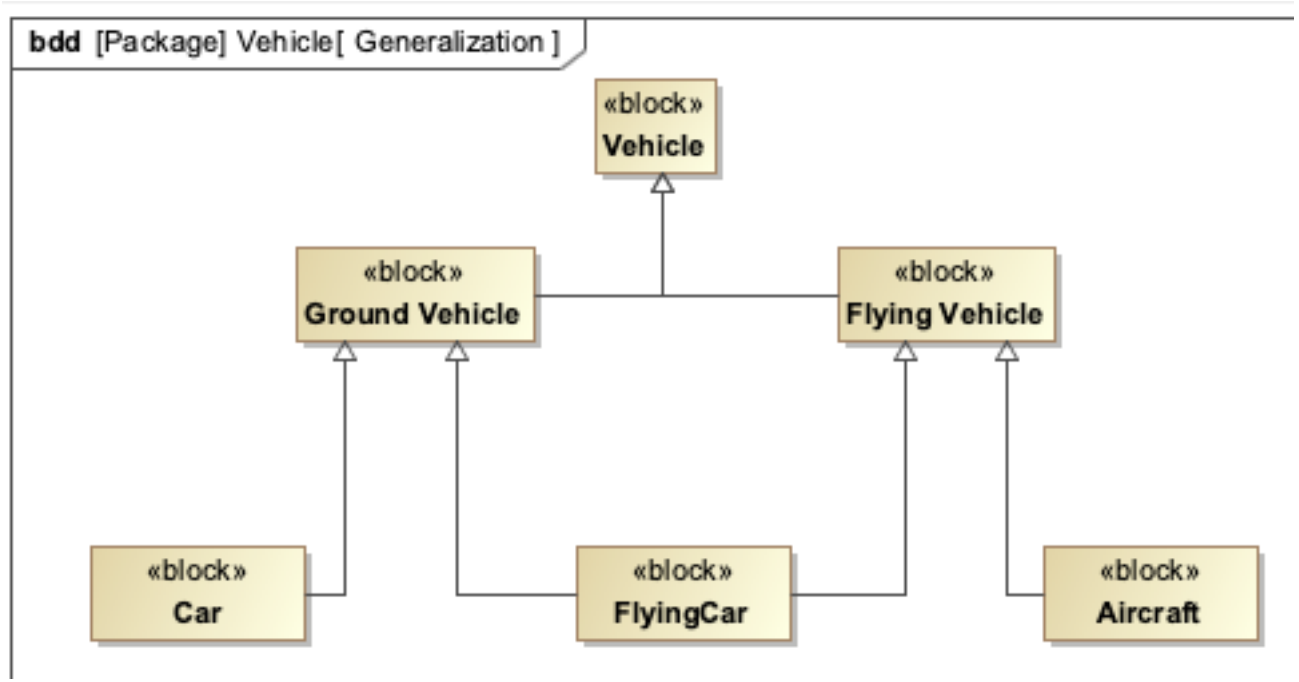
- Elemento pai da hierarquia
- Possui elementos comuns que são compartilhados

Subclasse

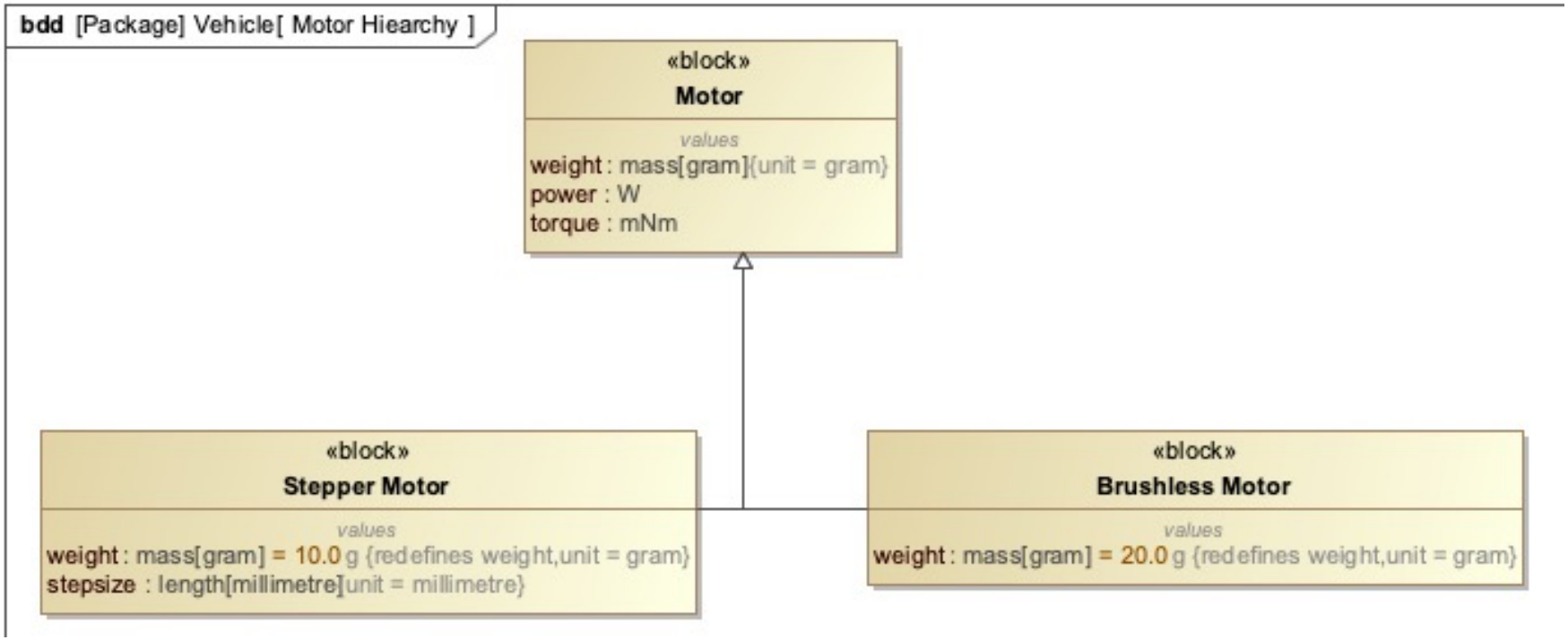
- Elementos filhos
- Possui elementos especializados
- Reusa elementos da superclasse
- Pode redefinir elementos da superclasse



Generalização Simples vs Múltipla

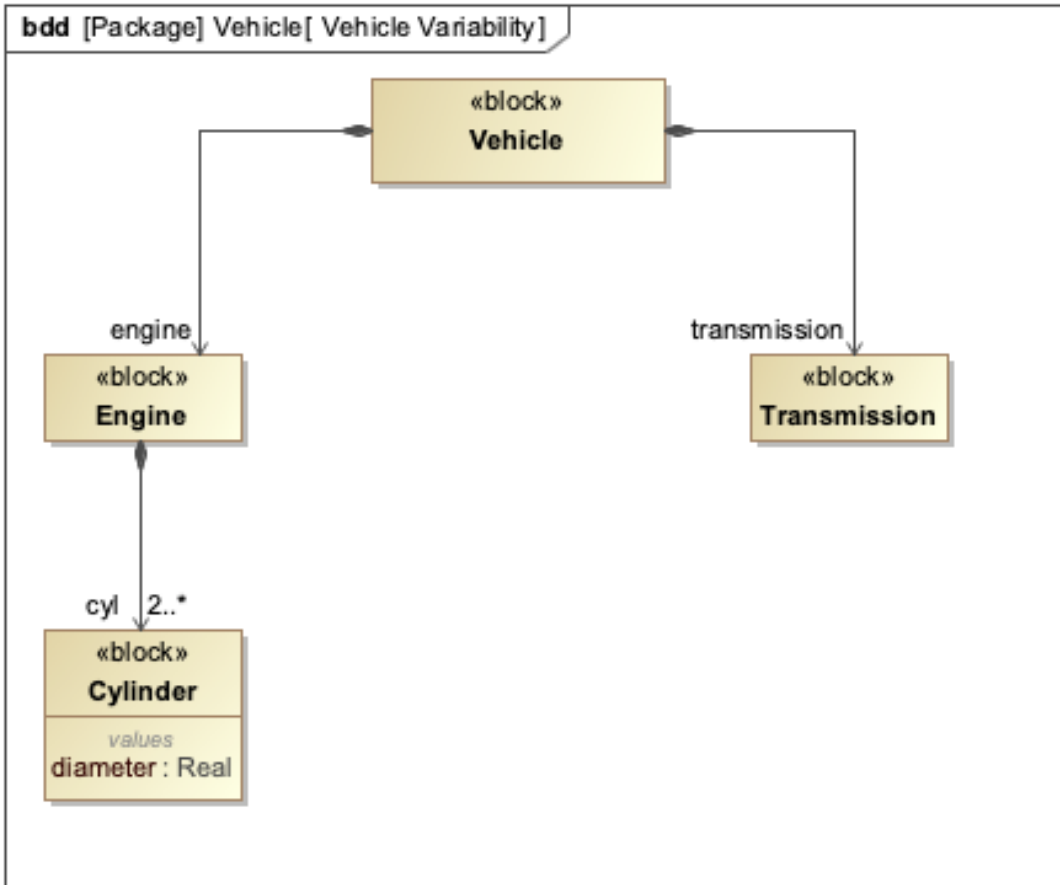


Redefinição

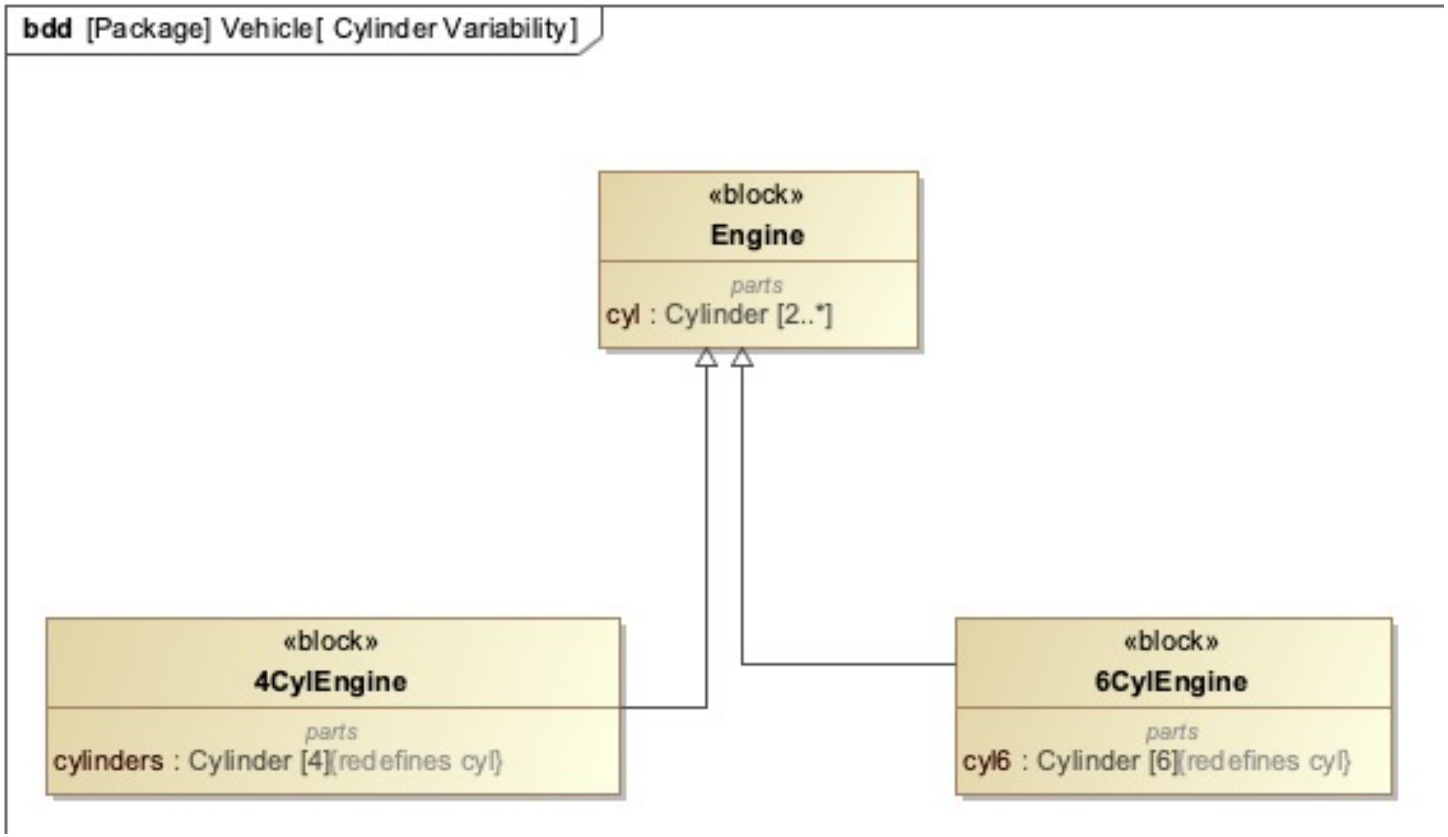


Modelando Variabilidade com Generalização

- Como modelar um veículo que possa ter variações de motores com 4 e 6 cilindros?



Modelando Variabilidade com Generalização

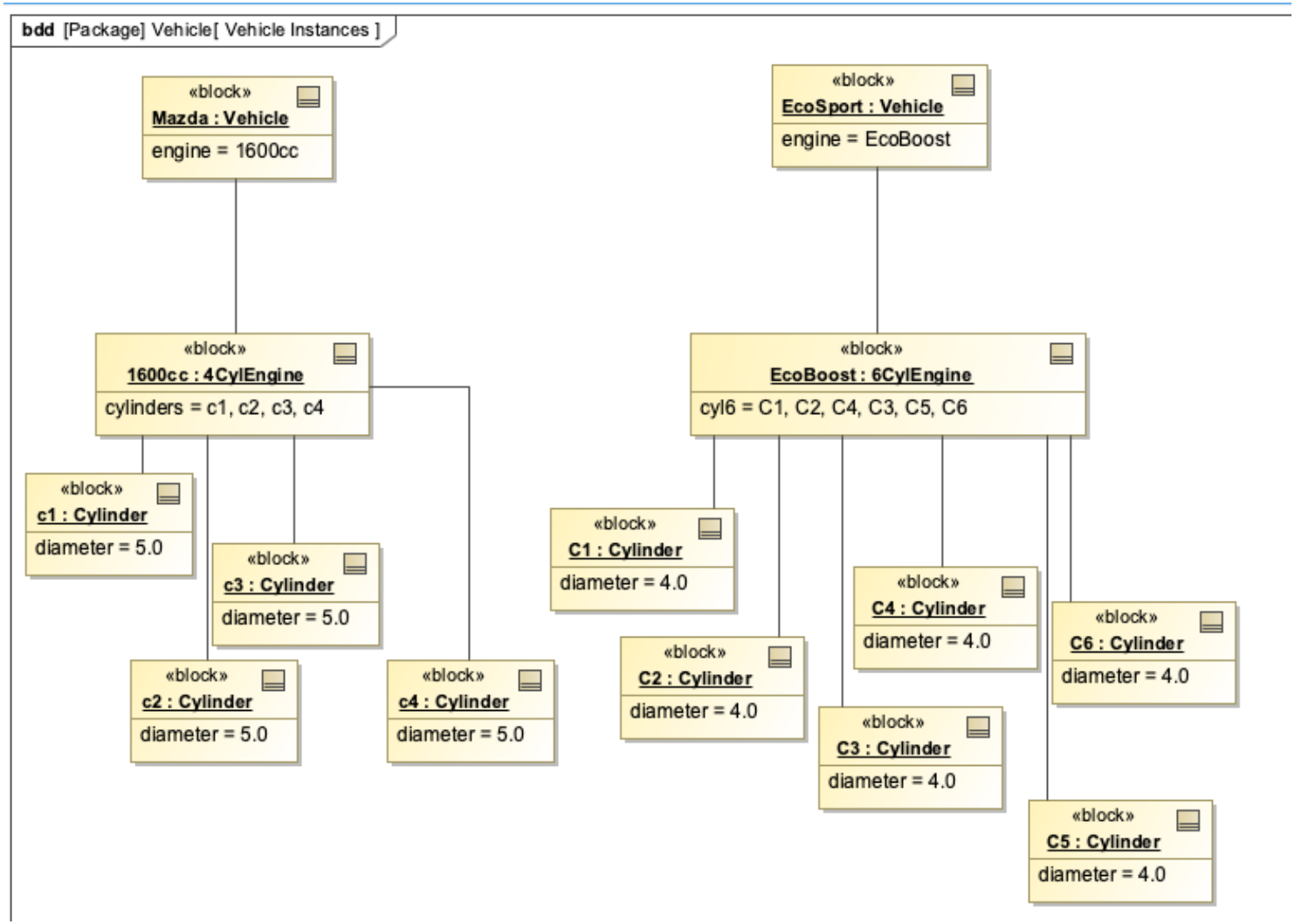


Exercício

- Verifique no seu modelo elementos onde generalização/especialização podem ser aplicados e atualize-o.

- Construa um IBD para contextualizar as conexões entre o seu veículo, o ambiente externo (estrada, entidade externa, atmosfera), e o motorista. Não use portas no momento. Tente utilizar apenas conexões entre as partes. Considere as seguintes conexões (sem tipá-las por enquanto):
  - O motorista deve interagir como o veículo com um comando de aceleração e seleção de marcha
  - Entidades externas podem ser visualizadas pelo motorista
  - O veículo deve receber informações da atmosfera
  - O veículo deve poder se mover pela estrada através do uso de torque

Modelando Configurações Usando Instâncias



Exercício

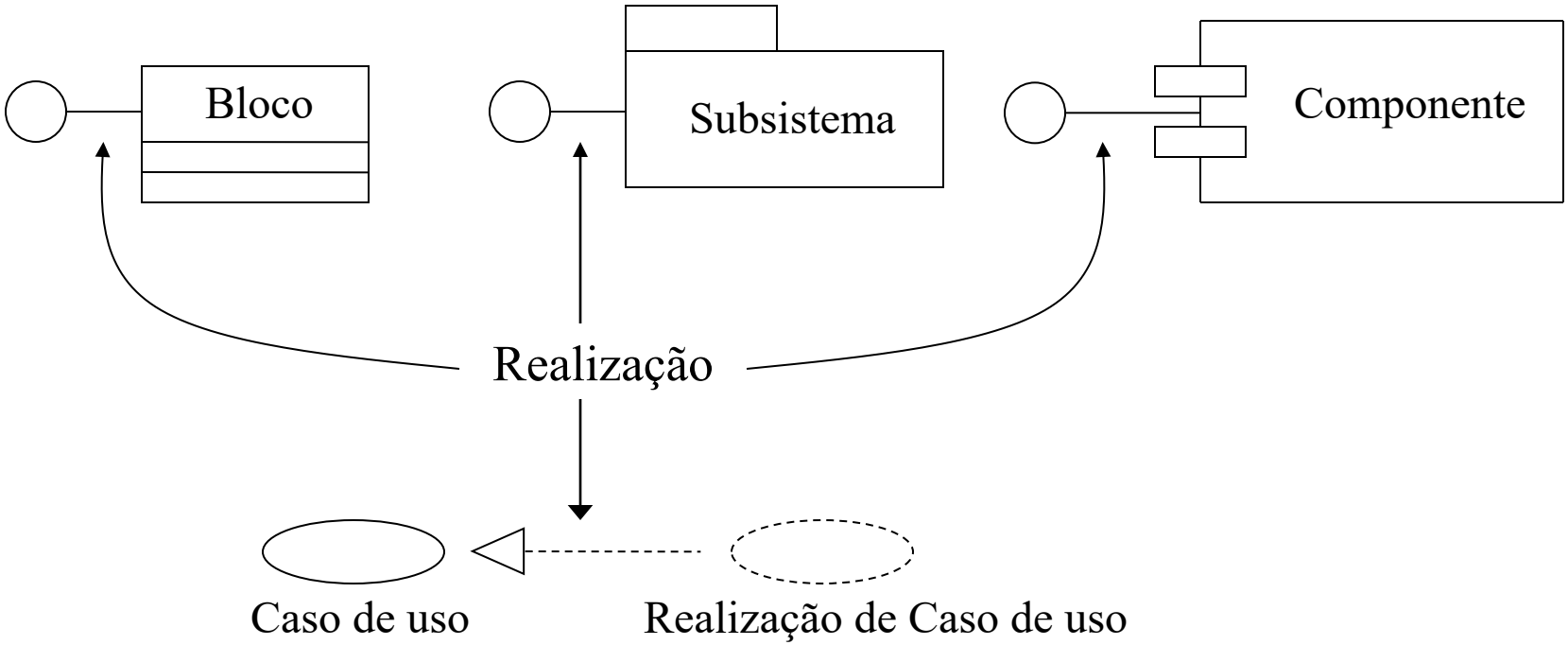
- Crie diferentes configurações para o ambiente externo, com estradas e atmosferas que reflitam diversas configurações.

- Altere o seu modelo para permitir a modelagem de veículos com Sistema de transmissão, os quais podem ser manual e automático. Modelar Veículos com Sistema de Transmissão Manual e Automático e suas variações. O primeiro considere que o Sistema possui apenas 5 marchas. No modo automático temos no mínimo 4 marchas e sem limite superior. Por fim, crie diferentes configurações de veículos com diferentes sistemas de transmissão.

Realização

- Indica que um elemento serve como contrato que o outro deve seguir

Exemplos:



Interface

- Interfaces definem um tipo especificando apenas a assinatura das operações
- Idealmente, interfaces deveriam prover contratos

Usadas para especificar um conjunto de características comportamentais, que, em geral, são realizadas por portas no modo <<provided>> ou <<required>>.

E InterfaceBlock?

InterfaceBlock

- Um tipo especial de bloco que não contém estrutura nem comportamento
- Usadas junto com portas proxy para:
  - tipificar portas
  - encapsular portas

Modelando Fluxos

- Em geral, fluxos modelam elementos físicos
- Ex: Bomba de Água
  - **Água** que flui para dentro e fora
  - **Eletricidade** que flui para dentro da Bomba

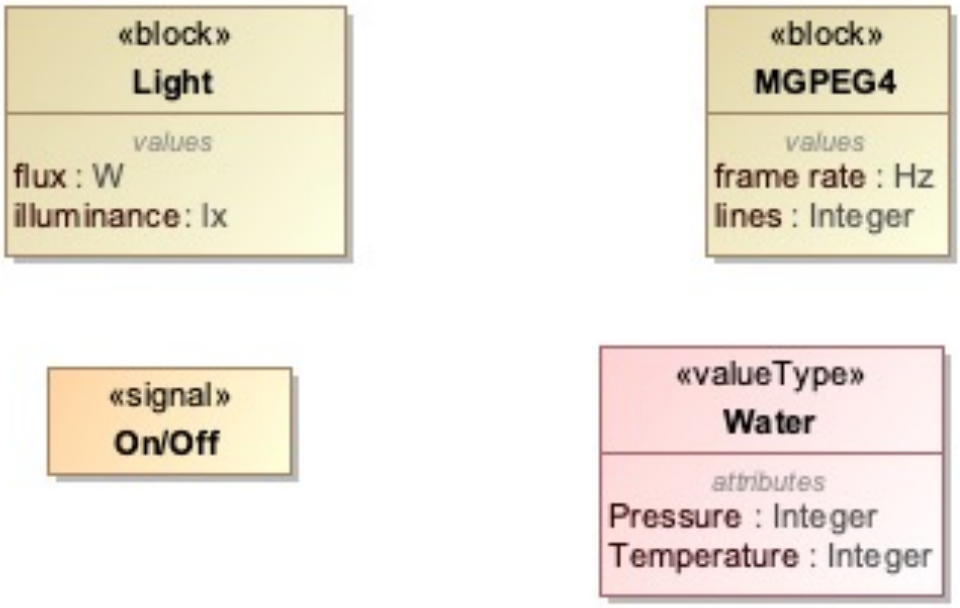
•Mas também podem ser informação ou sinais em sistemas elétricos

Modelando Fluxos

- *Item* é o termo usado para representar elementos que fluem entre partes
  - Podem ser tipificados por ***Blocos***, ***ValueType*** ou ***Signal***
- Item <- Bloco: Descrever itens complexos
  - Ex: **Água** pode ter valueProperties para representar **Pressão** e **Temperatura**
- Item <- ValueType: Elemento simples quantificável
  - Ex: TempAgua para representar a temperatura da águ
- Item <- Signal: Controle de comportamento
  - Ex: Sinal de ligado/desligado

Modelando Fluxos

- *Item* é o termo usado para representar elementos que fluem entre partes

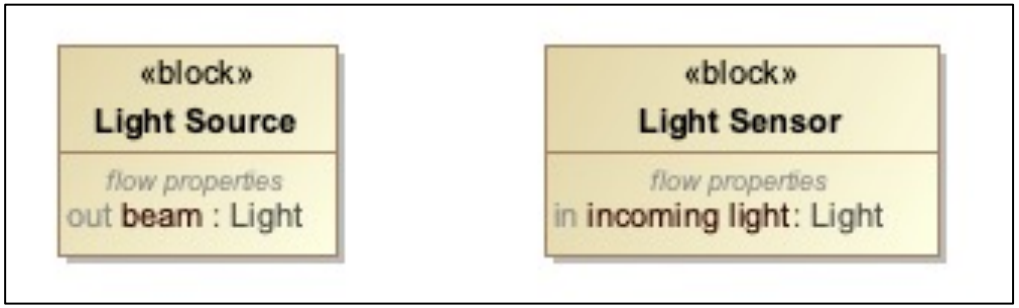


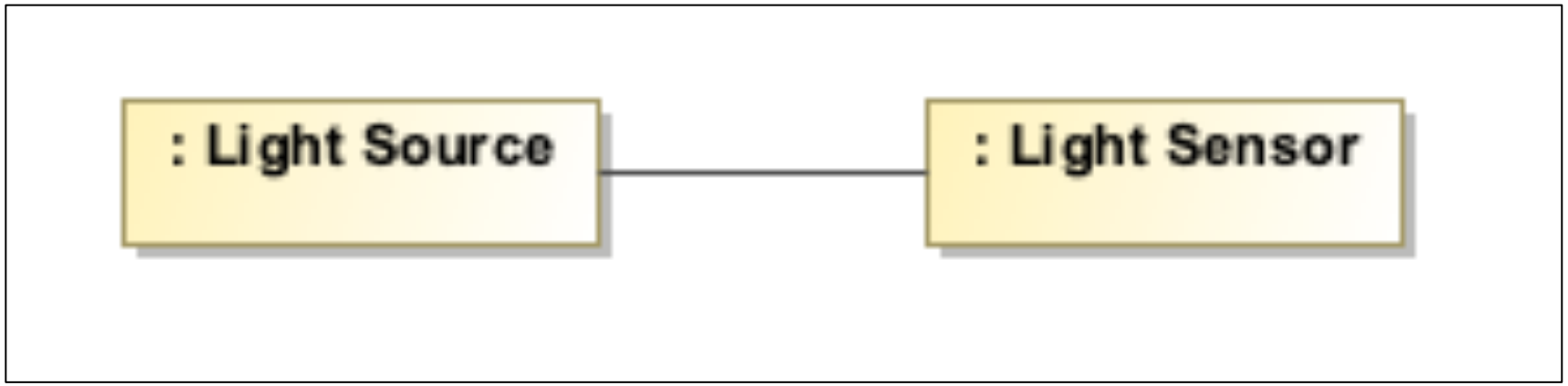
Modelando Fluxos

- Blocos podem tem FlowProperties para indicar propriedades que fluem para dentro e/ou fora dele.
- Elas contém: nome, tipo, multiplicidade e direção (in, out ou inout).

Modelando Fluxos no IBD

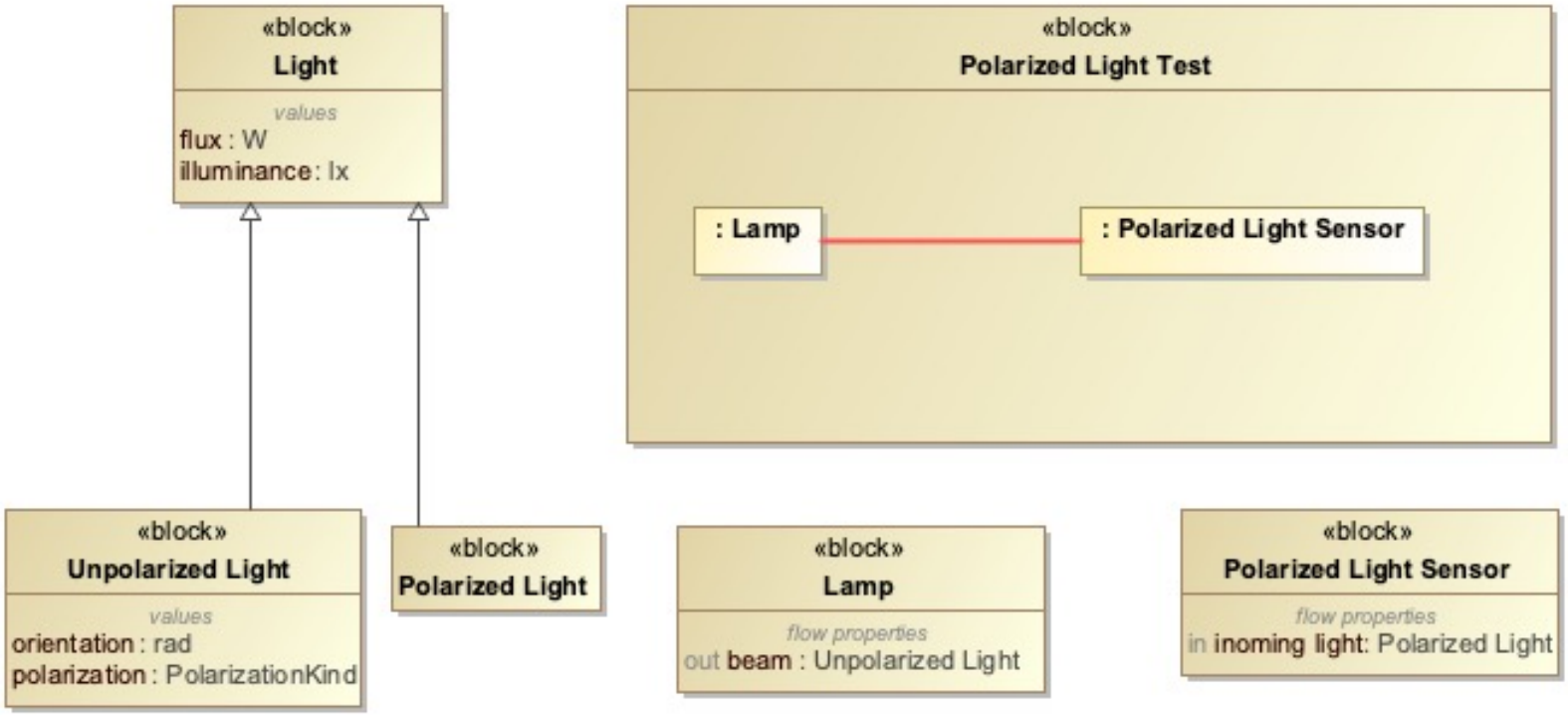
- Compatibilidade de FlowProperties





Modelando Fluxos no IBD

- Incompatibilidade de Fluxo



Exercício

- Altere o IBD do domínio automobilístico para considerar fluxos entre as conexões das diferentes partes. Relembrando:
  - O motorista deve interagir como o veículo com um comando de aceleração e seleção de marcha
  - Entidades externas podem ser visualizadas pelo motorista
  - O veículo deve receber informações da atmosfera (ar)
  - O veículo deve poder se mover pela estrada através do uso de torque

Comportamento de Blocos

- Comportamentos (Behaviors)
  - Global/Principal: executado quando o bloco é instanciado
  - Métodos/Operações: executados posteriormente ao longo do tempo de vida do bloco
- Um *behavior* pode invocar diversos outros *behaviors*
- *Behaviors* podem ter parametros de entrada, saída, e entrada/saída

Comportamento de Blocos

- Formalismos:
  - **Máquina de estado**: bloco reagindo a eventos
  - **Atividades**: fluxo de ações podendo transformar entradas em saídas
  - **Interações (diagrama de sequência)**: como as partes interagem através de troca de mensagens
  - **Comportamento Opaco**: comportamento textual em alguma linguagem externa
  - **Comportamento de função**: similar ao opaco só que não afeta o estado do bloco. Geralmente associado a funções matemáticas

Comportamento de blocos

- O que usar para modelar o comportamento principal de um bloco?
- Alguns exemplos:
  - Máquina de estados que descreve o funcionamento reativo de um bloco
  - Atividades para descrever um fluxo ativo de ações
  - Híbrido Máquina de estado com atividades

Comportamento de Blocos

- Tipos de *features* comportamentais de um bloco:
- Operações: comportamento geralmente disparado por uma requisição síncrona
- Recepção de Sinais: comportamentos somente disparados assíncronamente

•Ambos podem ter parâmetros de entrada, saída, e entrada/saída

Exercício

- Crie os blocos abaixo como parte de um veículo e defina as suas operações como descrito:
  - GPS
    - retornaPosicao() : Posicao
    - definaRota(Posicao atual, Posicao destino): Rota
    - enviaPosicao(Posicao atual): void
  - Multimedia
    - connectarBluetooth(Dispostivo d)
    - buscarDispositivosBluetooth(): Dispositivo[0..*]
    - atenderChamadaTelefonica(): void

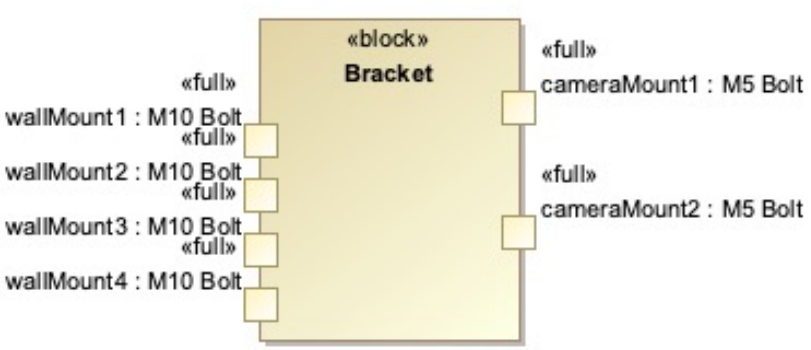
Modelando Portas

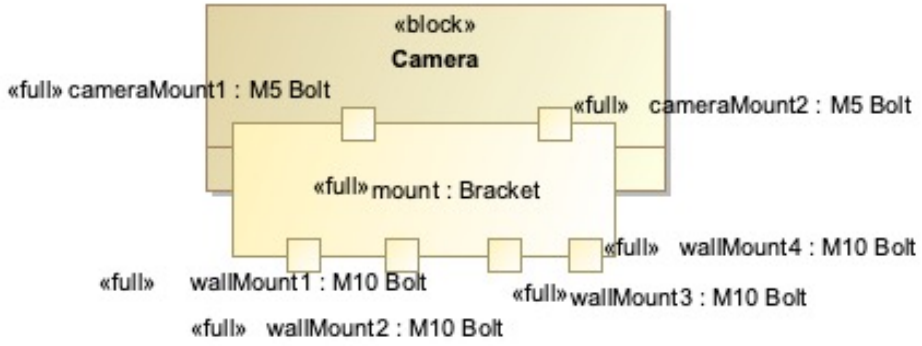
- Portas são pontos de acesso na fronteira de um bloco ou parte usados para encapsular comunicação
- Devem estar conectadas a outras portas
- Tipos:
  - Full port
  - Proxy port

Modelando Portas - Full port

- Similar a partes
- Possuem tipos
- Geralmente representam elementos físicos
- Podem ter portas aninhadas (*nested*)
- Diferença de uma parte é que quando a propriedade *isEncapsulated* do bloco que a contém é *true,* conectores externos não podem se ligar a ela

Modelando Portas - Full port



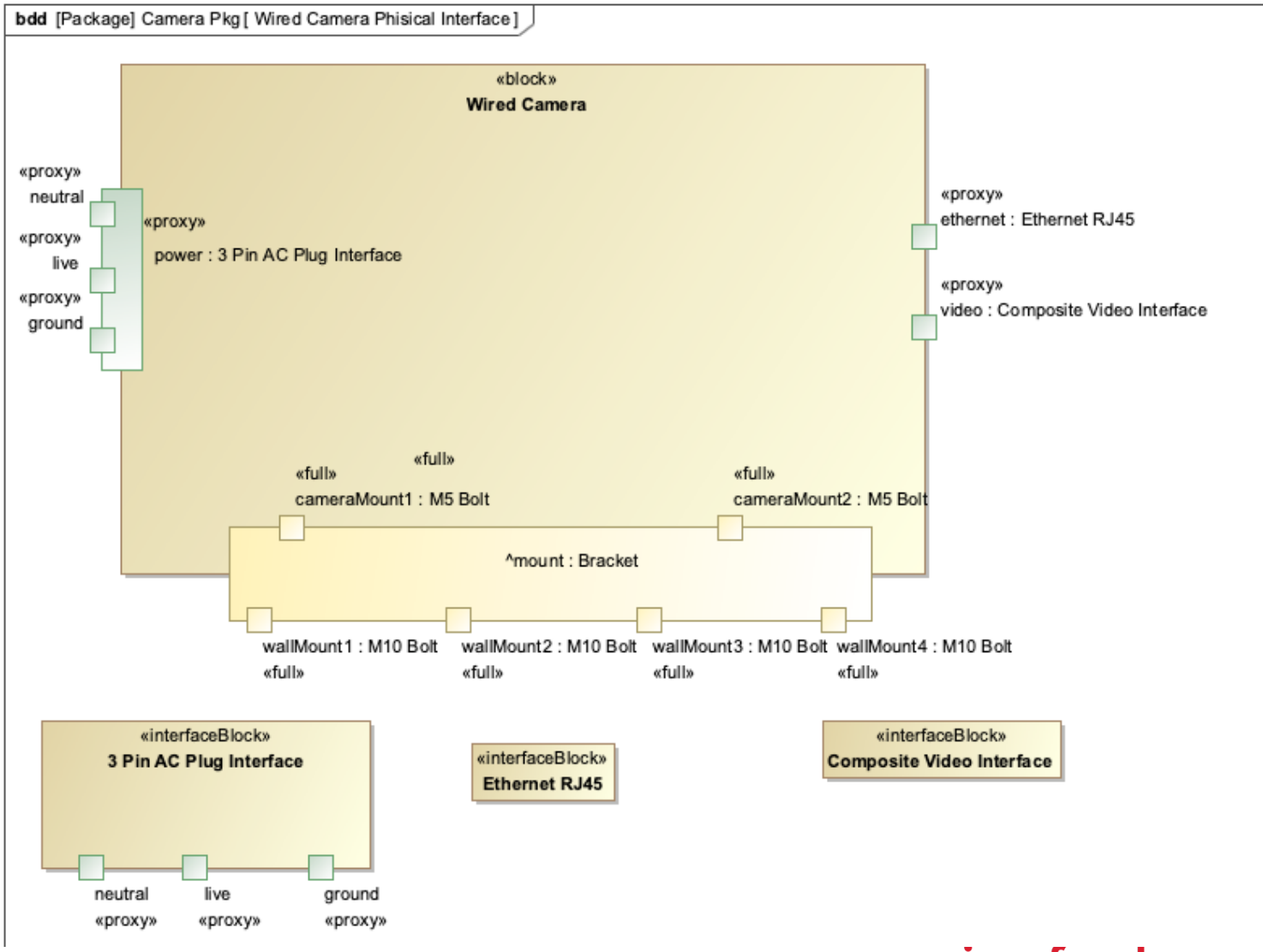


Modelando Portas - Proxy port

- Expõe elementos do bloco
- São tipadas somente por InterfaceBlocks
- Podem representar tanto elementos físicos quanto acesso a comportamentos (serviços)
- Não são partes como as full ports
- Podem ser aninhadas

Modelando portas

Proxy ports



Comparação full e proxy ports

Full Port

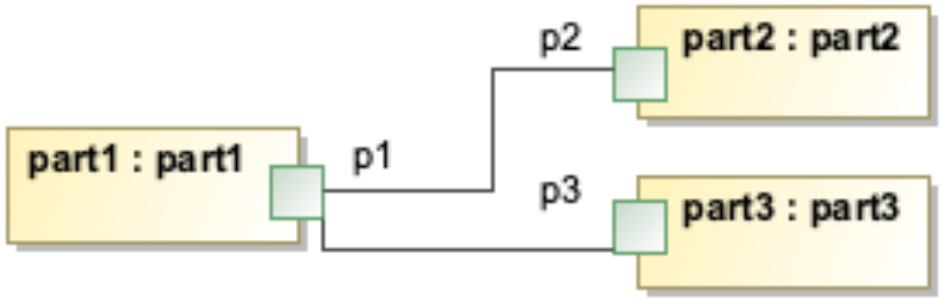
- É uma parte: tipada por bloco
- Normalmente representa elementos físicos
- Pode ter comportamento associado e mudar valores que fluem através dela

Proxy Port

- Não é uma parte: tipada por InterfaceBlock
- Não tem comportamento associado
- Portão de acesso aos elementos internos do bloco
- Não pode transformar elementos que fluem por ela
- Esconde elementos internos (expõe contrato de comunicação) – Black-box

Modelando Portas

- Conexão entre portas:
  - Portas podem se conectar a partes ou outras portas através de conectores
  - Portas podem estar conectadas a mais de uma outra porta ou parte.
  - Cada conexão deve ter seu próprio conector

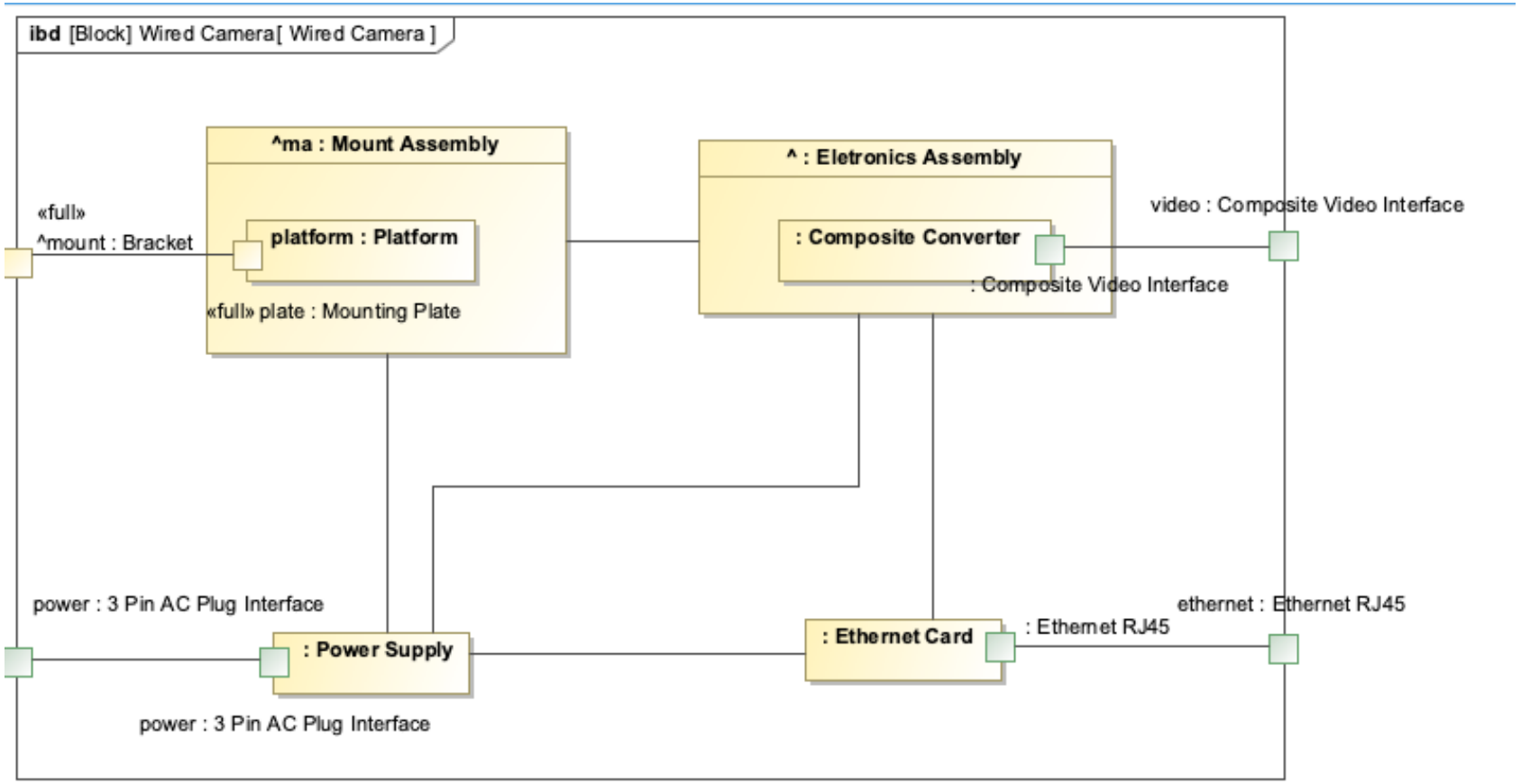


Modelando Portas

- Conexão entre portas:
- Regras de compatibilidade externa são iguais para full e proxy
- Regras de compatibilidade interna diferentes entre full e proxy:
- Full ports: podem estar conectadas a partes ou outras full ports de tipos diferentes
- Proxy ports: tipos das conexões internas devem ser compatíveis

Modelando Portas

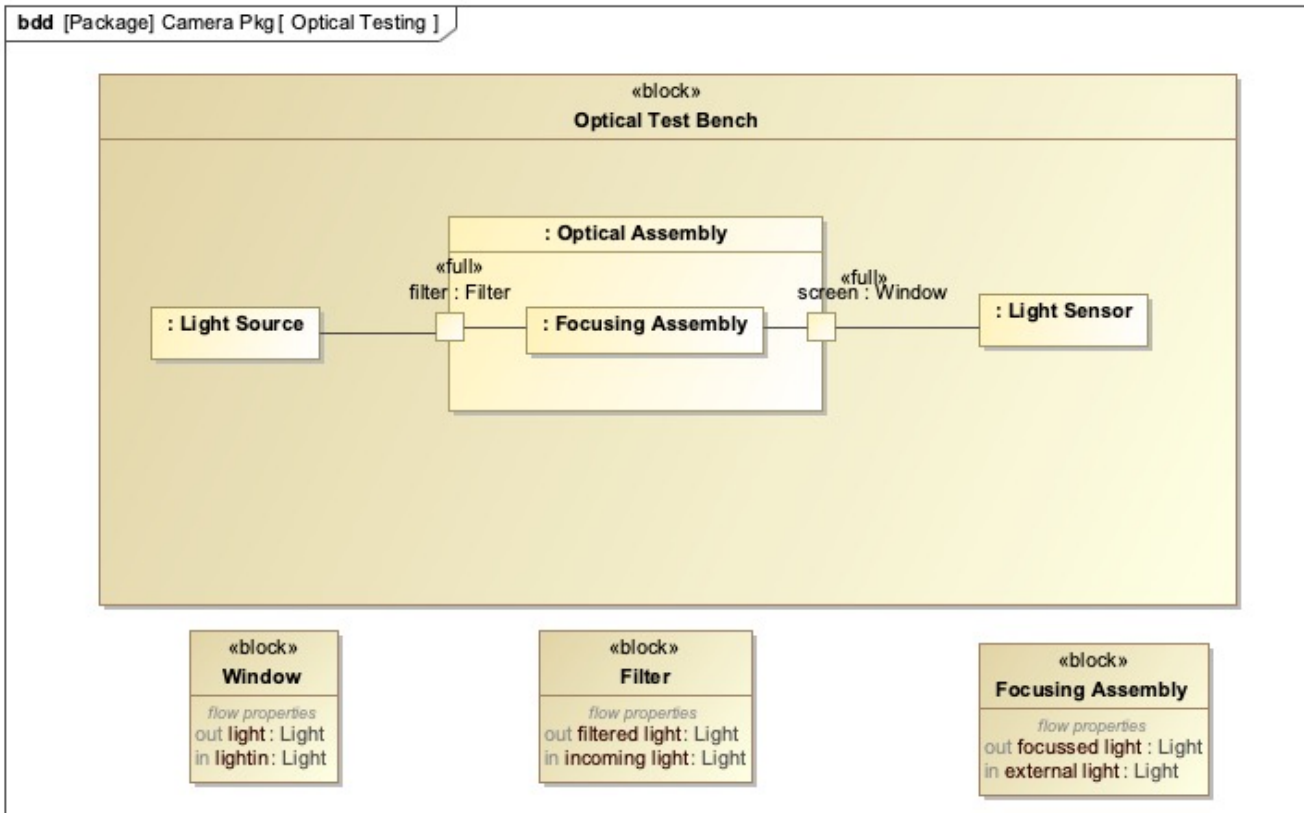
- Conexão entre portas:



Modelando Portas

Conexão full ports:

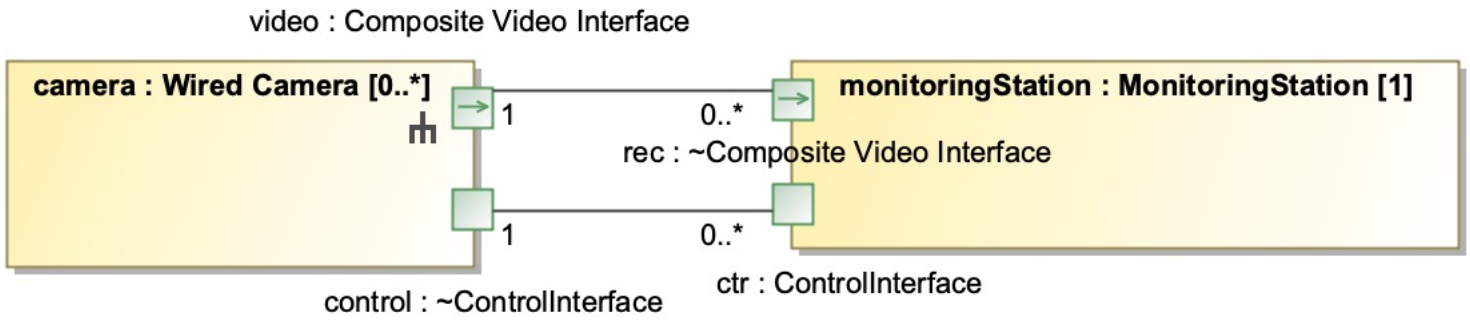
- Similar a conexão entre partes
- Regras de flow properties também se aplicam



Modelando Portas

Portas conjugadas (somente proxy ports):

- mecanismo para reusar um único InterfaceBlock para duas portas que se comunicam em direções opostas.
- Uma porta é a conjugada da outra
- Indicadas colocando um '~' no tipo da porta.

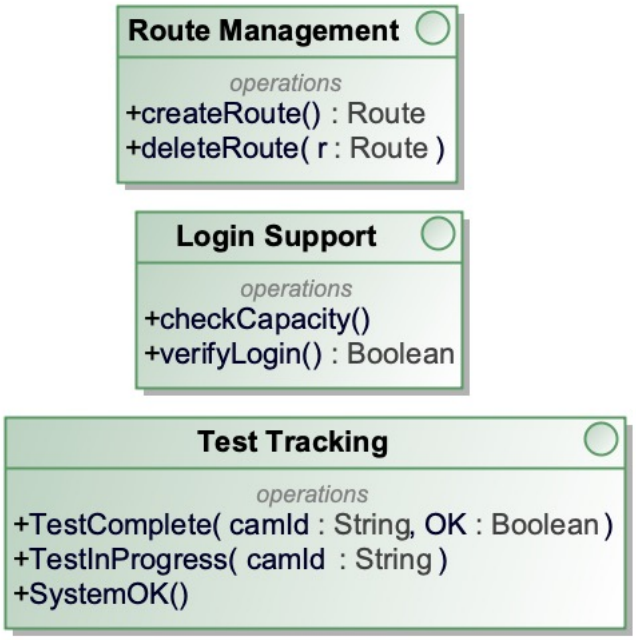


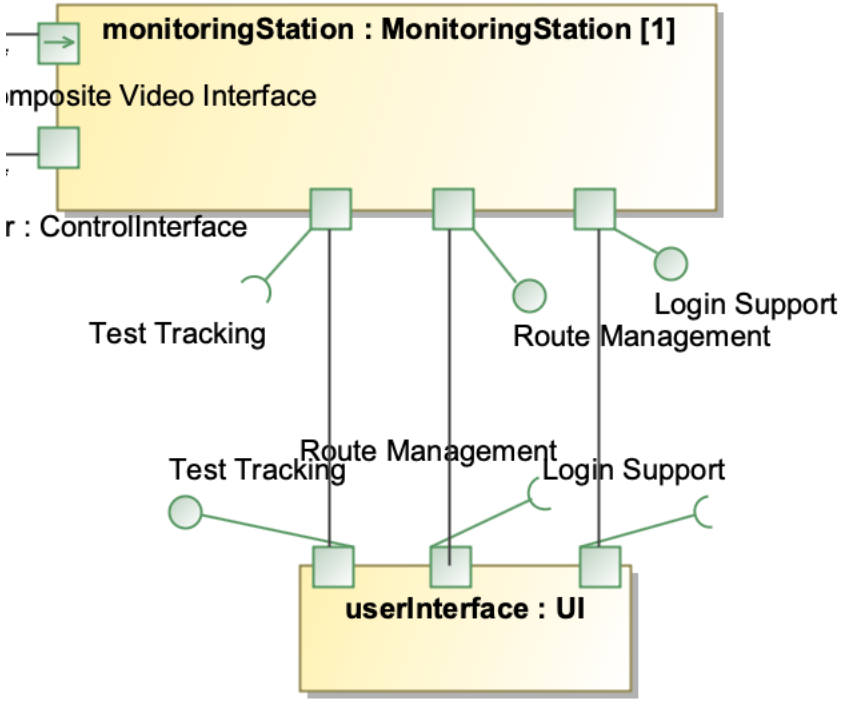
OBS: Multiplicidades devem ser compatíveis

Exercício

- Altere o componente veículo para utilizar portas ao invés de conexões diretas entre o mesmo e os elementos externos.
- Altere o modelo do Veículo para contemplar um novo componente chamado PowerTrain, o qual deve ser composto pelo Motor, Transmissão e duas Rodas dianteiras. As outras duas rodas traseiras continuam sendo partes do veículo. Crie portas e conecte os elementos internos do veículo, considere as ligações com as portas de fronteira do mesmo. Lembre-se de criar elementos para tipar as portas (Blocos no caso de full ports ou InterfaceBlocks no caso de proxy).

Definindo serviços através de portas





Exercício

- Refine o modelo dos blocos Multimedia, GPS, Bluetooth em termos de serviços expostos em portas. Assuma que Multimedia consome os serviços fornecidos por GPS e Bluetooth. Pense em outros serviços e possíveis interações entre Multimedia e outros blocos, e modele ao menos uma nova comunicação. Abaixo segue as assinaturas de operações de GPS e Multimedia já solicitadas.
  - GPS
    - retornaPosicao() : Posicao
    - definaRota(Posicao atual, Posicao destino): Rota
    - enviaPosicao(Posicao atual): void
  - Multimedia
    - connectarBluetooth(Dispostivo d)
    - buscarDispositivosBluetooth(): Dispositivo[0..*]
    - atenderChamadaTelefonica(): void

Centro de Informática UFPE

Universidade Federal de Pernambuco

Modelando estrutura com Blocos em SysML

Curso Ford

Prof. Lucas Albertins