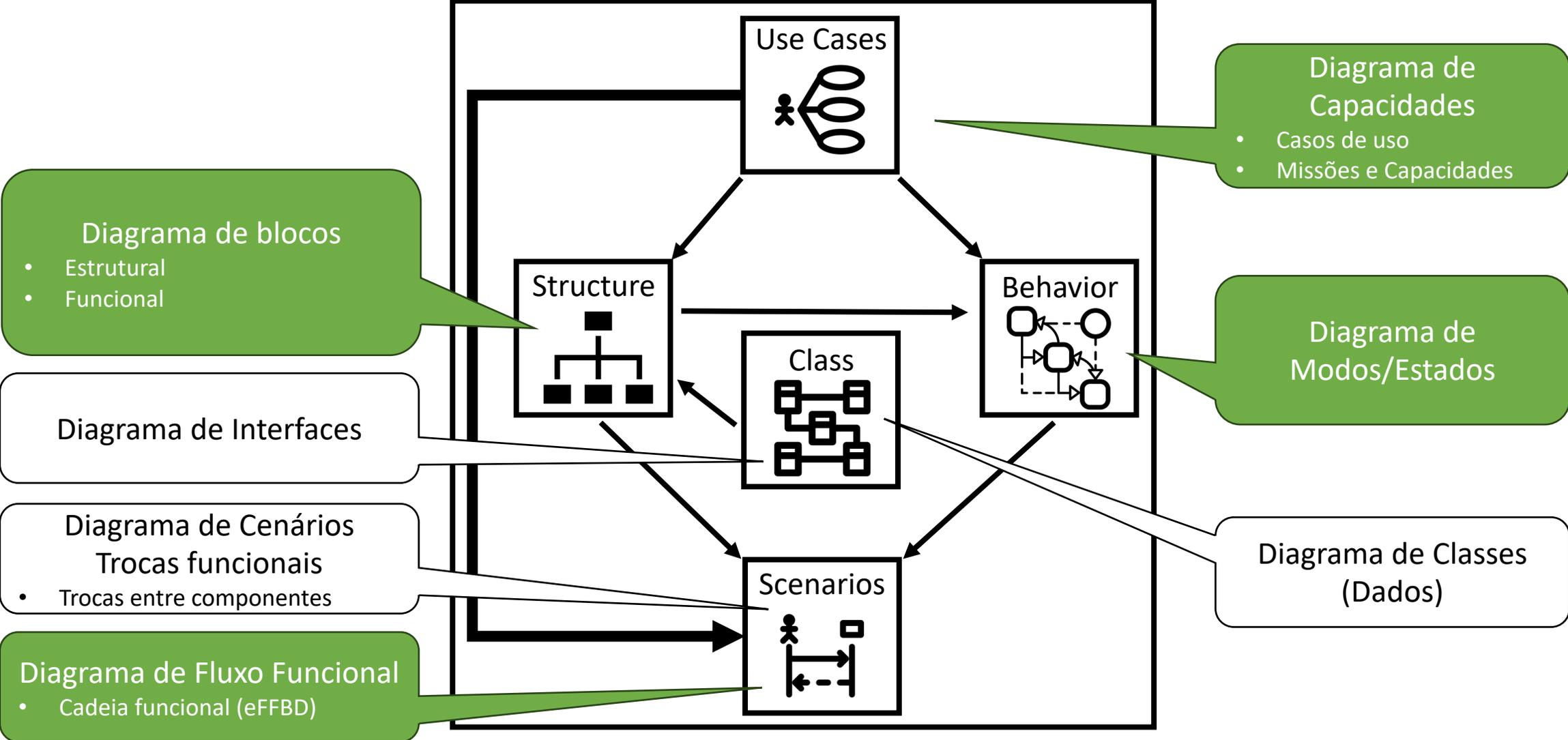




# ROTEIRO DOS DIAGRAMAS





IEA-P – DEPARTAMENTO DE PROJETOS  
(PROJECT DEPARTMENT)

# Diagramas de Classe (Blocos de Dados) e de Sequencia

[2024]

Prof. Dr. Christopher S. Cerqueira



# DIAGRAMA DE SEQUÊNCIA

MODELING MESSAGE-BASED BEHAVIOR WITH INTERACTIONS – CHAPTER 10

<https://developer.ibm.com/articles/the-sequence-diagram/>



# INTRODUÇÃO AO DIAGRAMA DE SEQUÊNCIA

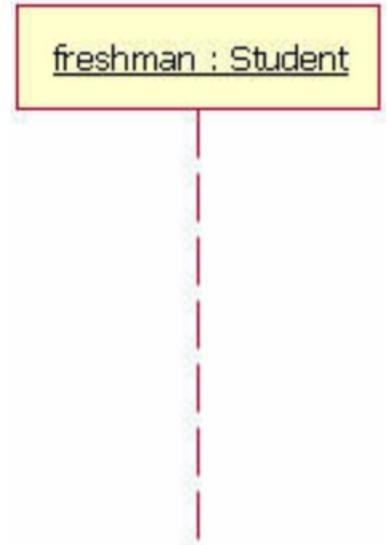
- Representar a **interação entre elementos em um modelo** como uma sequência de trocas de mensagens.
- 
- Um diagrama de sequência pode ser escrito como uma especificação de **como as partes de um sistema devem interagir** e também pode ser usado como um **registro de como as partes de um sistema interagem**.





# INTRODUÇÃO AO DIAGRAMA DE SEQUÊNCIA

- Os elementos estruturais são representados por **linhas do tempo** indicando as sequências de mensagens.
- 
- O diagrama de sequência descreve a **interação entre essas linhas do tempo** como uma série ordenada de ocorrência que descrevem diferentes tipos de eventos, como:
  - o **envio e recebimento** de mensagens,
  - a **criação e destruição** de objetos ou
  - o **início e o fim** de execuções de comportamento.





# EXEMPLO

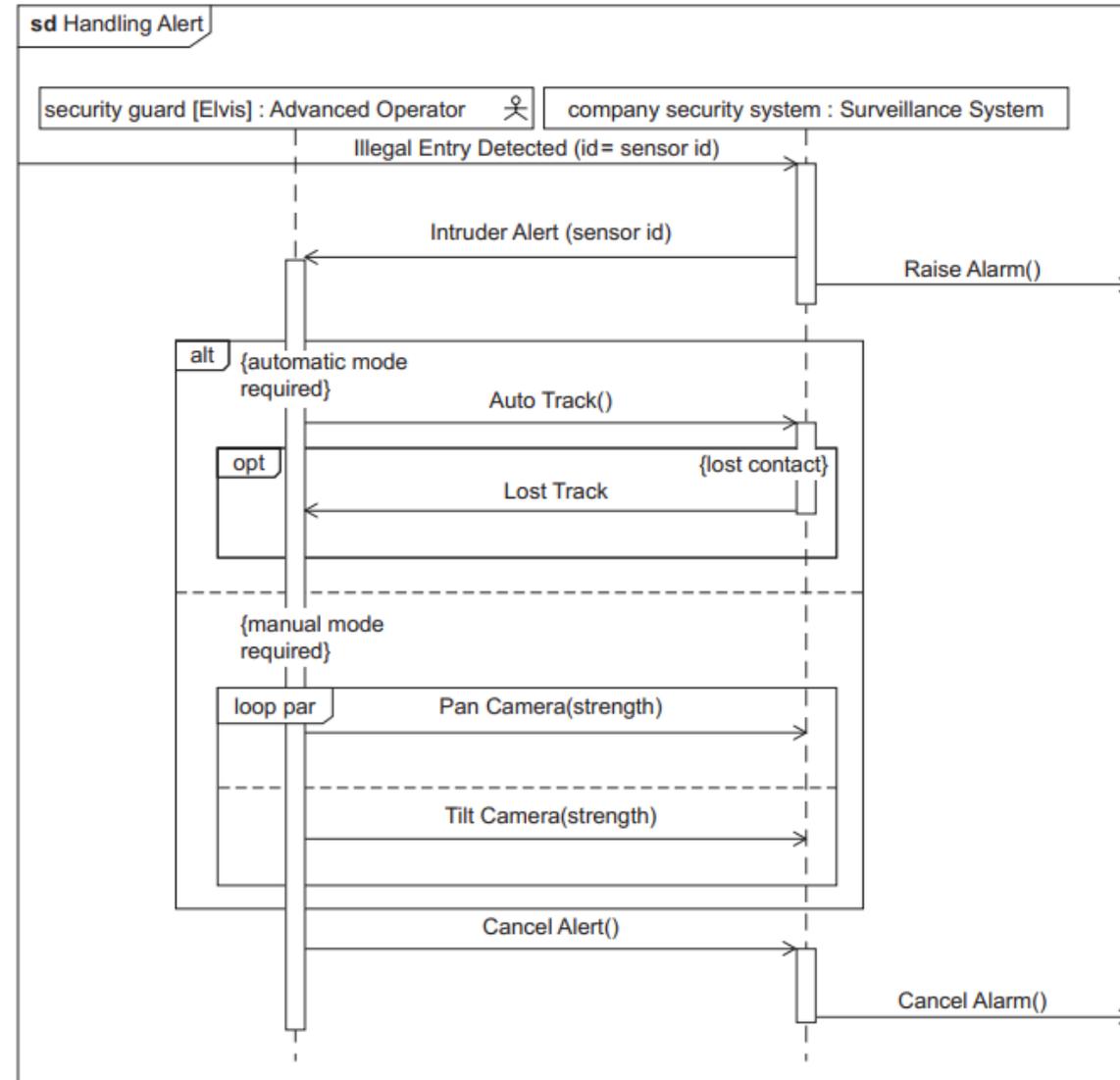


FIGURE 10.1

An example sequence diagram.



# LINHAS DO TEMPO

- A principal característica estrutural de uma interação é a **linha do tempo**. Uma linha do tempo representa o **tempo de vida relevante de uma entidade**.



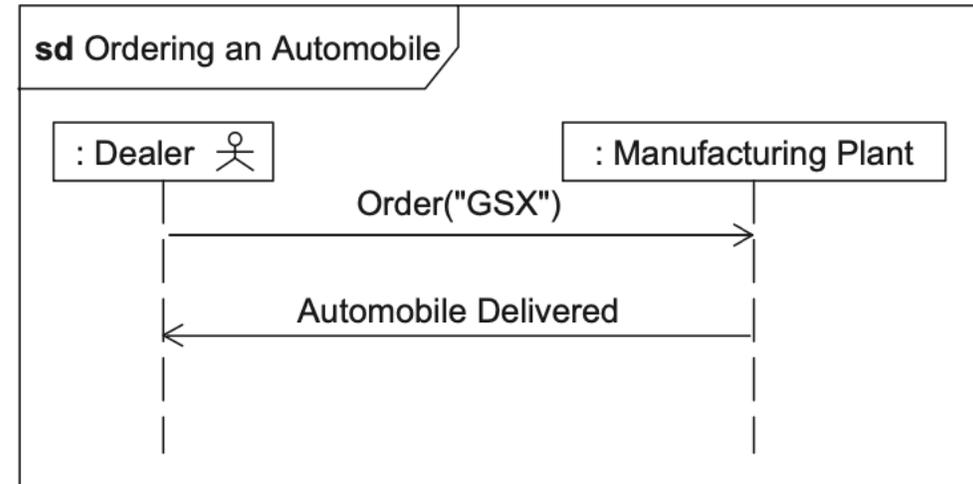
**FIGURE 10.3**

An interaction with lifelines.



# DESCRIÇÃO DE SEQUENCIA DE EVENTOS

- Quando uma interação é executada, o **conjunto de ocorrências ordenadas no tempo é chamado de rastro (trace)**. Uma comparação da ordem e estrutura das ocorrências reais determina se o rastro é consistente com a interação.
- Mensagens podem ser trocadas entre as entidades. Uma mensagem pode ser enviada de uma linha do tempo para si mesma para representar uma mensagem que é enviada e recebida pela mesma entidade.





# MENSAGENS

- A entidade que envia **mensagens assíncronas** continuam a executar depois que a mensagem é enviada, no entanto se é enviada uma **mensagem síncrona** ele deve aguardar a resposta da execução.
  - Uma ponta de seta aberta significa uma mensagem assíncrona.
  - Uma ponta de seta fechada significa uma mensagem síncrona.
  - Uma ponta de seta em uma linha tracejada mostra uma mensagem de resposta.

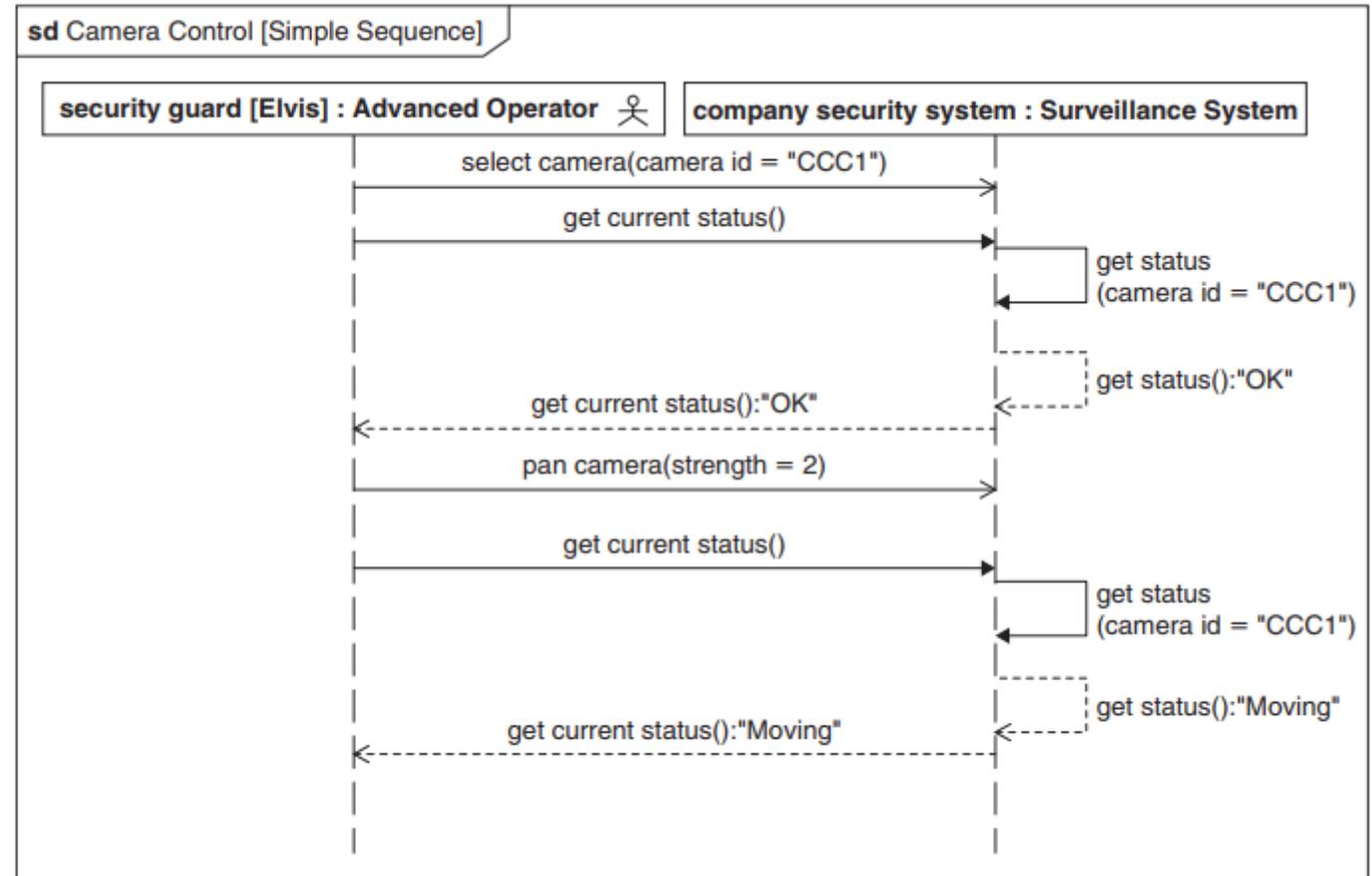


FIGURE 10.5

Synchronous and asynchronous messages exchanged between lifelines.



# EXECUÇÕES

- O recebimento de uma mensagem por uma linha do tempo pode **desencadear a execução de um comportamento**.
  - As ativações são símbolos retangulares sobrepostos verticalmente em linhas de vida.

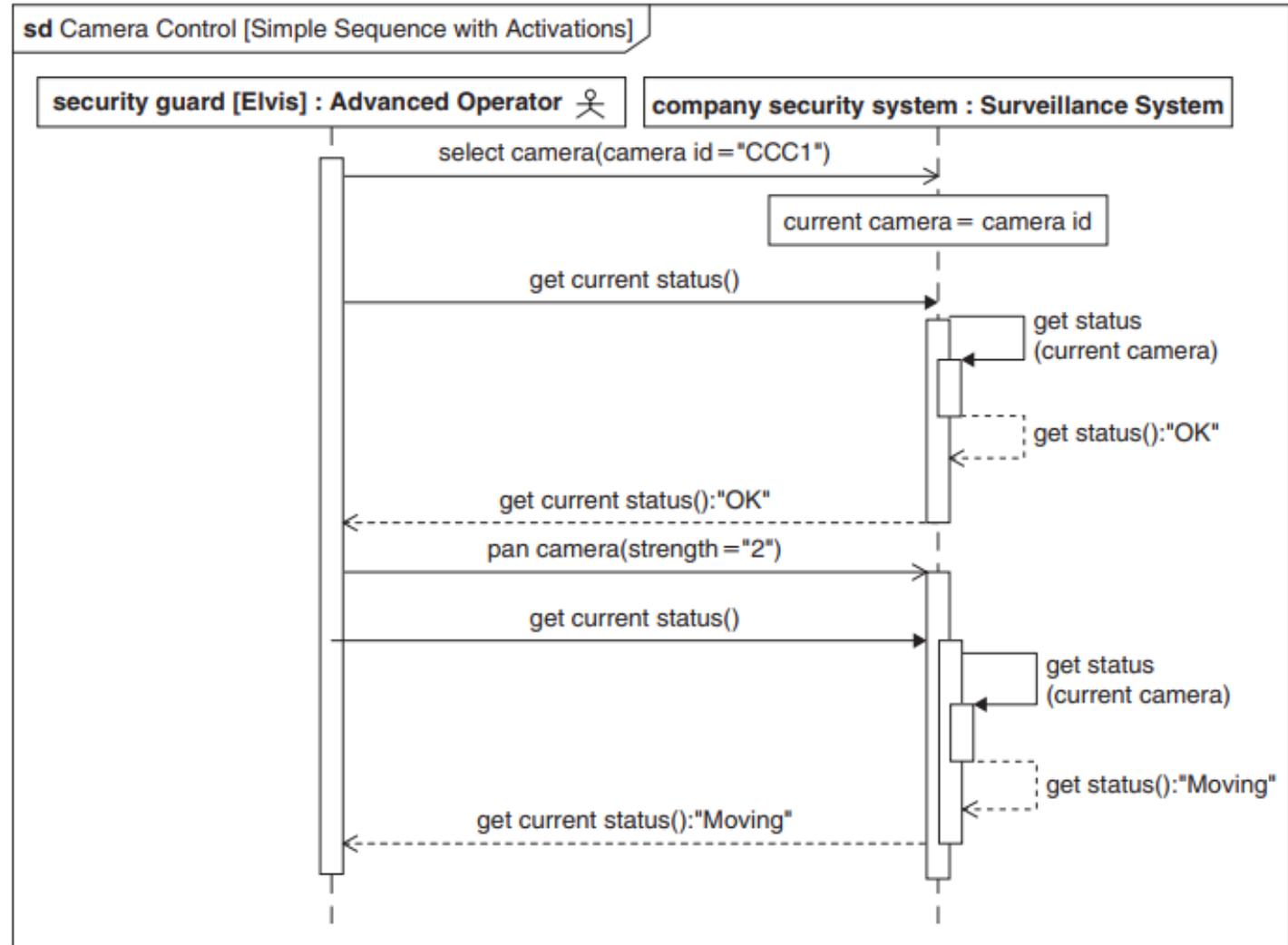


FIGURE 10.7

Lifelines with activations.



# CRIAÇÃO E EXCLUSÃO

- Uma **mensagem de criação** representa a criação de uma instância e, portanto, é a primeira ocorrência na linha do tempo.
- Uma **mensagem de exclusão** termina em um tipo especial de ocorrência chamada ocorrência de destruição, que deve ser a última ocorrência em uma linha do tempo.

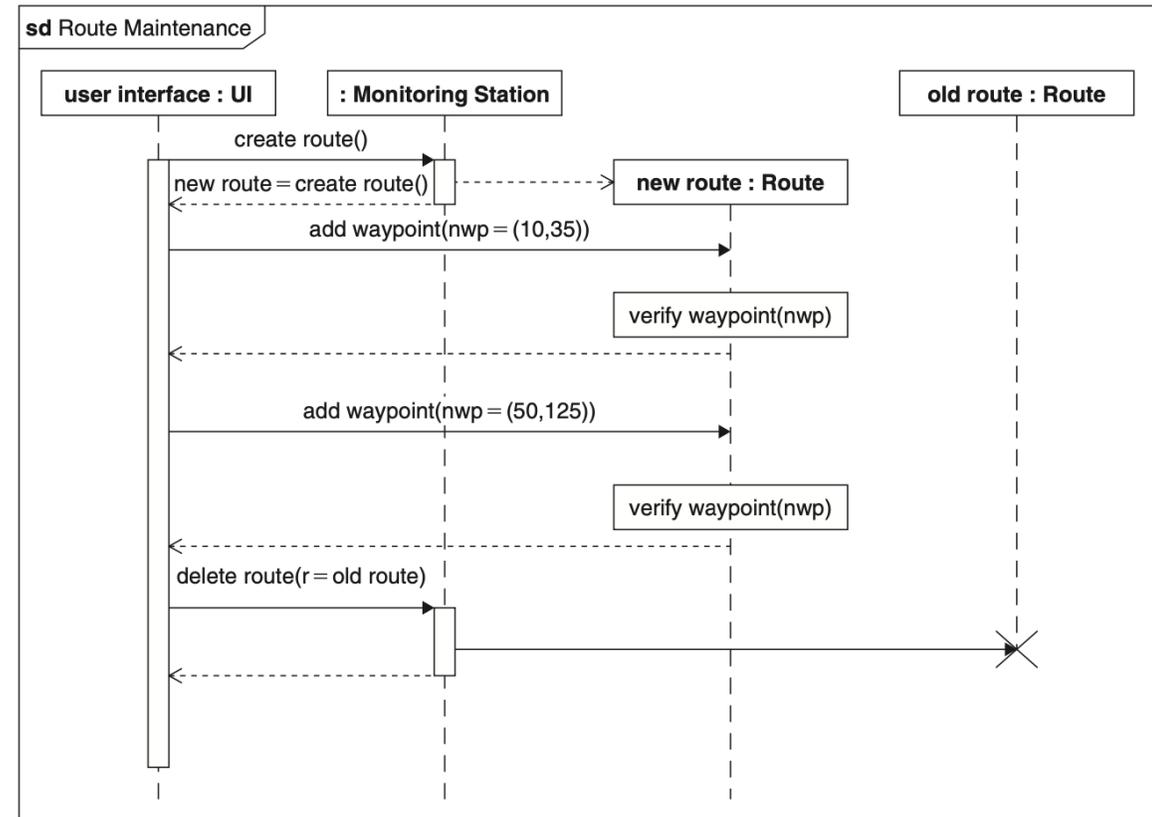


FIGURE 10.8

Create and destroy messages.



# REPRESENTANDO O TEMPO

- Uma **observação de tempo** refere-se a um instante no tempo correspondente à ocorrência de algum evento durante a execução da interação, e uma **observação de duração** refere-se ao tempo gasto entre dois instantes durante a execução da interação.
  - Uma restrição de tempo e uma restrição de duração podem usar observações para expressar restrições envolvendo os valores dessas observações.

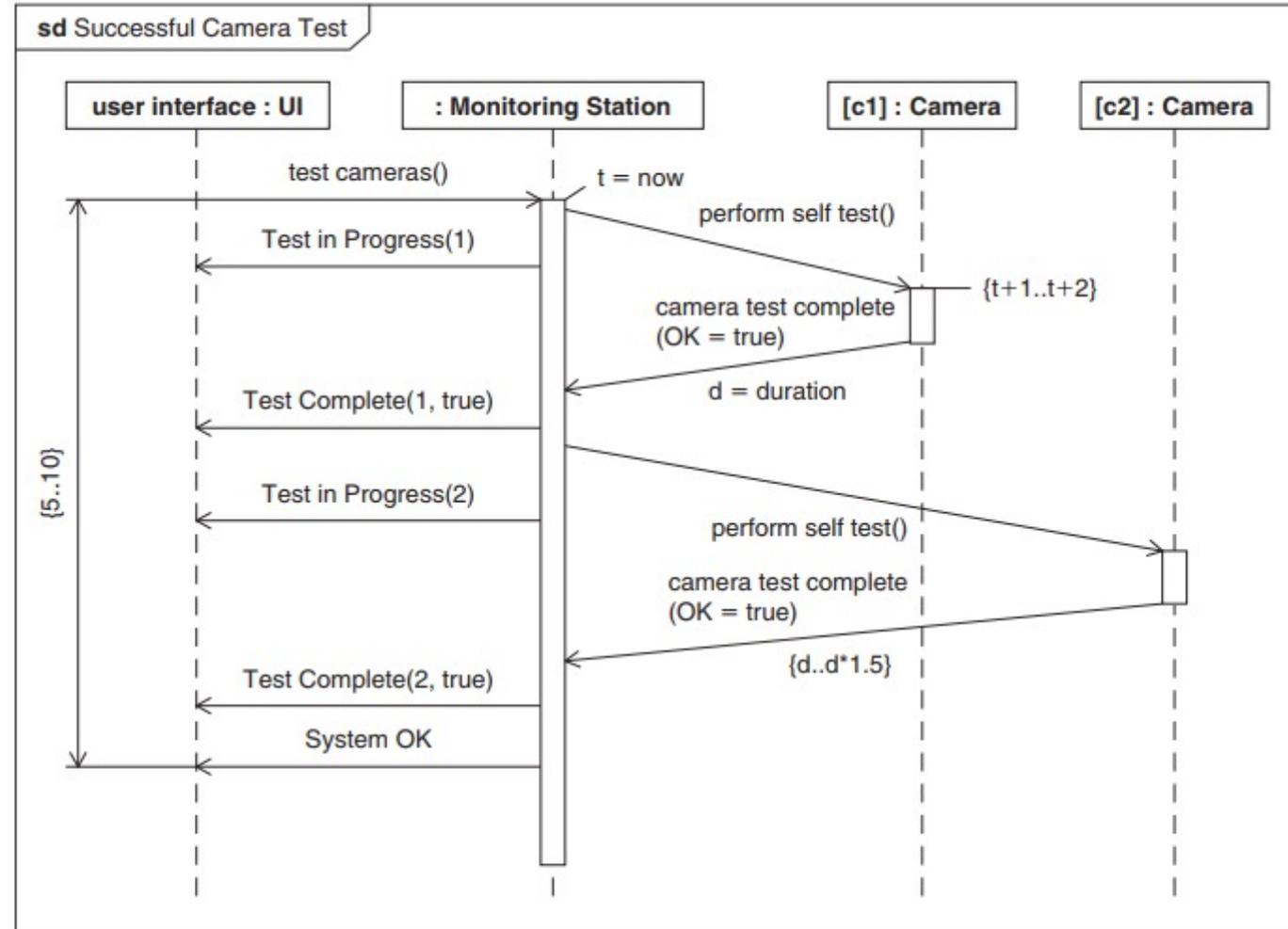


FIGURE 10.10

Representing time on a sequence diagram.

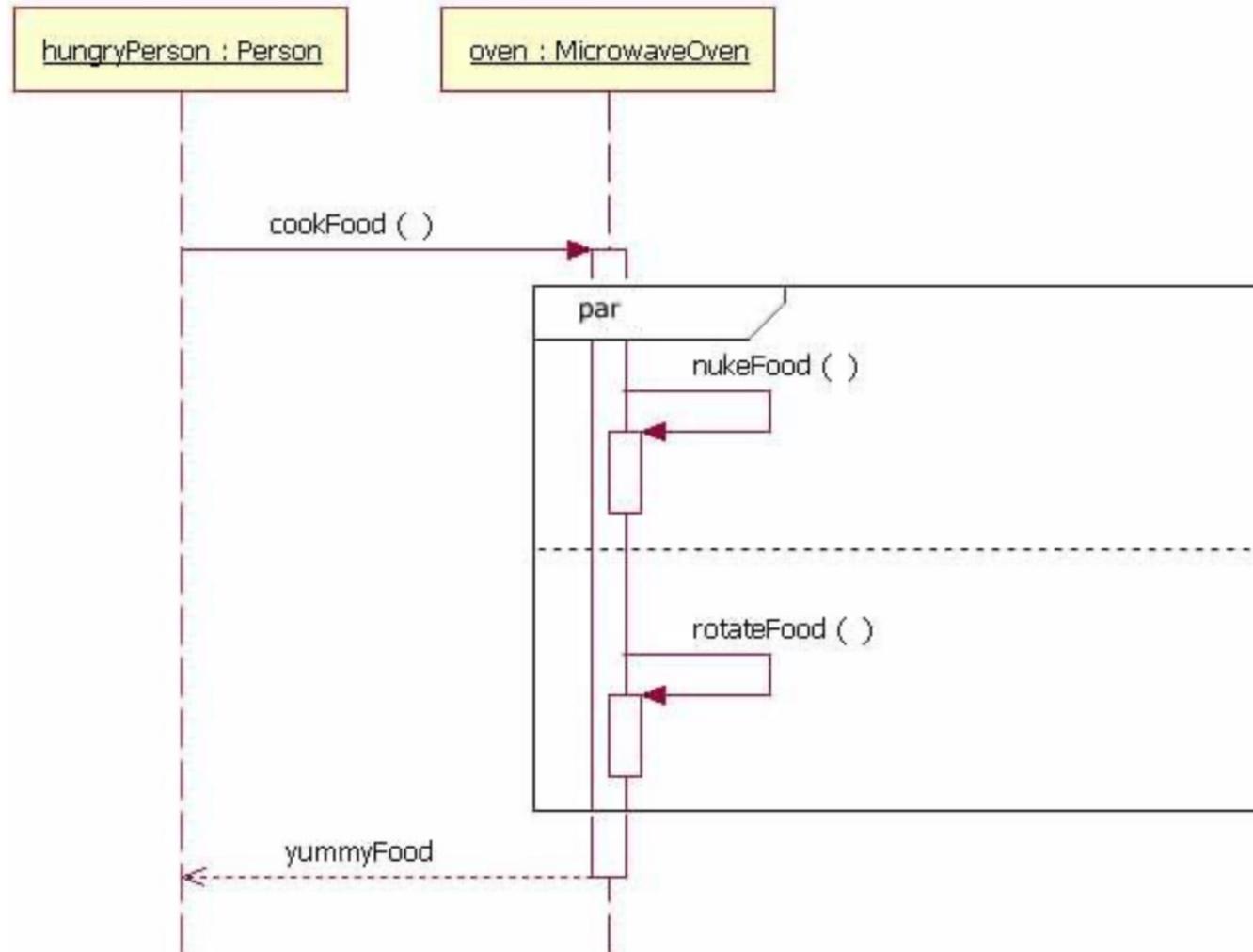


# FRAGMENTOS

- Padrões mais complexos de interação podem ser modelados usando construções chamadas fragmentos combinados.
  - **Seq** - O sequenciamento fraco é a forma padrão de sequenciamento para todos os operandos, portanto, raramente é indicado explicitamente.
  - **Par** - um operador no qual os operandos podem ocorrer em paralelo, cada um seguindo regras de sequenciamento fracas. Não há ordem implícita entre ocorrências em diferentes operandos.
  - **Alt/else** - um operador no qual exatamente um de seus operandos será selecionado com base no valor de sua proteção. O protetor (guarda) em cada operando é avaliado antes da seleção, e se o protetor em um dos operandos é válido, então esse é selecionado. Se mais de um operando tiver um protetor válido, a seleção não será determinística. Um fragmento else opcional é válido somente se nenhum dos protetores nos outros operandos for válido.
  - **Opt** - um operador unário que é equivalente a um alt com apenas um operando. Isso implica que o operando é executado ou ignorado, dependendo da validade do guarda..
  - **Loop** - um operador no qual o traço representado por seu operando se repete até que sua restrição de terminação seja atendida. Um loop pode definir limites inferiores e superiores no número de iterações, bem como a expressão de guarda.

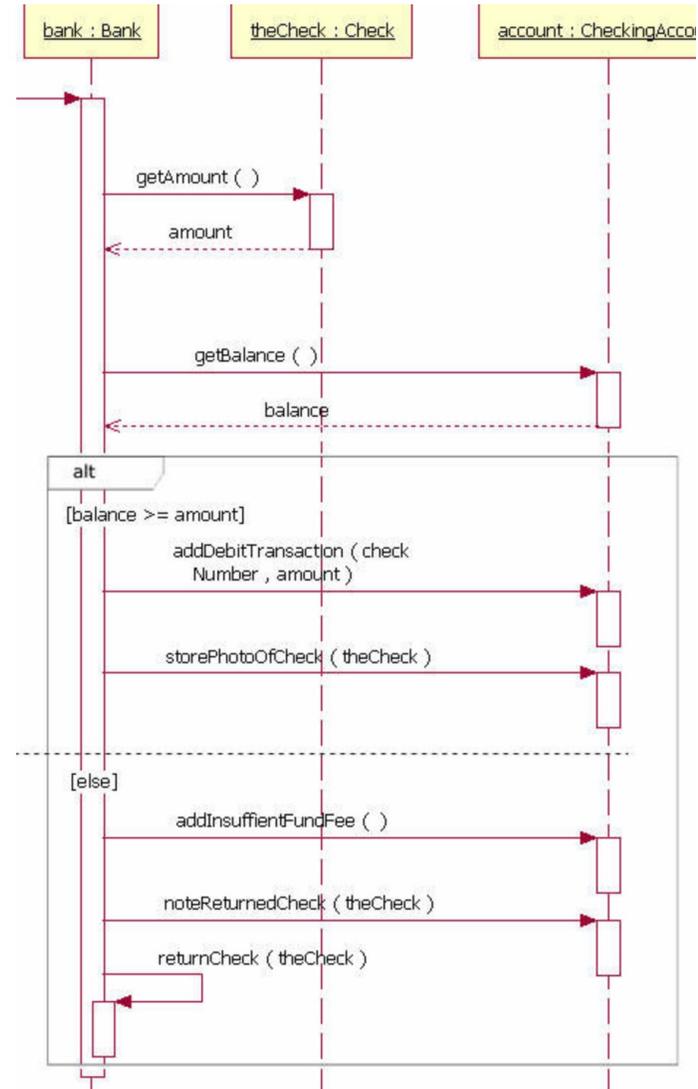


# LINHAS PARALELAS



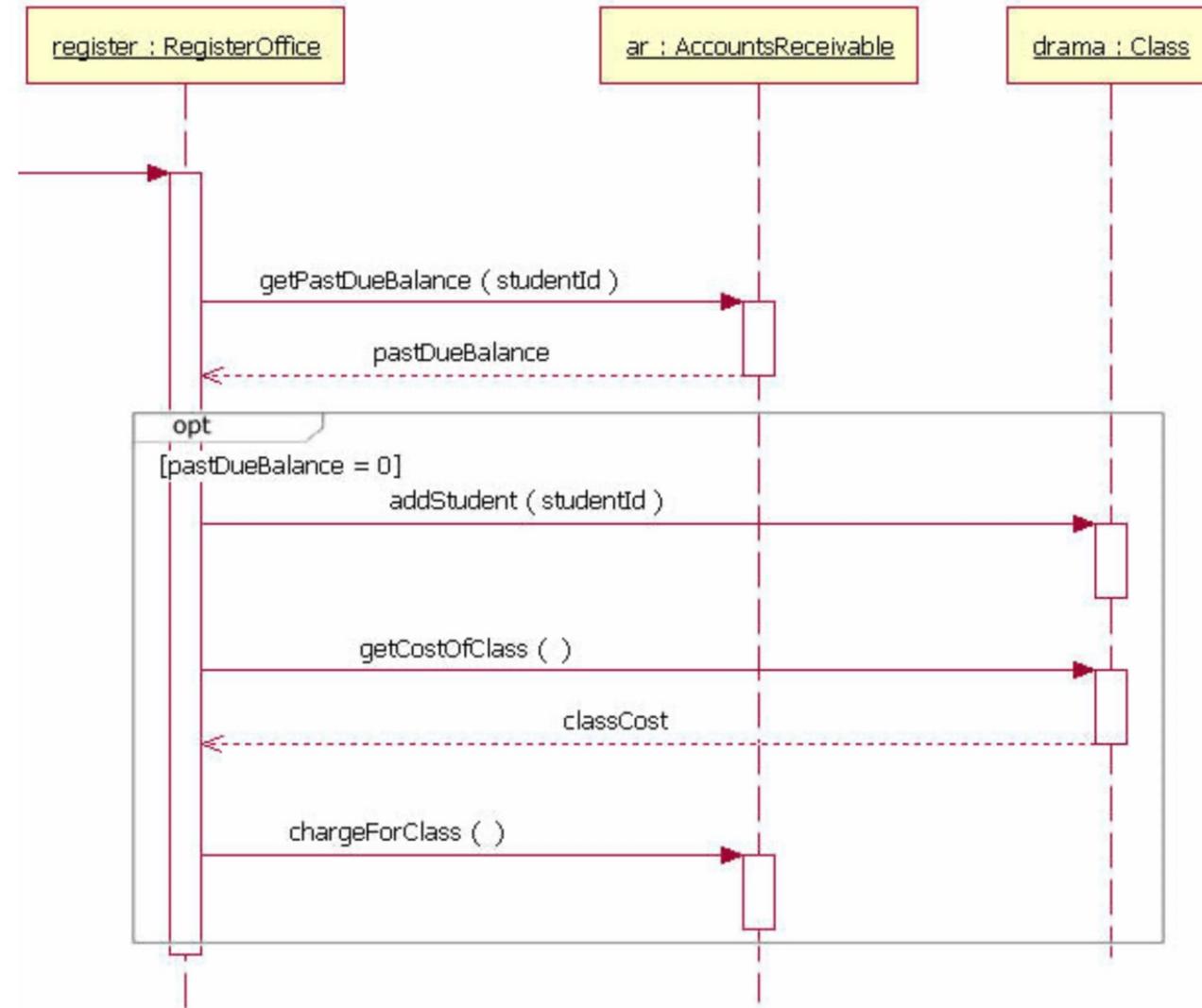


# LINHAS ALTERNATIVAS



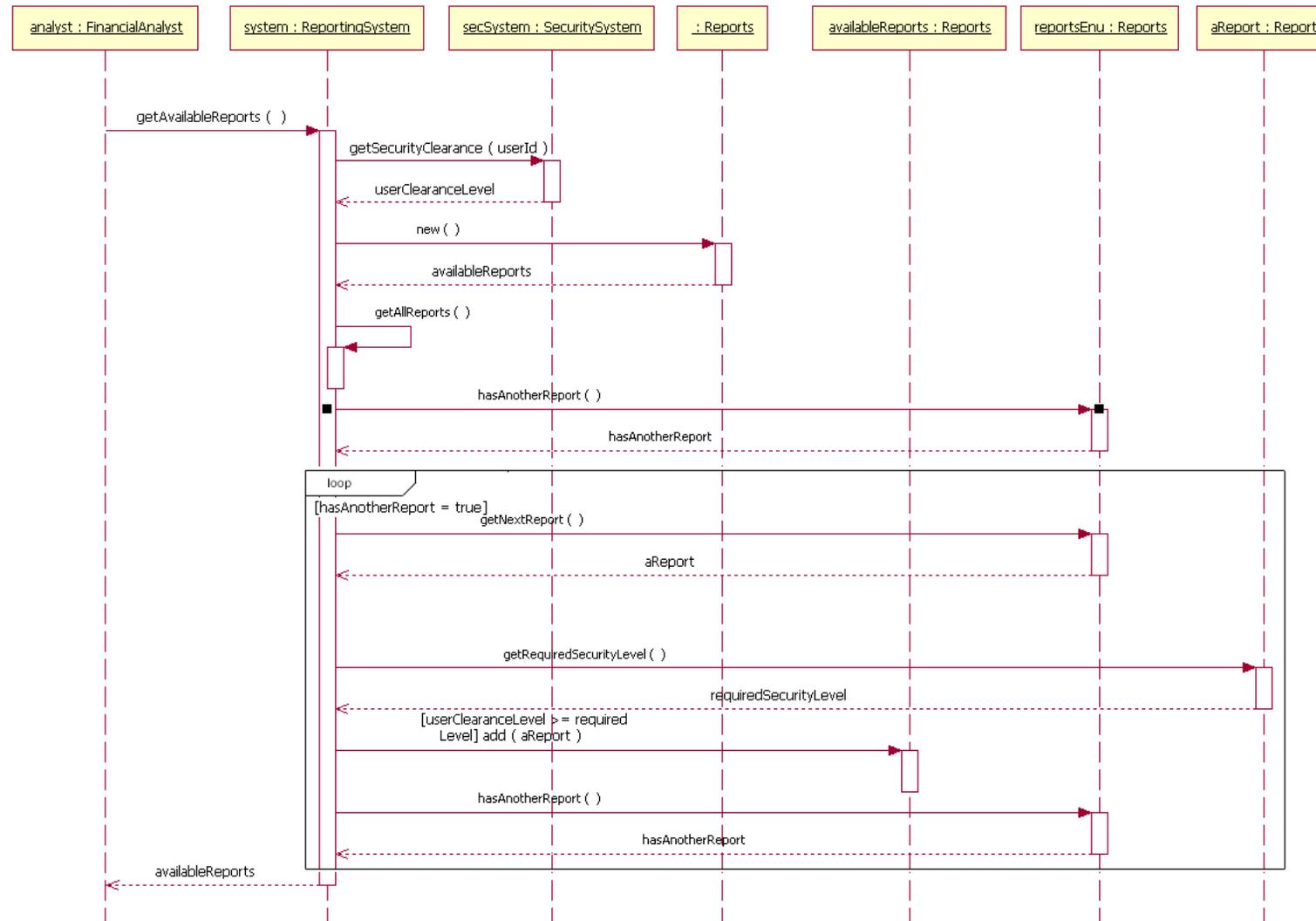


# LINHA OPCIONAL





# LOOPS





# INTERAÇÕES COMPLEXAS DESCRITAS USANDO OPERADORES FRAGMENT

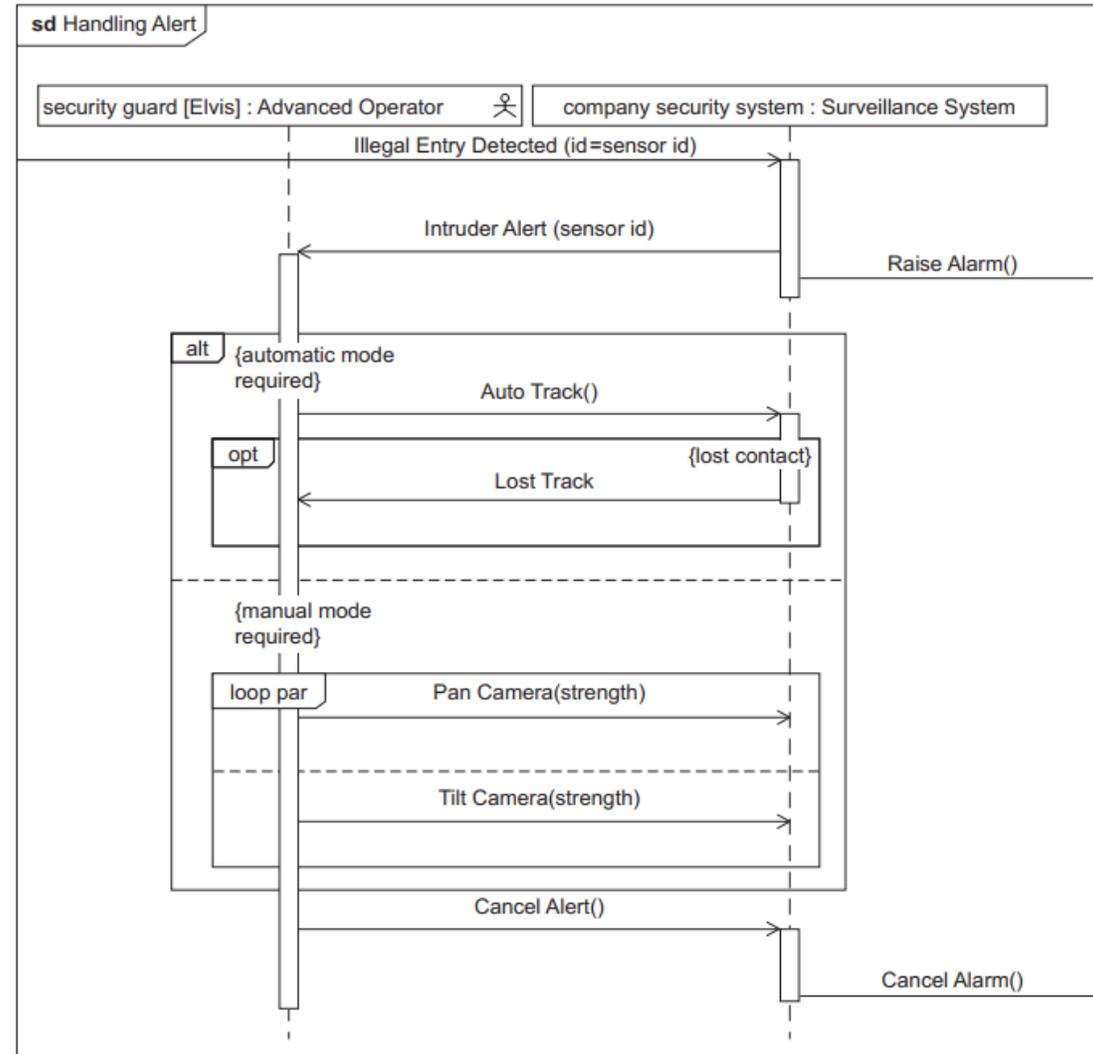


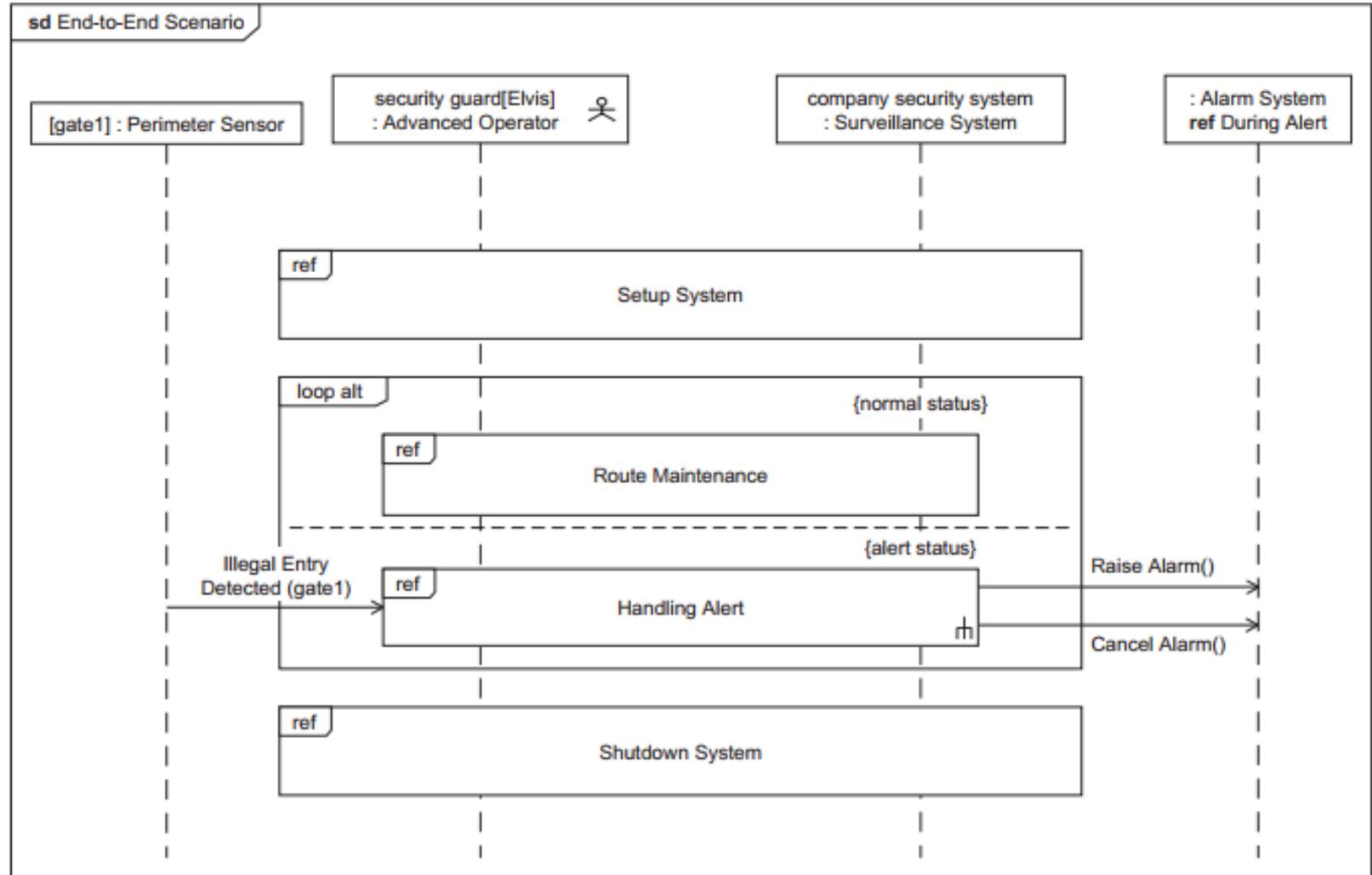
FIGURE 10.12

Complex interactions described using interaction operators.



# USANDO REFERÊNCIAS

- Para dar suporte a usos de interações em larga escala, uma interação pode incluir um uso de **interação que faça referência a uma interação descrita em outro diagrama de sequência**.
- Os usos de interação podem ser aninhados, porque uma interação referenciada pode, por sua vez, fazer referência a outra.
  - Esse recurso aumenta significativamente a escalabilidade das interações.
  - Também facilita a reutilização, uma vez que uma interação pode ser usada (ou seja, referenciada) por mais de um usando interação.

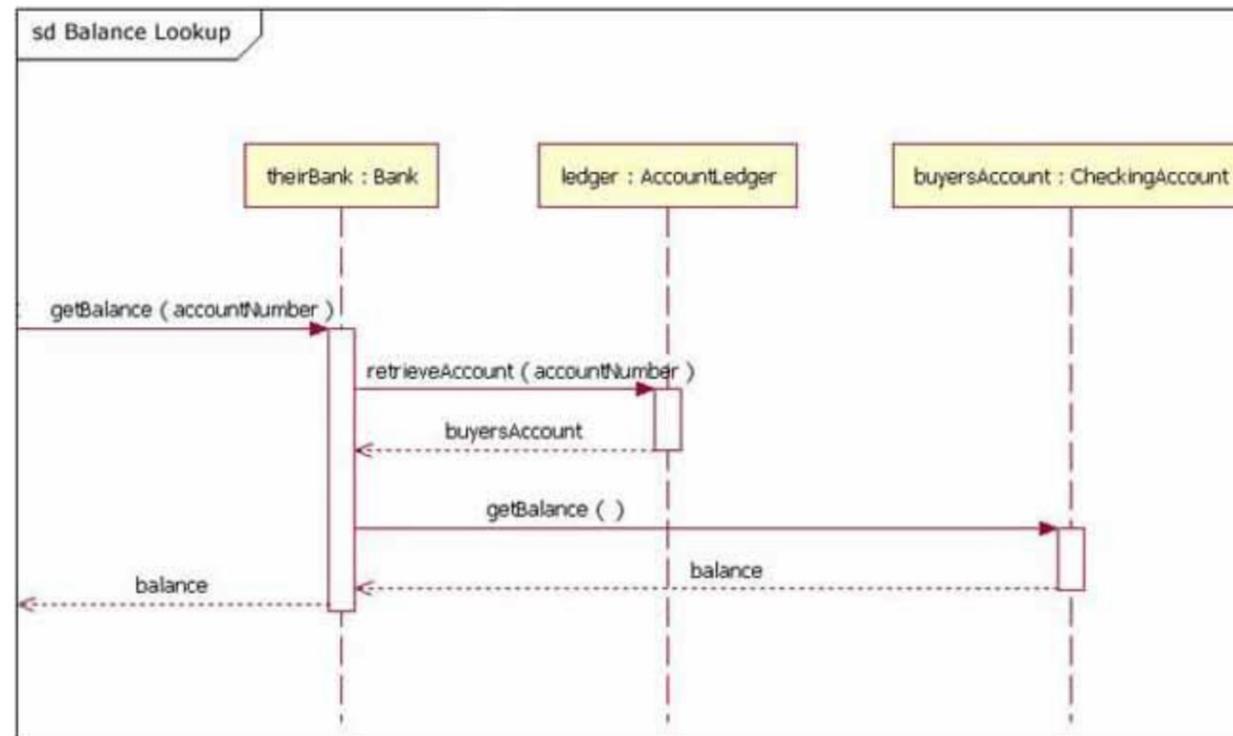
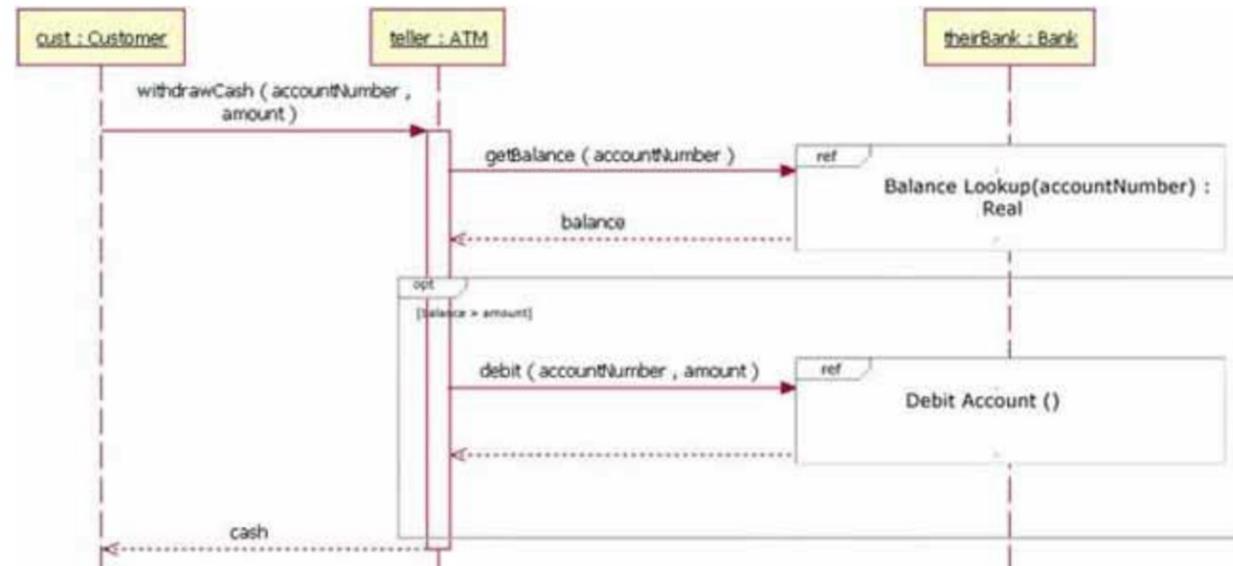


**FIGURE 10.15**  
Reference to another interaction.



# GATES

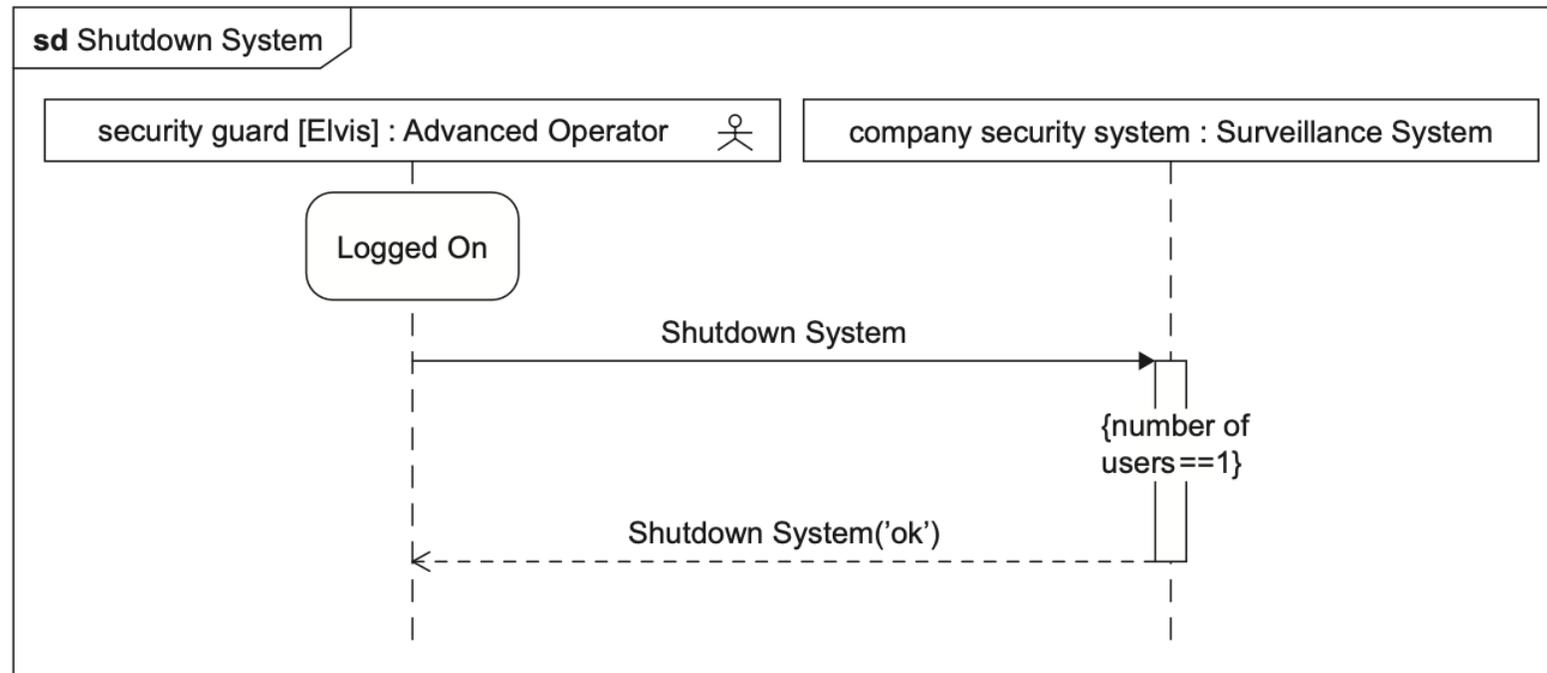
- Gates pode ser uma maneira fácil de **modelar a passagem de informações entre um diagrama de sequência e seu contexto.**
- Um gate é meramente uma mensagem que é ilustrada com uma extremidade conectada à borda do quadro do diagrama de sequência e a outra extremidade conectada a uma linha de vida.





# ESTADOS

- Muitas vezes é útil aumentar a representação adicionando restrições sobre o estado necessário de uma linha de vida em um determinado ponto em uma sequência de ocorrências. Isso pode ser alcançado usando um estado em uma linha do tempo.



**FIGURE 10.14**

State invariants.



Nosso “diagrama  
de Classes”

# DIAGRAMAS DE BLOCOS

MODELING STRUCTURE WITH BLOCKS – CHAPTER 7



# INTRODUÇÃO AO DIAGRAMA DE BLOCOS

- **O bloco é a unidade modular da estrutura no SysML** que é usado para definir um tipo de sistema, componente, interconexão de componente ou item que flui através do sistema, bem como entidades externas, entidades conceituais ou outras abstrações lógicas. Um bloco descreve um conjunto de instâncias que compartilham a definição do bloco. **Um bloco é definido por suas características, que podem ser subdivididas em características estruturais e características comportamentais.**
- O diagrama de definição de blocos (bdd) é usado para definir blocos em termos de suas características e suas relações estruturais com outros blocos:

*bdd [model element kind] model element name [diagram name]*

- O diagrama de blocos interno ou ibd se assemelha a um diagrama de blocos tradicional e mostra as conexões entre as partes de um bloco:

*ibd [block] block name [diagram name]*



# EXEMPLO DE DIAGRAMA DE DEFINIÇÃO DE BLOCO

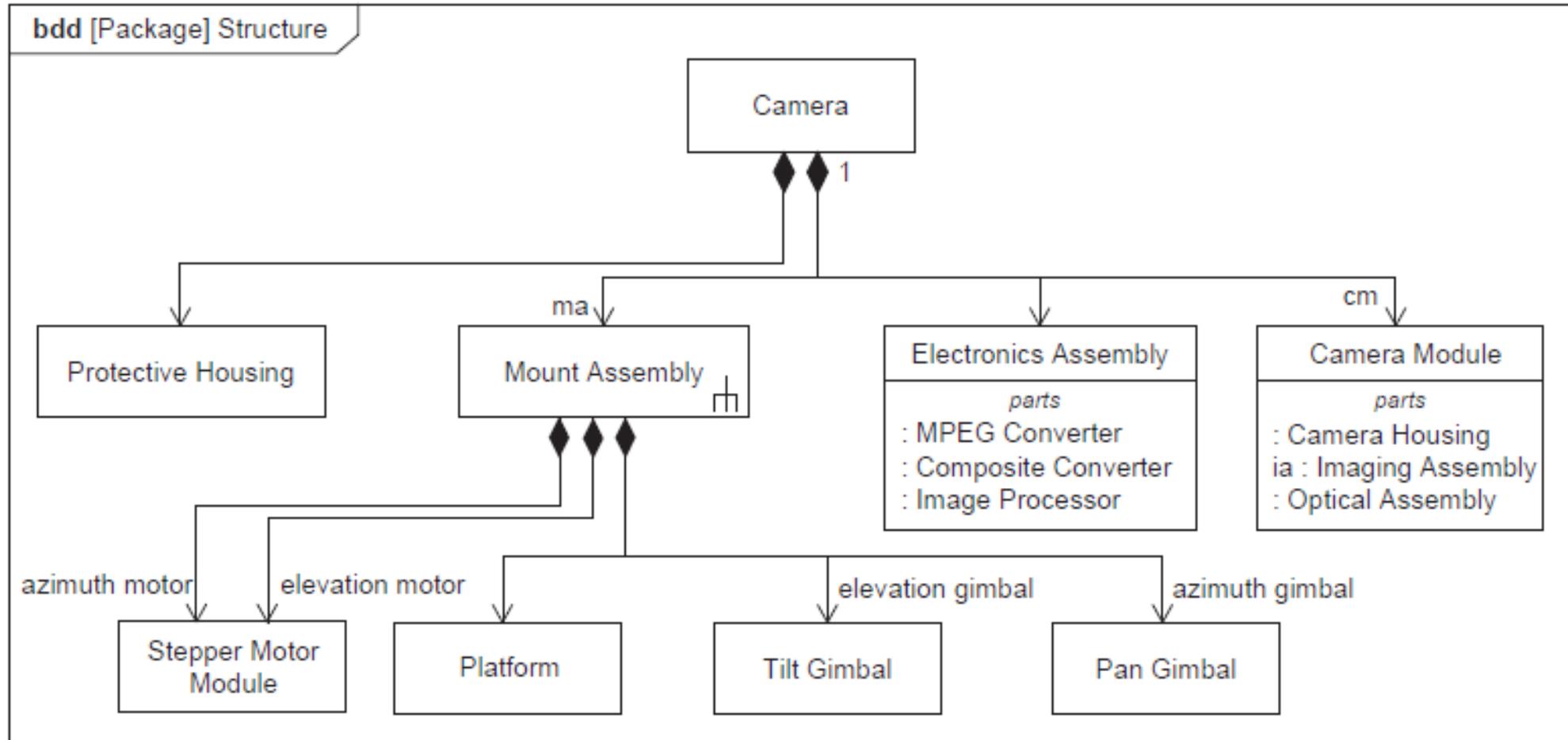
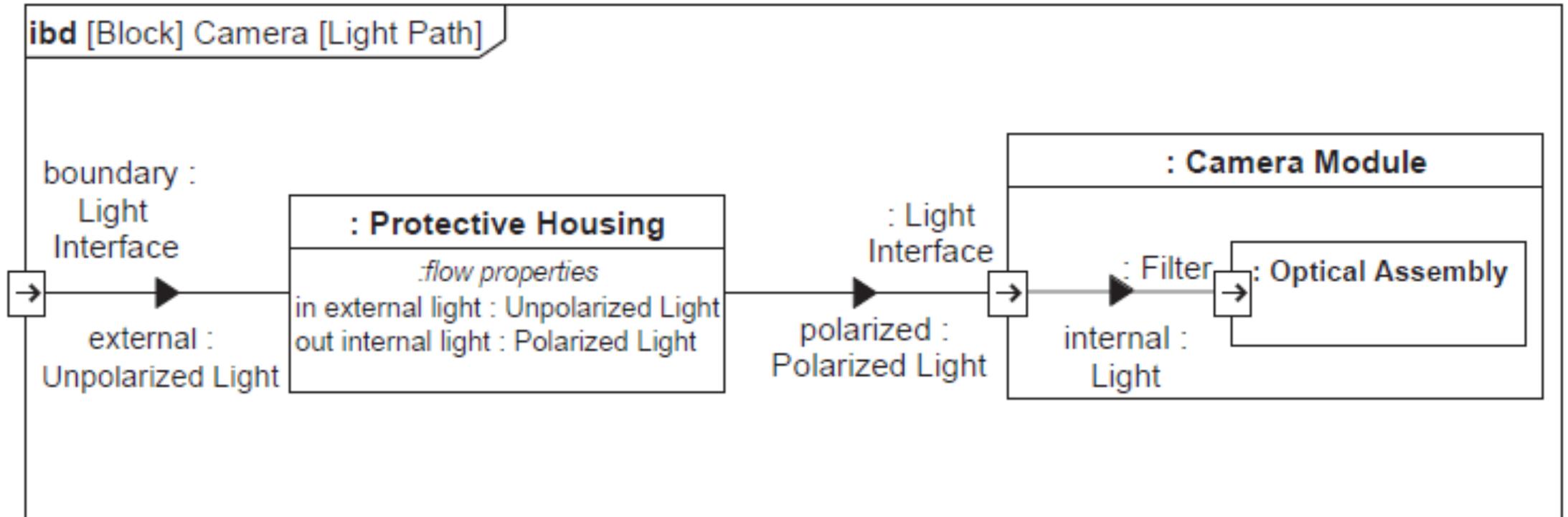


FIGURE 7.1

Example block definition diagram.



# EXEMPLO DE DIAGRAMA DE BLOCOS INTERNOS



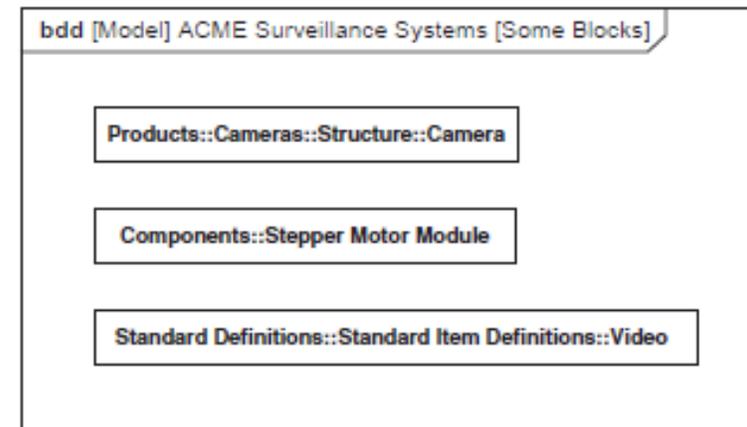
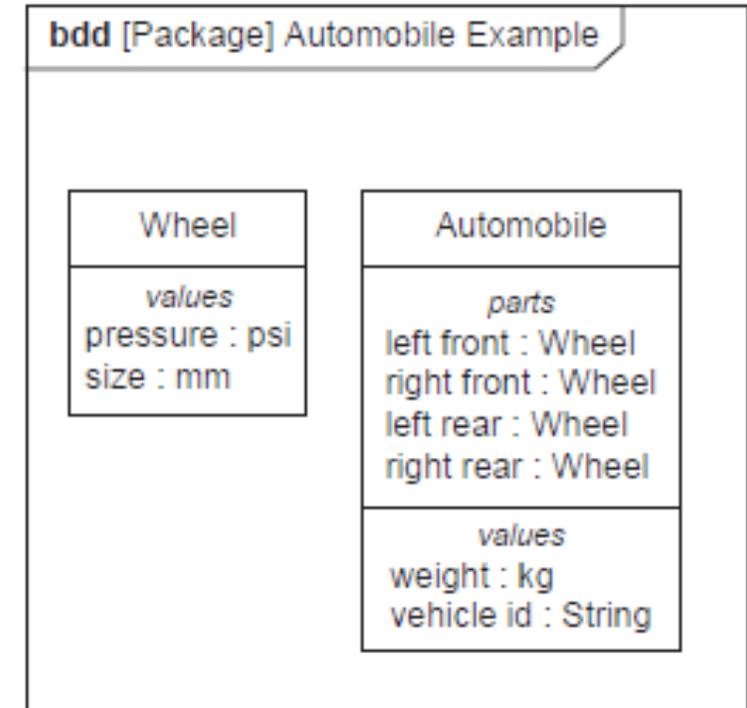
**FIGURE 7.2**

Example internal block diagram.



# BLOCOS

- O bloco é a **unidade modular fundamental** para descrever a estrutura do sistema no SysML.
- Outros tipos de recursos de bloco— *parts*, *operations*, *value properties*, and *ports* – podem ser apresentado em outros compartimentos do símbolo do bloco.



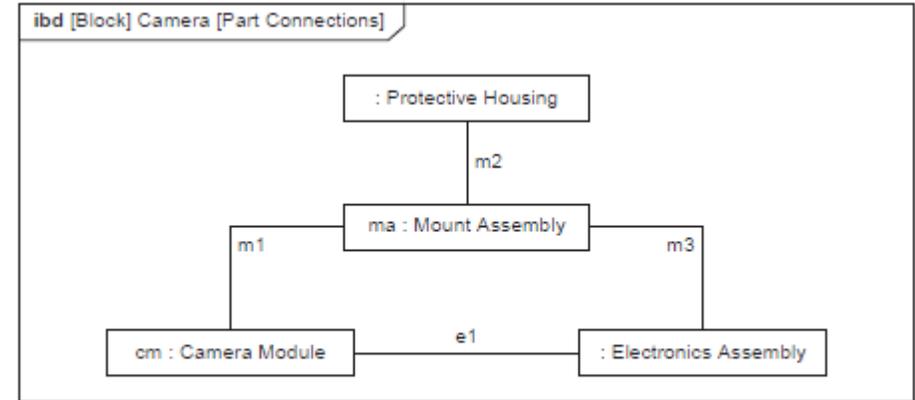
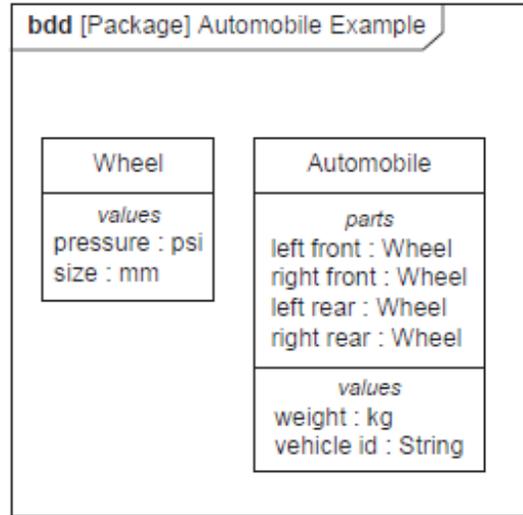


# PROPRIEDADES

- Propriedades são características estruturais de um bloco. Uma propriedade tem um tipo que define suas características, que pode ser outro bloco ou algum tipo mais básico, como um “inteiro”, “float”, “booleano” ou “string”.
  - ***Part properties*** (*parts for short*) descreve a decomposição de um bloco em seus elementos constituintes.
  - ***Reference properties*** cujos valores se referem a partes de outros blocos.
  - ***Value properties*** descreve as características quantificáveis de um bloco, como seu peso ou velocidade.

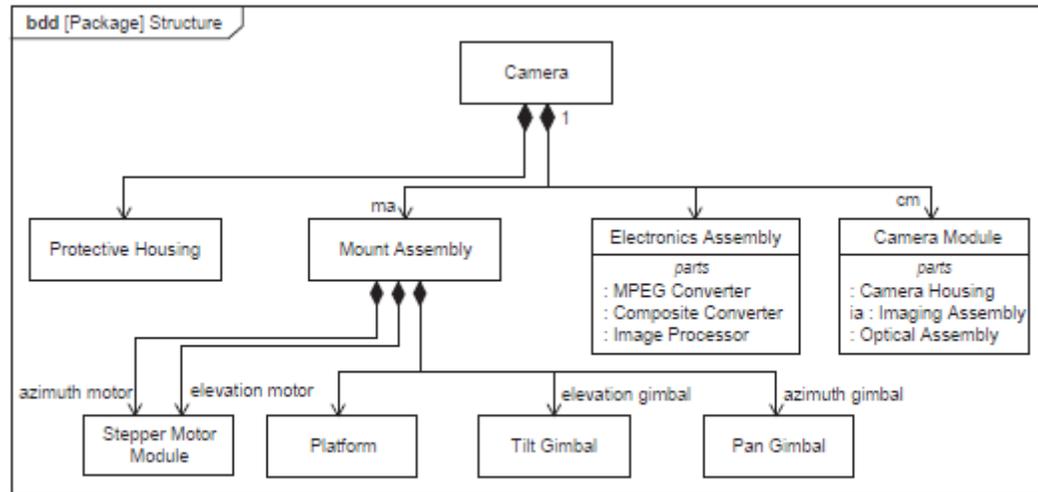


# PARTES



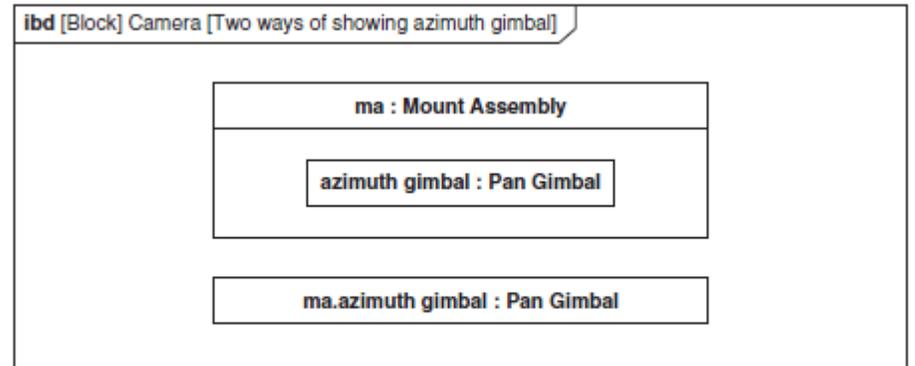
**FIGURE 7.7**

Connecting parts on an internal block diagram.



**FIGURE 7.5**

Showing a block composition hierarchy on a block definition diagram.

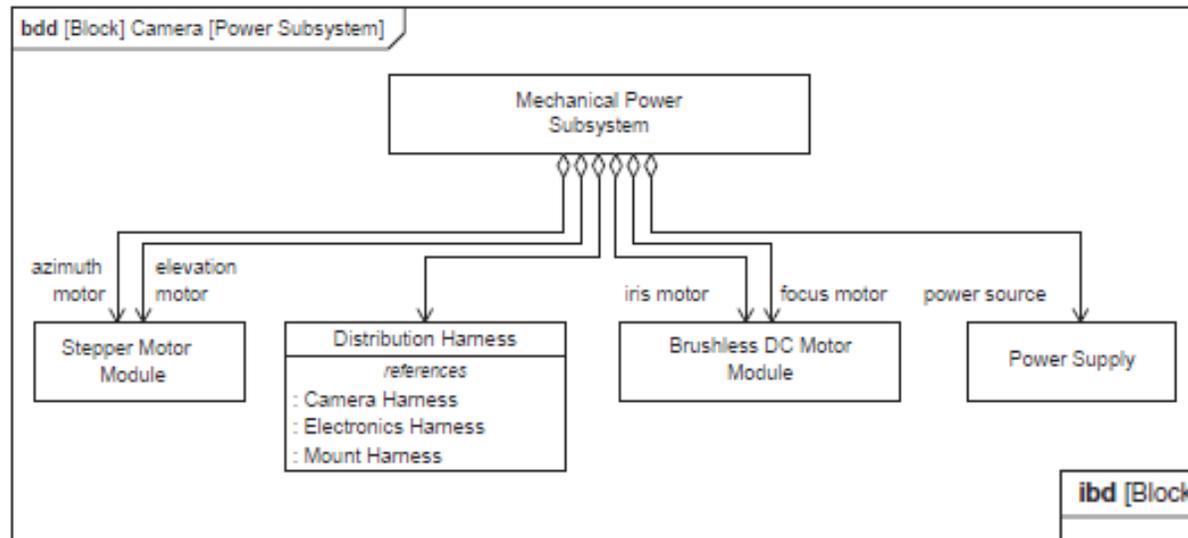


**FIGURE 7.8**

Showing deep-nested parts on an internal block diagram.

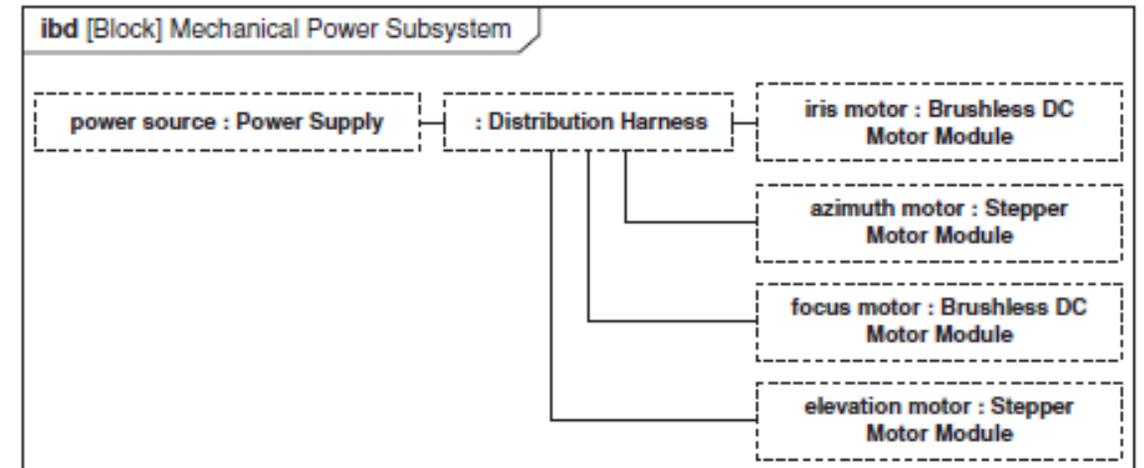


# REFERÊNCIAS



**FIGURE 7.10**

Reference associations on a block definition diagram.



**FIGURE 7.11**

Reference properties and their interconnections on an internal block diagram.



# VALORES

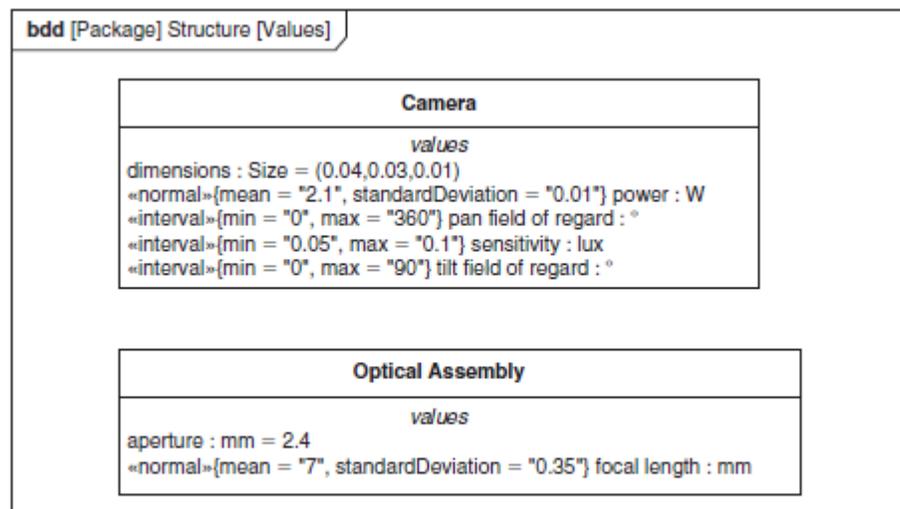


FIGURE 7.22

Examples of property values and distributions.

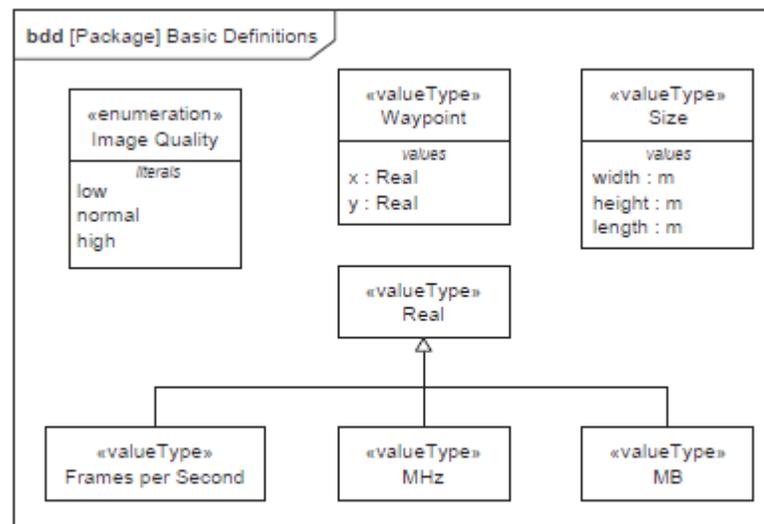


FIGURE 7.17

Definition of basic value types in a block definition diagram.

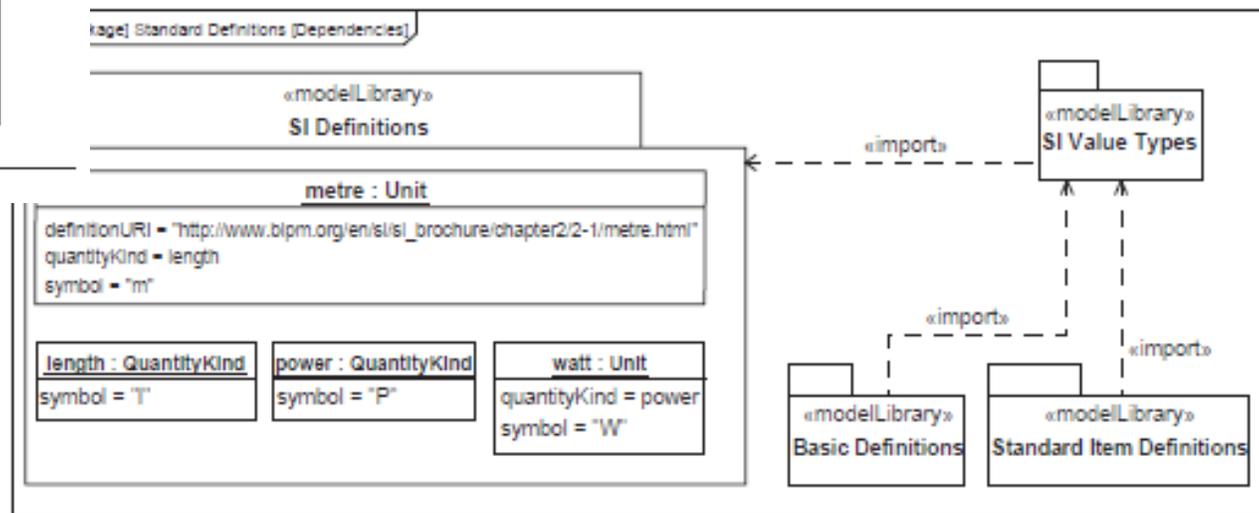


FIGURE 7.18

Importing the SI definitions defined by SysML.

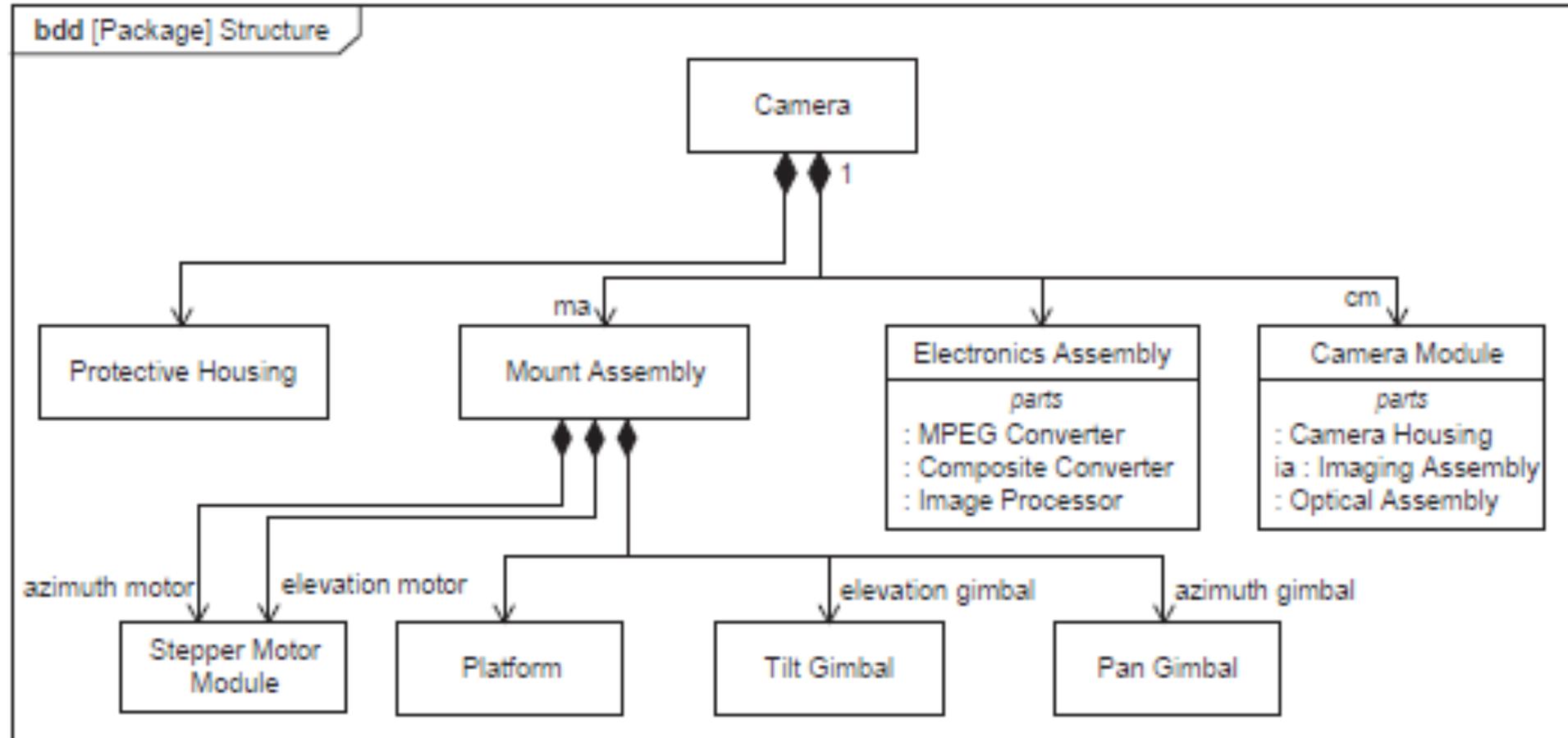


# RELAÇÕES ENTRE BLOCOS

- COMPOSITE ASSOCIATIONS - descrever relacionamentos entre partes e o todo.
- REFERENCE ASSOCIATIONS - descrever uma hierarquia lógica que faz referência a blocos.
- HERITAGE - descrever a especialização hierárquica dos elementos
  - Polymorfism – objetos diferentes, com a mesma raiz, que podem responder à mesma solicitação



# COMPOSITION

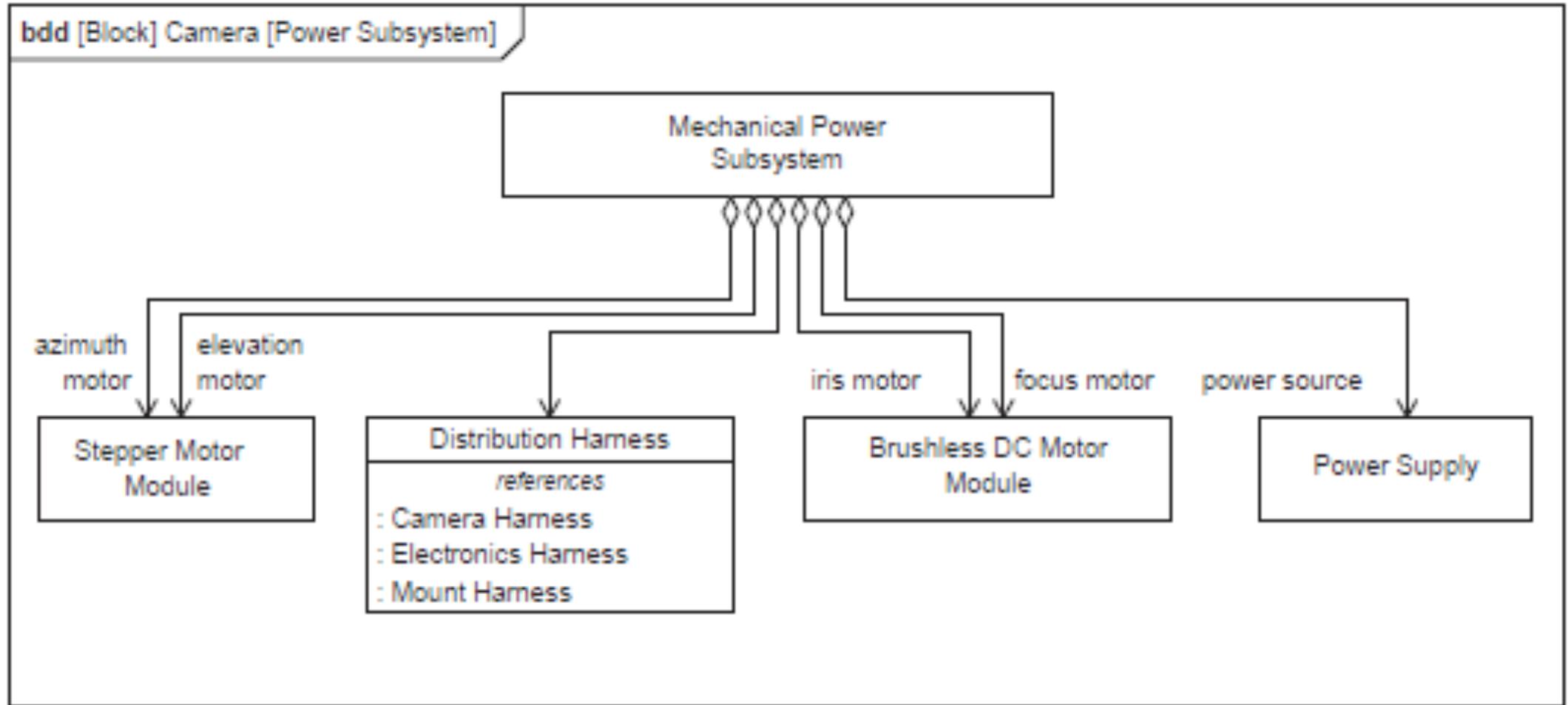


**FIGURE 7.5**

Showing a block composition hierarchy on a block definition diagram.



# REFERENCE

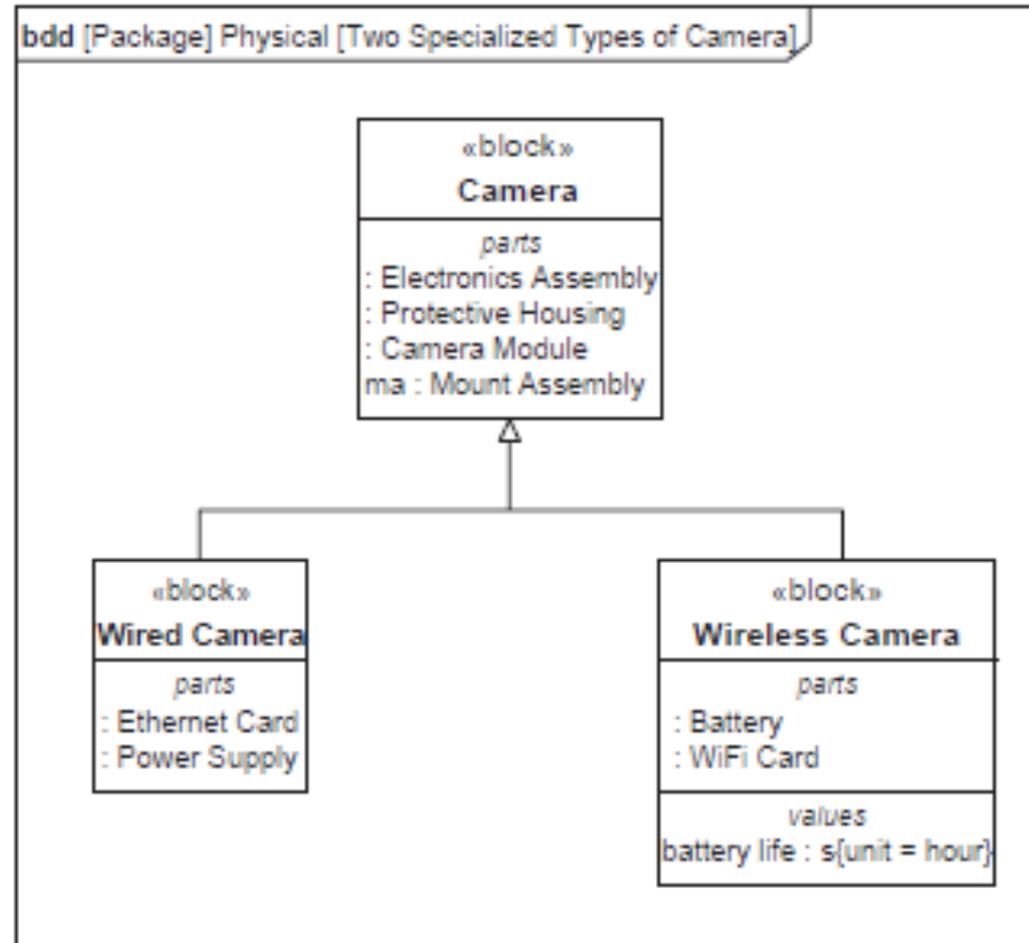


**FIGURE 7.10**

Reference associations on a block definition diagram.



# HERITAGE

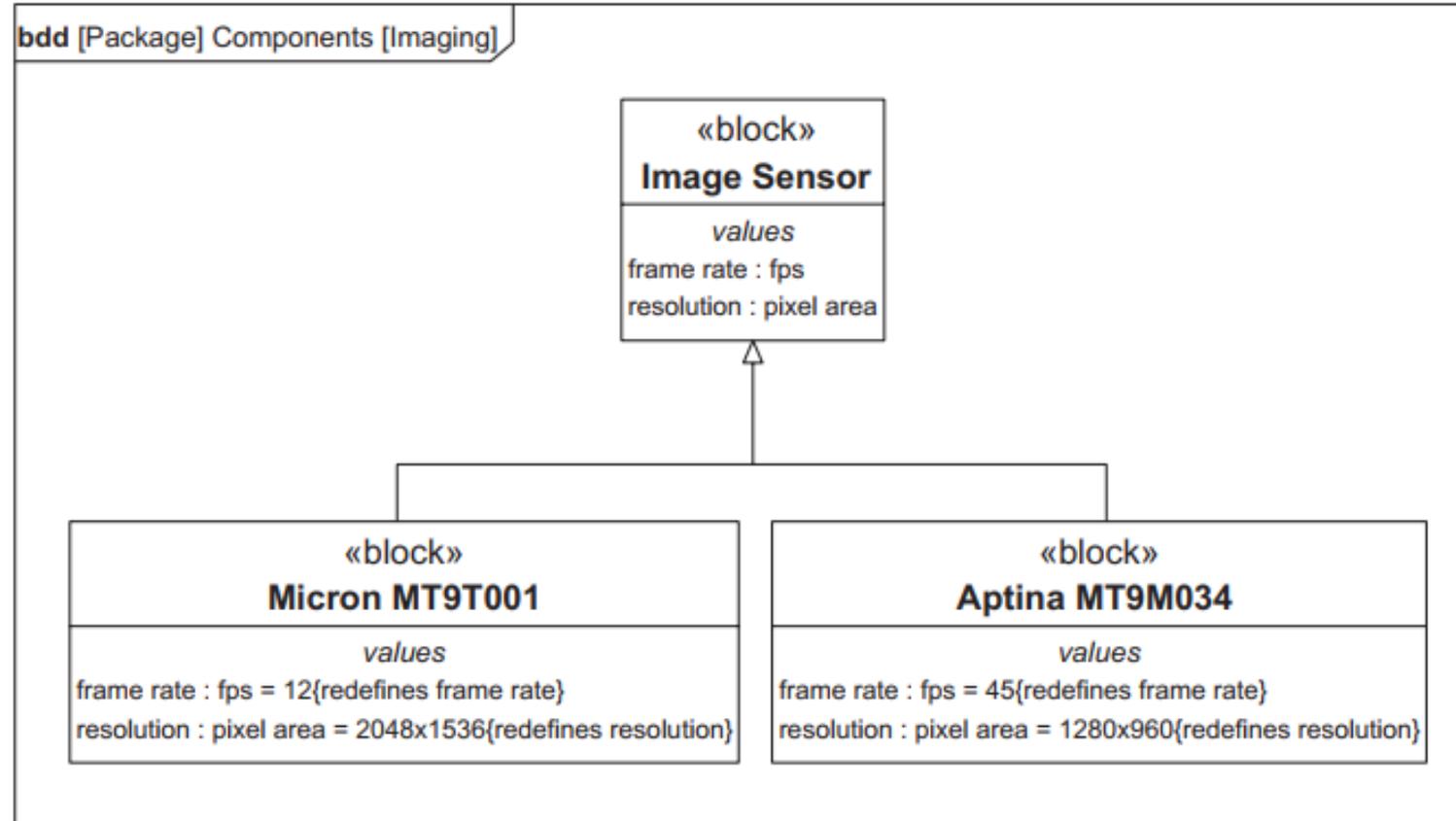


**FIGURE 7.48**

Example of block specialization.



# POLYMORPHISM



**FIGURE 7.52**

Two kinds of Imaging Assembly.



# FLUXOS

- Definir os fluxos (trocas) entre diferentes partes de um sistema pode fornecer uma visão de suas interações.
- Cada propriedade de fluxo tem um nome, tipo, multiplicidade e direção.
- Um fluxo de item especifica o tipo do item fluindo e a direção do fluxo.

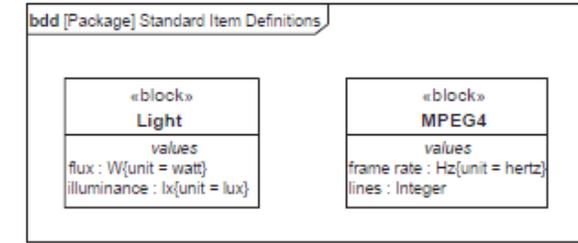


FIGURE 7.23

Items that flow in the *Camera* system.

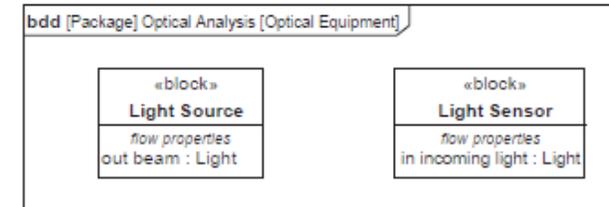


FIGURE 7.24

Flow properties on blocks.

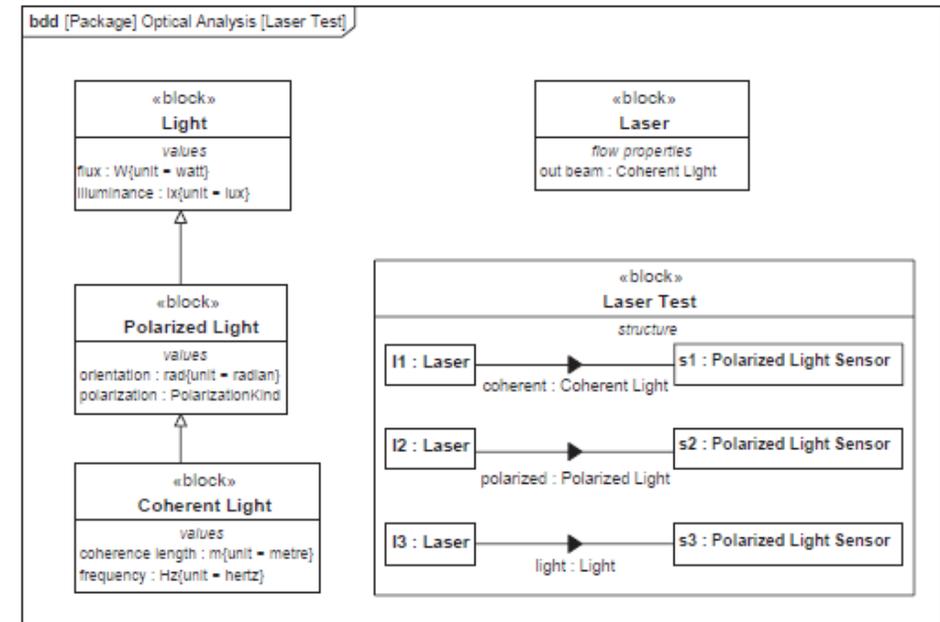


FIGURE 7.27

Item flows between parts.



# OS BLOCOS PODEM MODELAR O COMPORTAMENTO (FUNÇÕES)

- Um bloco pode designar um comportamento como seu comportamento principal, que começa a ser executado quando o bloco é instanciado. Outros comportamentos podem ser designados como métodos, que fornecem o detalhe de como as solicitações de serviço são tratadas.
- **Os comportamentos têm parâmetros que são usados para passar itens para dentro ou para fora do comportamento antes, depois e, às vezes, durante a execução.**
- Uma operação é um recurso comportamental que normalmente é acionado por uma solicitação síncrona (ou seja, quando o solicitante aguarda uma resposta). Cada operação define um conjunto de parâmetros que descreve os argumentos passados com a solicitação ou repassados depois que uma solicitação é manipulada, ou ambos. Uma recepção está associada a um sinal que define uma mensagem com um conjunto de atributos que representam o conteúdo da mensagem; Os parâmetros da recepção devem ser os mesmos que os atributos do sinal associado.
- **O comportamento principal (também chamado de comportamento do classificador) de um bloco começa a ser executado no início da vida útil do bloco e geralmente termina no final de sua vida útil, embora possa terminar antes disso.**
- O SysML suporta a noção de polimorfismo, o que significa que muitos blocos diferentes podem responder ao mesmo estímulo, mas cada um pode fazê-lo de uma maneira específica, invocando um método específico.

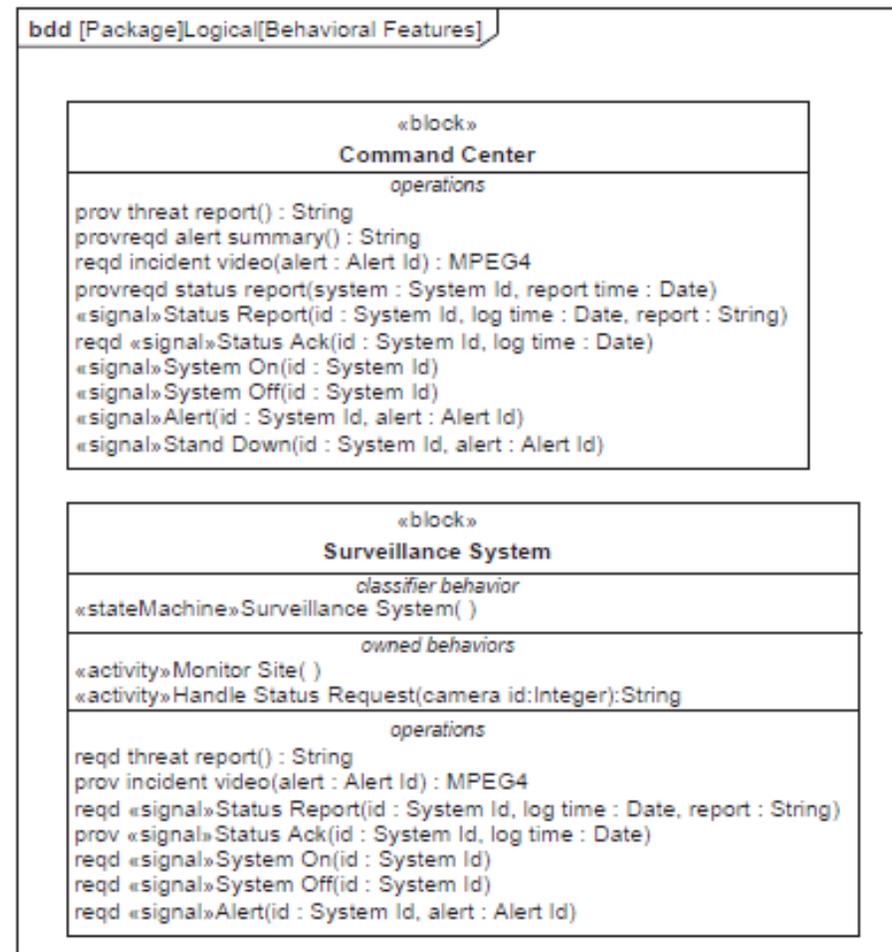


FIGURE 7.30

Blocks with behavioral features.



# PORTAS

- Uma porta representa um ponto de acesso no limite de um bloco e no limite de qualquer parte ou referência digitada por esse bloco. Um bloco pode ter muitas portas que especificam pontos de acesso diferentes. As portas podem ser conectadas umas às outras por conectores em um diagrama de blocos interno para suportar a interação entre as partes.
  - A full port é equivalente a uma parte no limite do bloco pai que é disponibilizada como um ponto de acesso de/para o bloco.
  - A proxy port não constitui uma parte de seu bloco pai, mas fornece acesso externo de/para os recursos de seu bloco pai ou partes do bloco sem modificar suas entradas ou saídas.

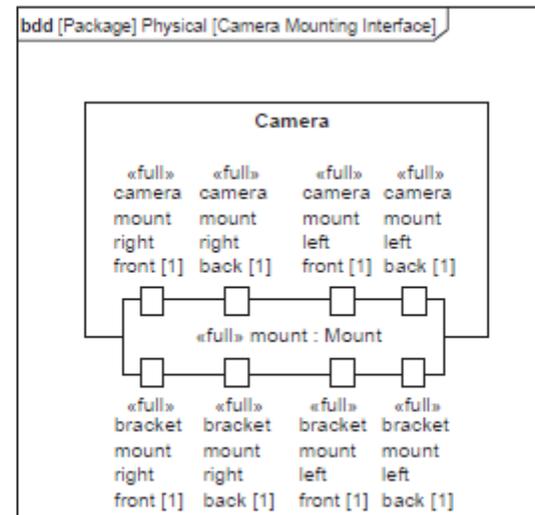


FIGURE 7.32

A full port with nested ports.

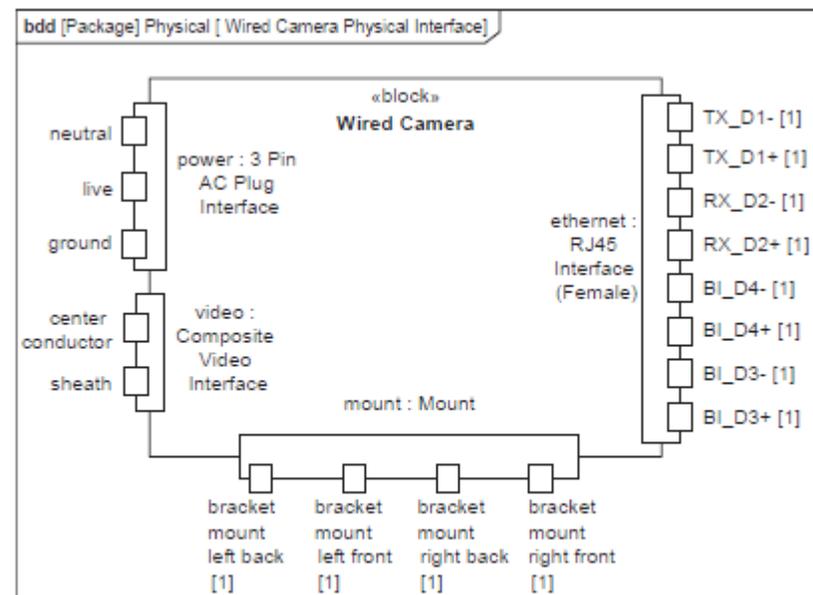
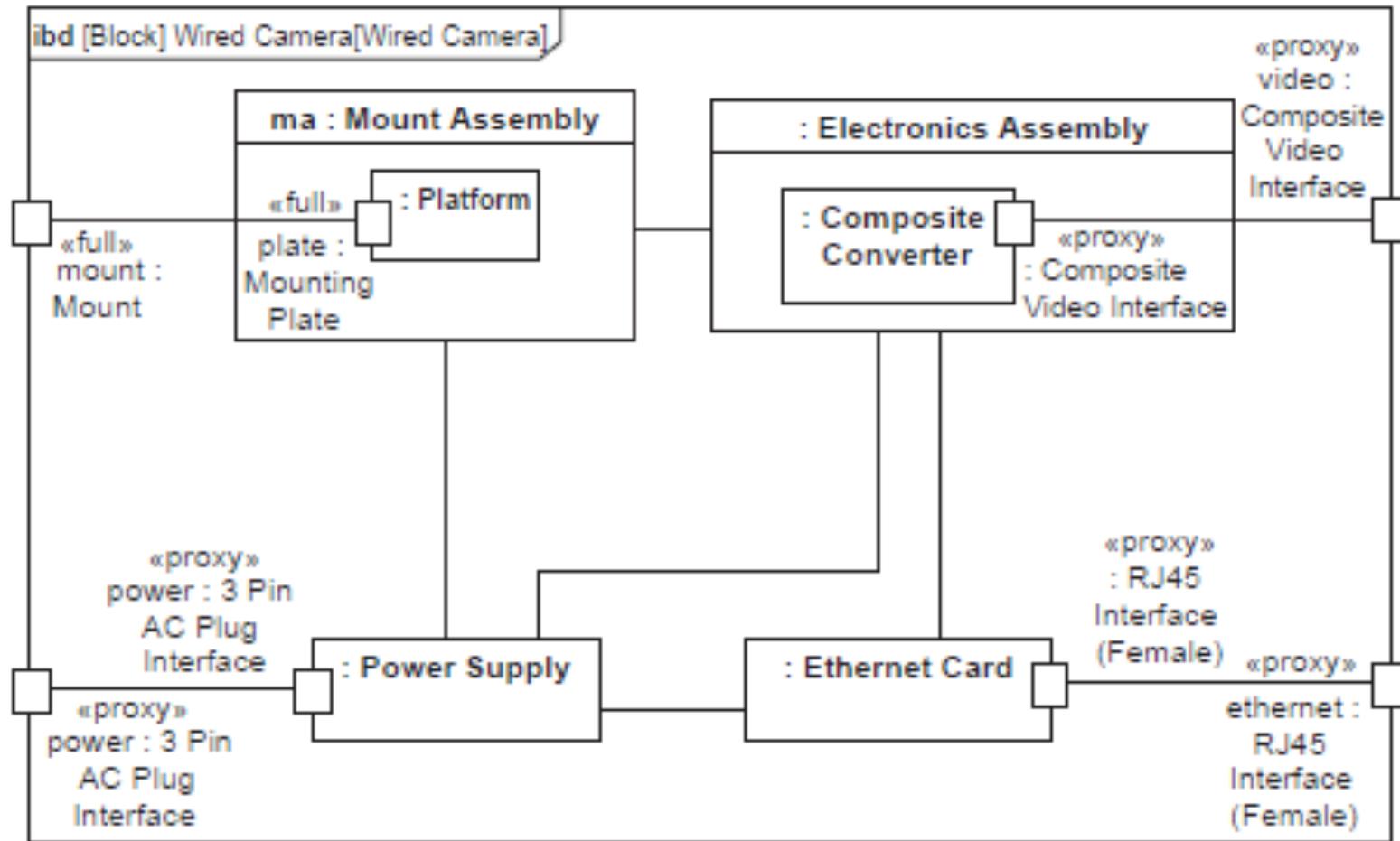


FIGURE 7.34

A block with nested ports.



# CONECTANDO PORTAS

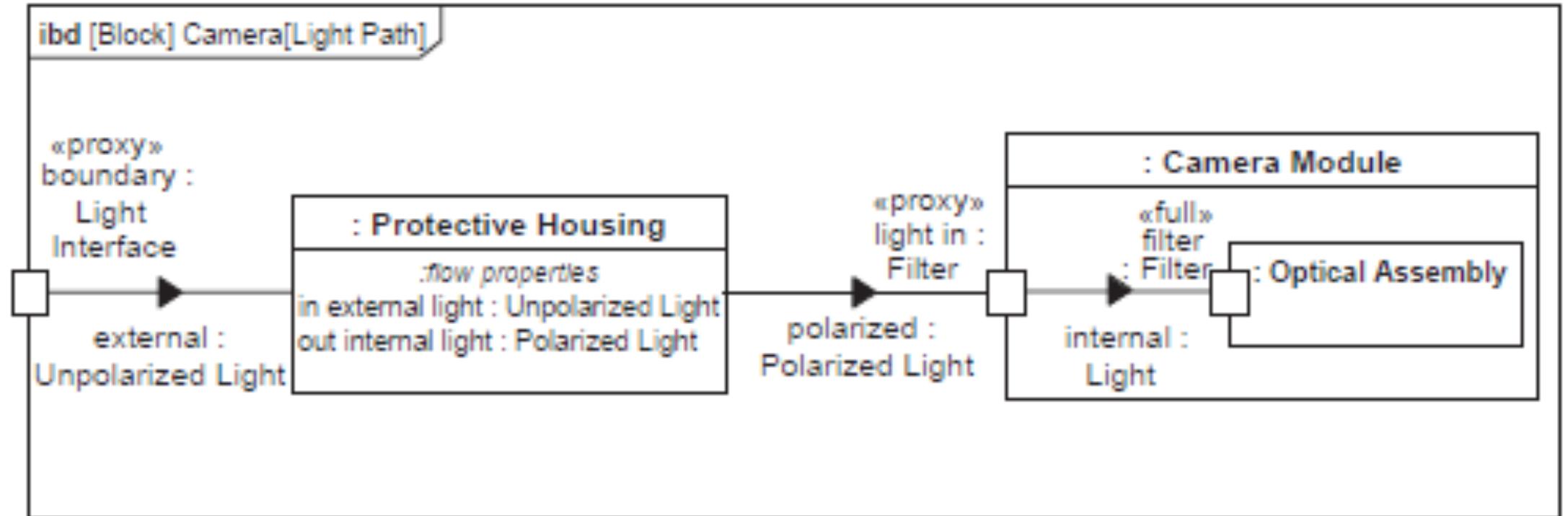


**FIGURE 7.35**

Connecting ports internally to a block.



# FLUXOS DE ITENS ENTRE PORTAS.

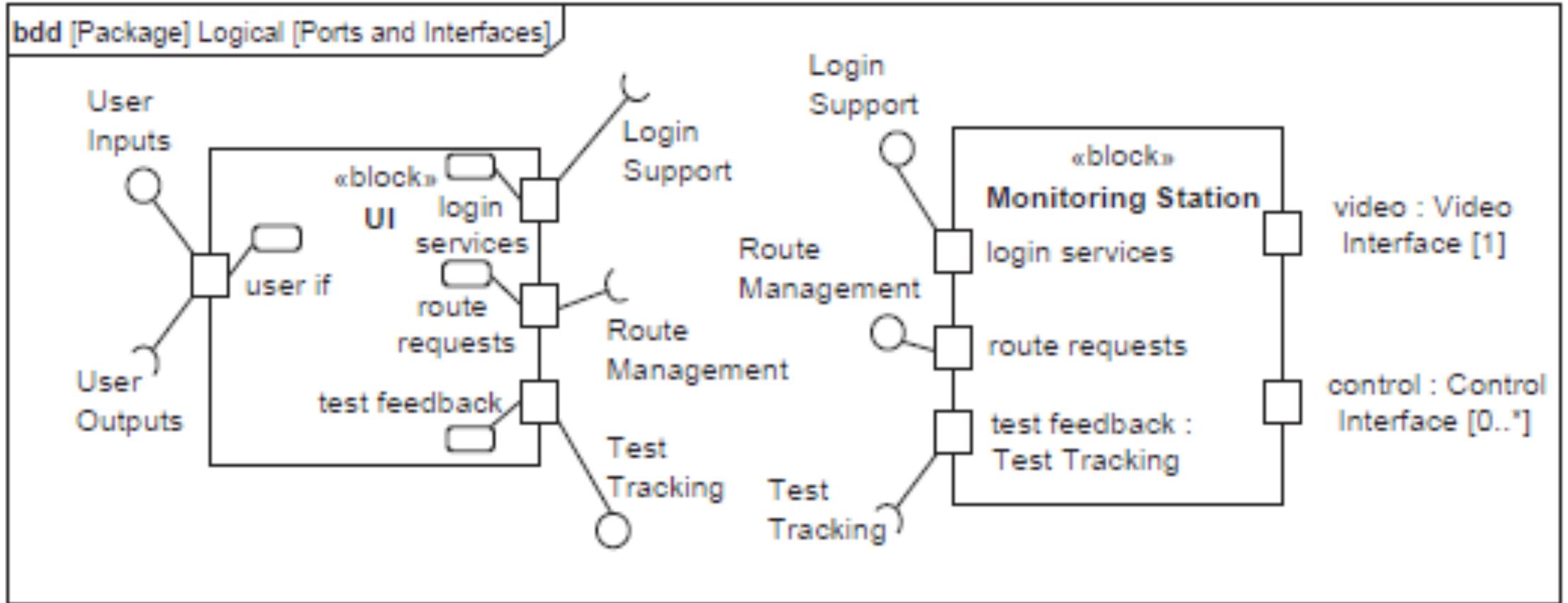


**FIGURE 7.44**

Item flows between ports.



# INTERFACES



41

**FIGURE 7.46**

Defining a service-based interface using proxy ports.



# DIAGRAMA DE MODELAGEM PARAMÉTRICA

MODELING CONSTRAINTS WITH PARAMETRICS – CHAPTER 8



# INTRODUÇÃO AO DIAGRAMA PARAMÉTRICO

- Um esforço de projeto típico inclui a necessidade de executar **muitos tipos diferentes de análises de engenharia**, como estudos comerciais, análise de sensibilidade e otimização de projeto.
- O SysML introduz o bloco para suportar a construção de modelos paramétricos.
  - **Um bloco de restrição** é um tipo especial de bloco usado para definir equações para que elas possam ser reutilizadas e interconectadas.
- Os blocos de restrição têm duas características principais: **um conjunto de parâmetros e uma expressão que restringe esses parâmetros.**



# DIAGRAMA PARAMÉTRICO

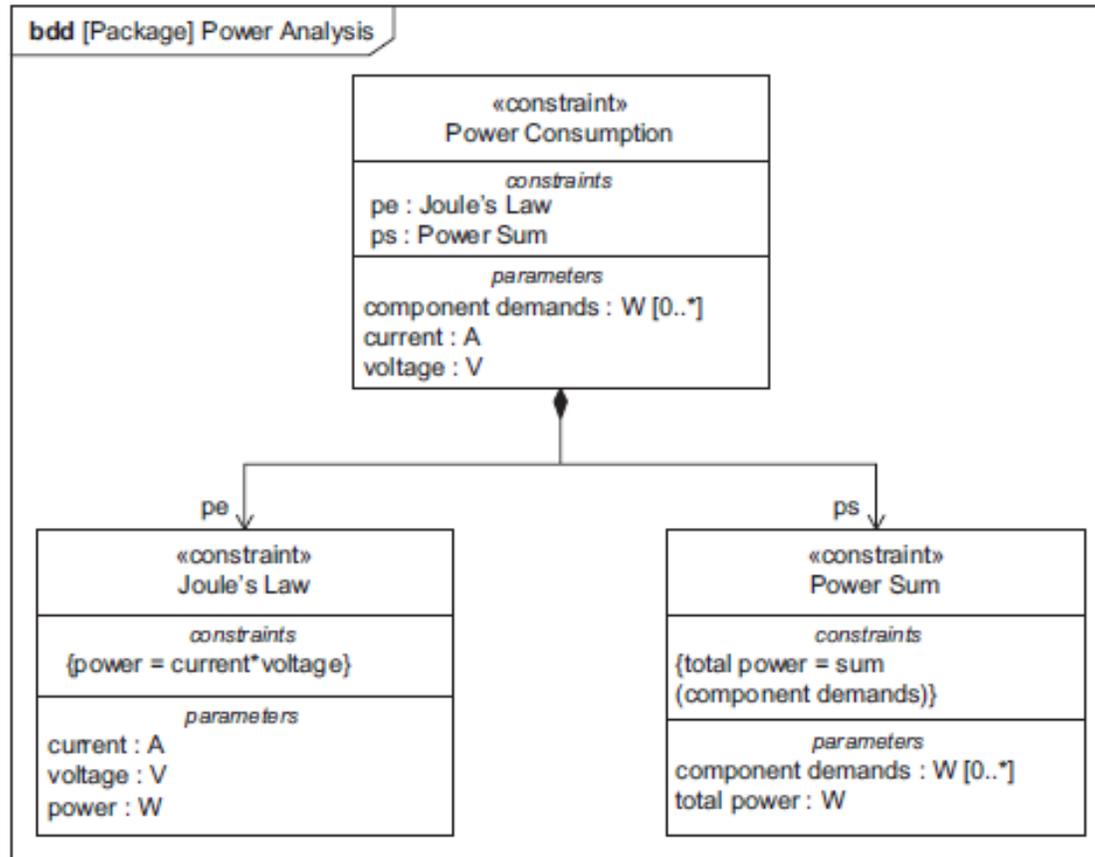


FIGURE 8.5

A hierarchy of constraints on a block definition diagram.

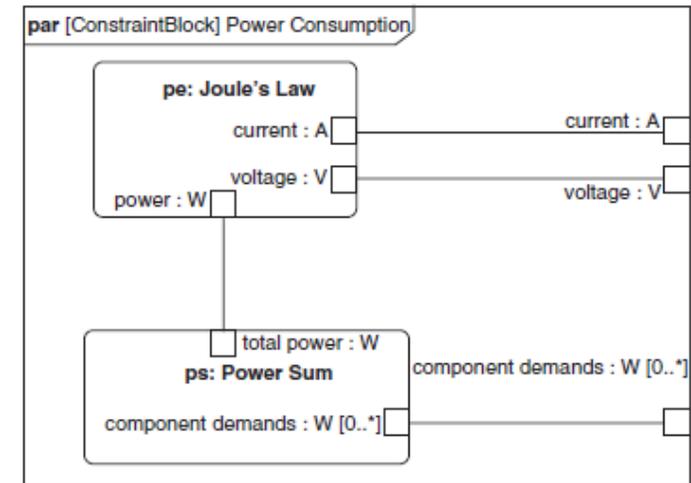
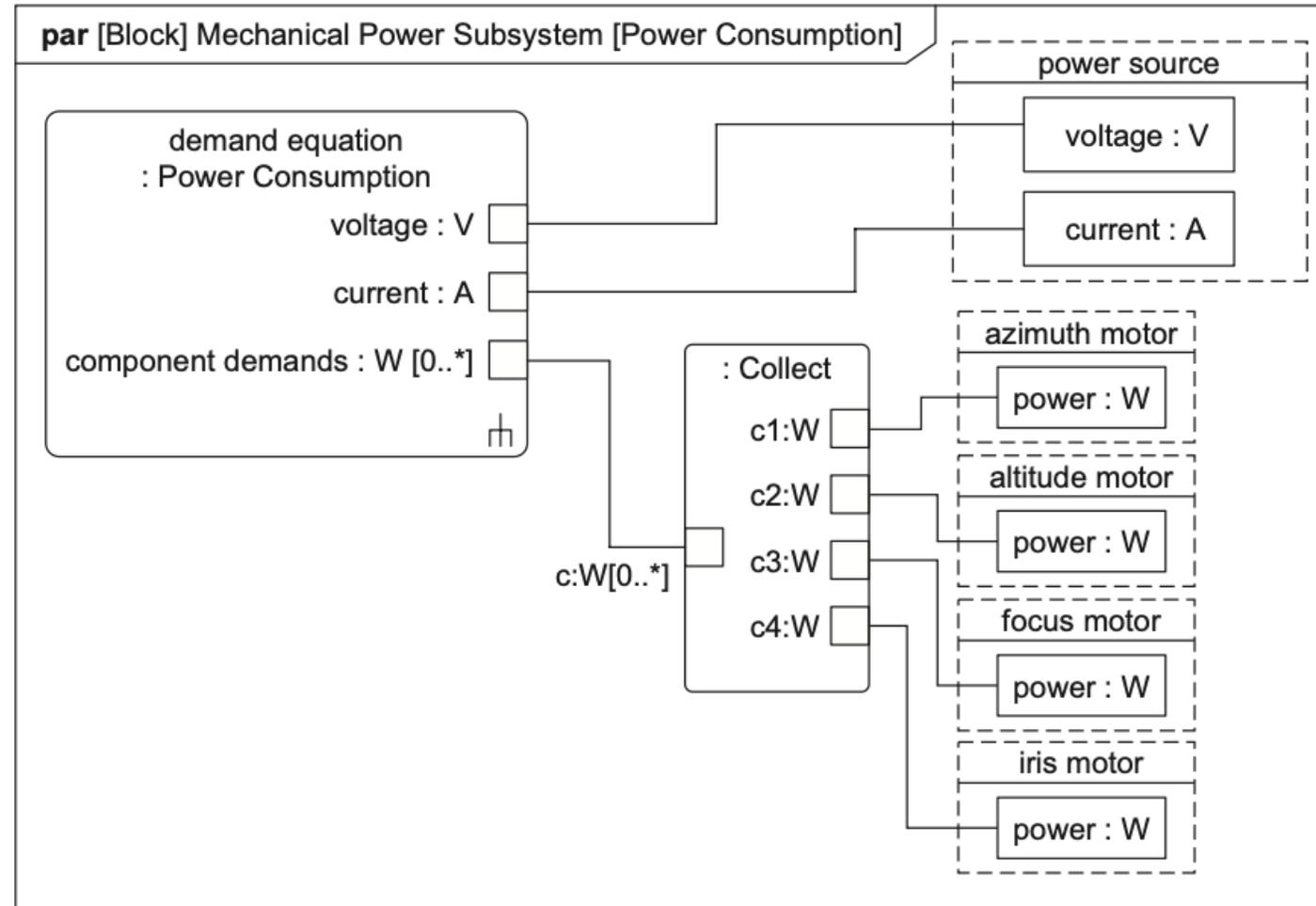


FIGURE 8.6

Internal details of the power consumption equation using a parametric diagram.



# INTERLIGANDO PARAMETROS

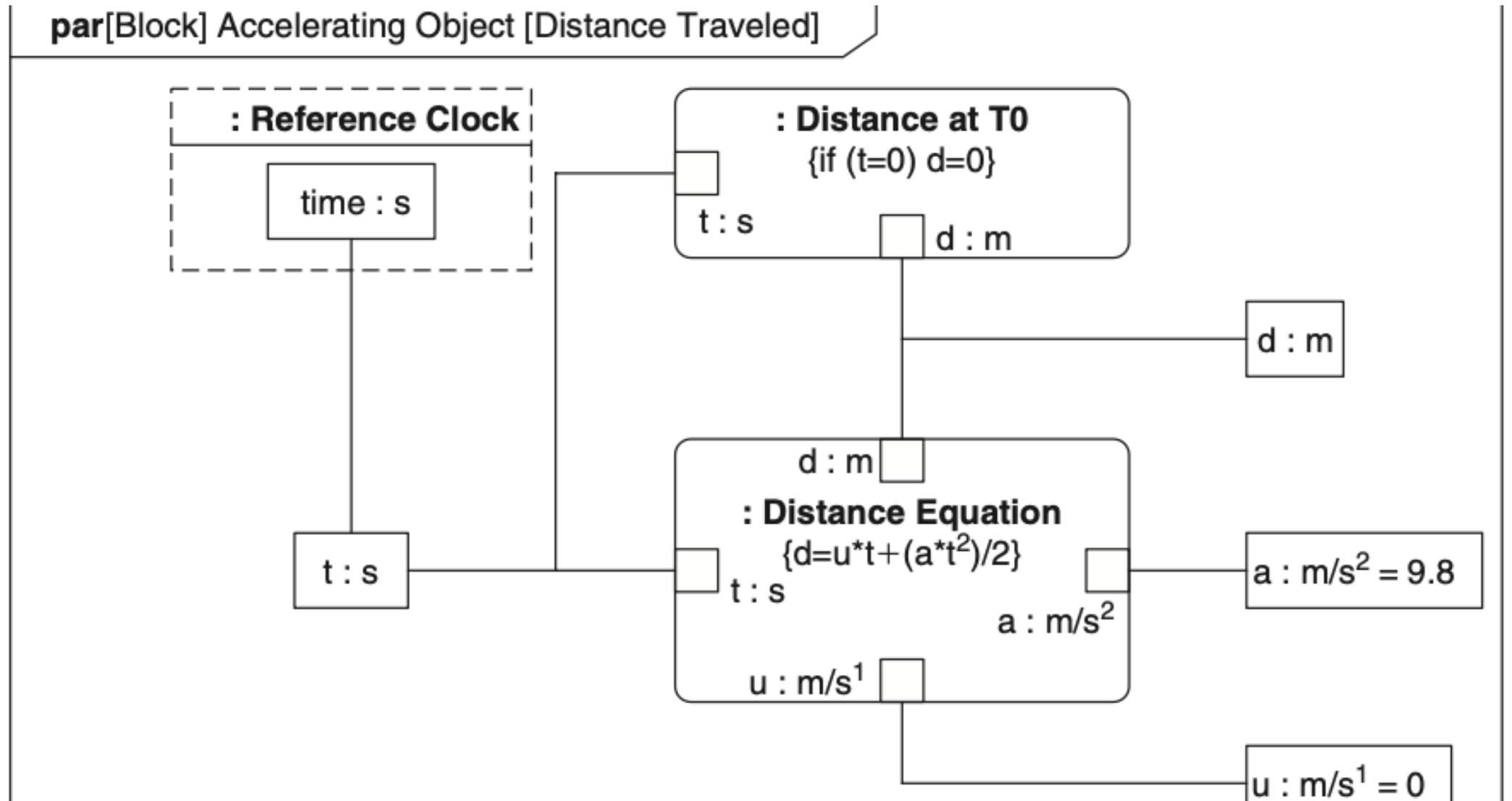


**FIGURE 8.7**

Binding constraints to properties on a parametric diagram.

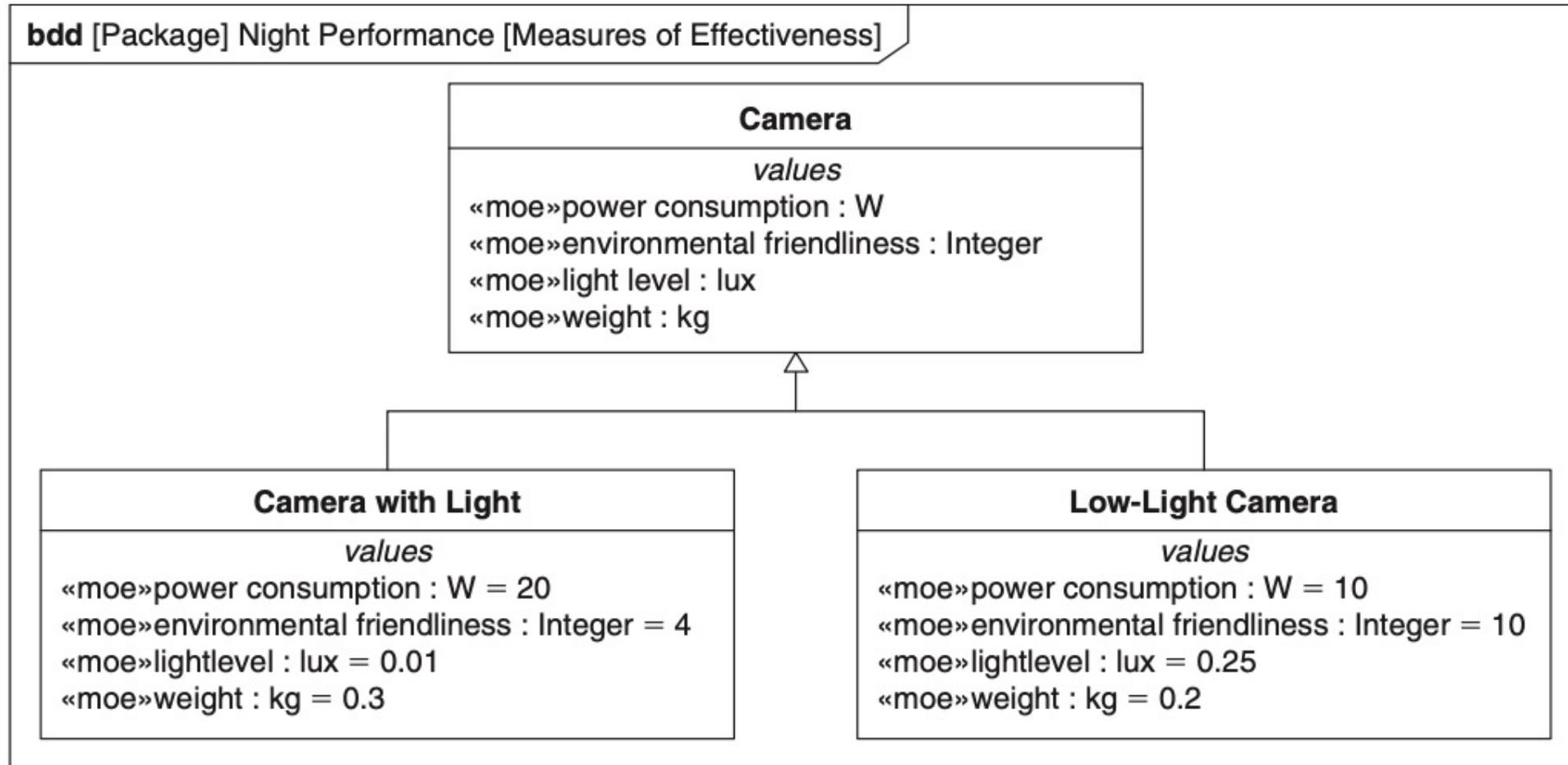


# POSSIBILIDADE DE SIMULAÇÃO DINÂMICA





# PARAMETRIZANDO PERMITE COMPARAÇÃO



**FIGURE 8.14**

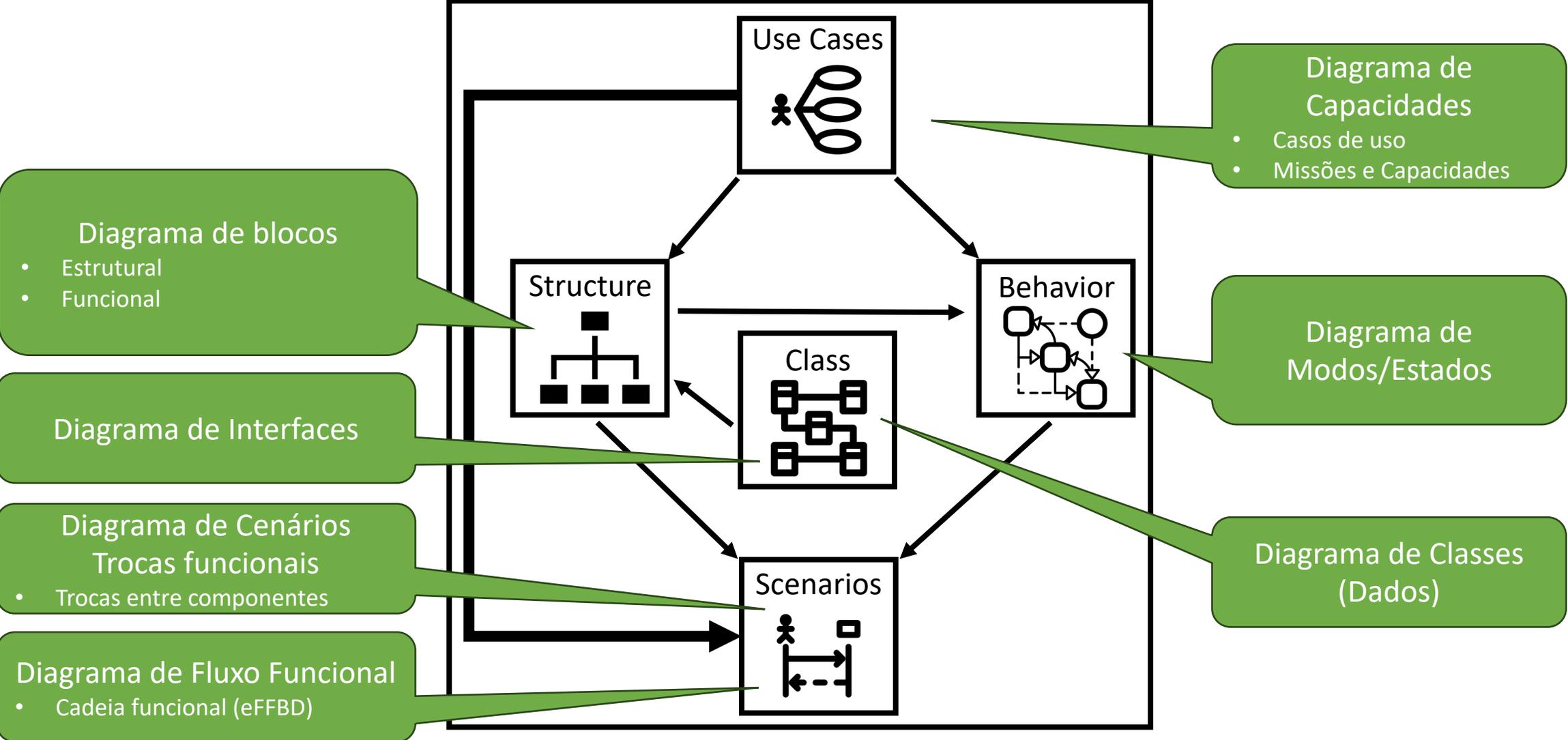
Two variants of a camera for handling low-light conditions.



# CONSIDERAÇÕES FINAIS



# ROTEIRO DOS DIAGRAMAS





# TEORIA

- Stakeholders
- CONOPs
  - *Validação*
- Requisitos
  - *Verificação*
- Ciclo de Vida
- Função
- Arquitetura
  - Funcional / Hierárquica
  - Coesão e Acoplamento
- Árvore de Produto /  
Decomposição do Sistema
- Dualidade  
Comportamento/Estrutura