

Modelando Operações

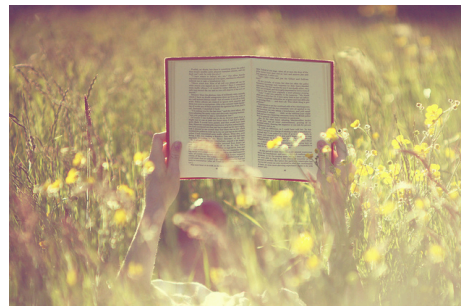
Letícia Duboc
IME / DICC - UERJ
2º Semestre de
2015

Flickr CC By play4smee

Bibliografia

- Bibliografia

A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*, Wiley, 2009 -
Capítulo 12



FlickrCCByBethan

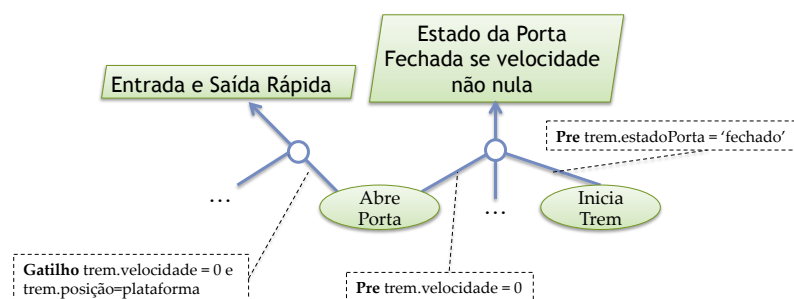
Agenda

- Modelo de Operações
- Objeto Conceitual
- Operações
- Operacionalizações
- Operações no KAOS
- Operações na UML



O Modelo de Operações

- Representa a visão **funcional** do sistema
 - A **dimensão “o quê”** da engenharia de requisitos
 - Determina os **serviços** que o sistema deve prover para satisfazer suas metas



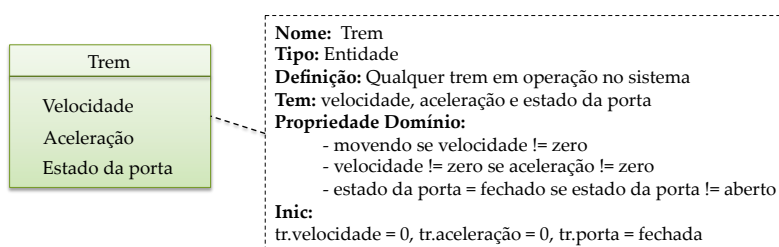
Porque Construir Modelo de Operações?

- Especificação do software para desenvolvimento
- Estimar tempo e custos
- Manual do usuário
- Suporte a argumentos de satisfação, rastreamento e evolução

Objetos Conceituais

- Um objeto é um conceito específico do domínio manipulado pelo sistema

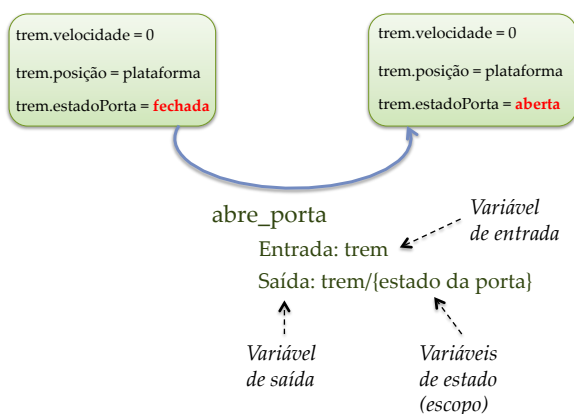
Ex. Trem, Plataforma, Bloco



Tipos: entidade, agente, associação e evento

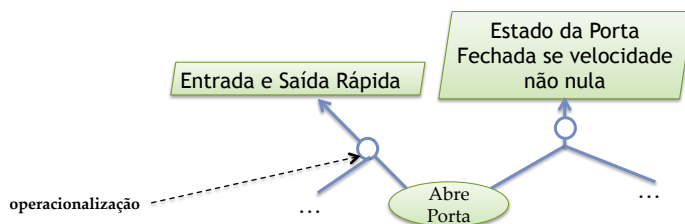
O que são operações?

- Uma operação é uma **relação binária** entre estados do sistema

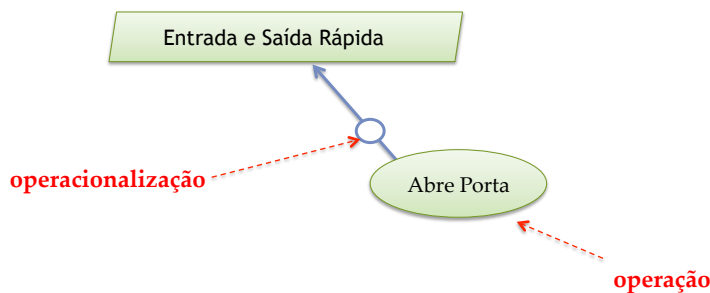


Operacionalização

- Operações **operacionalizam** metas
 - uma ou mais metas
- Operações **não devem ser refinadas**
 - operações devem ser atômicas
 - operações devem ser determinísticas
- Categorias**
 - de software – Ex. *PlanejaReunião (software Agenda)*
 - do ambiente – Ex. *EnviaRestrição (Participante)* – tarefa



São Conceitos Diferentes!



Elementos da Operação

- **Nome**
 - único
- **Definição**
 - clara e precisa
- **Categoria**
 - software/ ambiente
- **Entrada / Saída**
 - instância: Objeto{escopo}
- **Agente executor**
 - único
 - deve poder controlar/ monitorar os atributos do objeto
- **Pre e pós condições de domínio**
 - descritivas, não prescritivas
 - não determinam quando a operação deve ou não ser aplicada

Operação: *AbrePorta*

Def: *Operação do software controlando a abertura de todas as portas do trem*

Categoria: *Software*

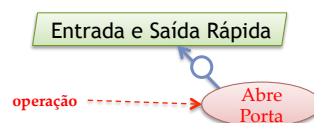
Entrada: *trem:Trem/{velocidade, posição}*

Saída: *trem:Trem/estado da porta*

Agente: *ControladorDoTrem*

Pré-condição de domínio:
trem.estadoPorta = fechado

Pós-condição de domínio:
trem.estadoPorta = aberta



Elementos da Operacionalização

- **Pré-condição Necessária**
 - condição **necessária** nas variáveis de estado
 - representa uma **permissão**
- **Pré-condição de Gatilho**
 - condição **suficiente** nas variáveis de estado
 - representa um **obrigação**
- **Pós-condição Necessária**
 - efeito **adicional** para garantir a meta

Pré-condição necessária:

trem.velocidade = zero

Pré-condição de gatilho:

trem.velocidade = zero &

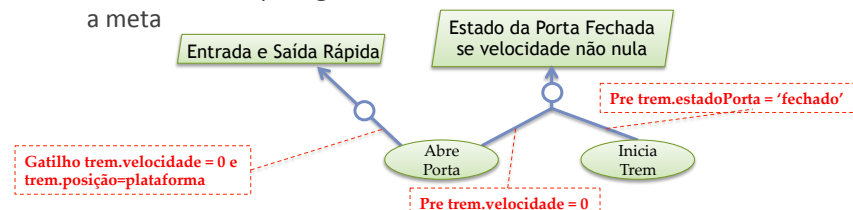
trem.posição = plataforma

Pós-condição necessária :

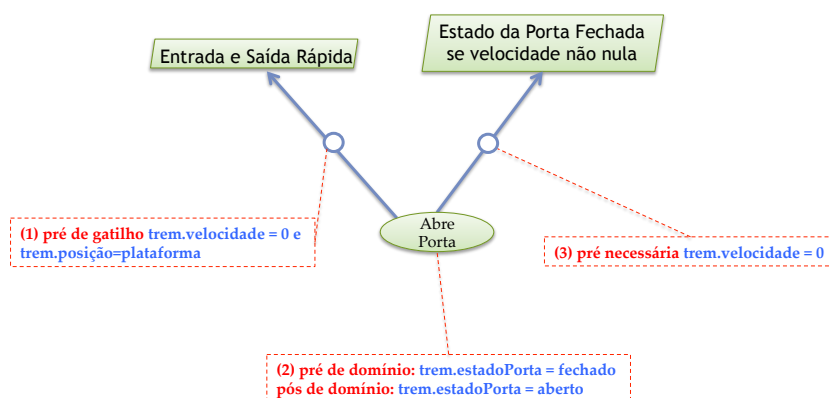
operação: enviaComandoAceleração

trem.aceleração = [X, Y]

(em intervalo seguro)



Qual a Diferença entre Pré-Condições?



O que Ocorre?

1. O que ocorre quando uma **pré-condição de gatilho** passa a ser verdadeira?
2. O que ocorre quando uma **pré-condição necessária** passa a ser verdadeira?
3. O que ocorre se a **pré-condição de gatilho** passa a ser verdadeira, mas a **pré-condição de domínio** é falsa?
 - E se a pré-condição de domínio passa a ser verdadeira?
4. O que ocorre se a **pré-condição necessária** passa a ser verdadeira, mas a **pré-condição de domínio** é falsa?
 - E se a pré-condição de domínio passa a ser verdadeira?

Unindo os Conceitos - exemplo

Operação *AbrePorta*

Definição *Operação de software controlando abertura de todas as portas do trem*

Entrada *trem:Trem*

Saída *trem:Trem/estadoPorta*

Pré-cond. de domínio *trem.estadoPorta = fechada*

Pós-cond. de domínio *trem.estadoPorta = aberta*

Pré-cond. necessária para *Alcança[Porta fechada se velocidade medida não nula]: trem.velocidade = 0*

Pré-cond. necessária para *Alcança[Entrada e Saída Segura]: trem.posição = plataforma*

Pré-cond. de gatilho para *Alcança[Entrada e Saída Rápida]: trem.posição = plataforma e trem.velocidade = 0*

Unindo os Conceitos – outro exemplo

Operação *PlanejaReunião*

Definição *Operação de software marcando a data e local da reunião a partir das restrições dos participantes convidados*

Entrada *r:Reunião*

Saída *r:Reunião/{data, local}*

Pré-cond. de domínio *reunião r não tem data ou local*

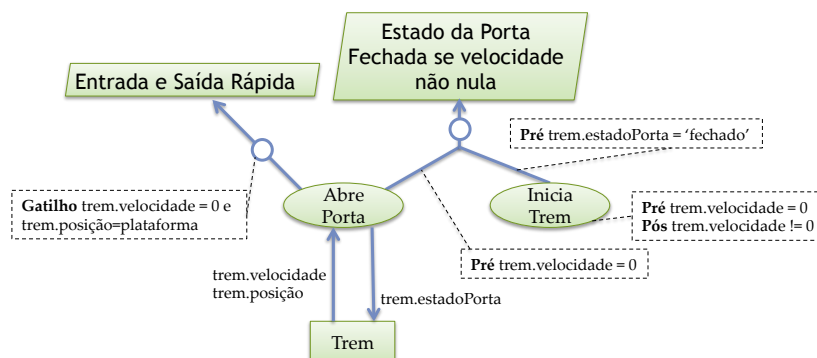
Pós-cond. de domínio *reunião r tem data e local*

Pré-cond. necessária para *Alcança[Data conveniente para reunião]: todos os convidados importantes estão disponíveis*

Pós-cond. necessária para *Alcança[Data conveniente para reunião]: a data marcada não está entre as datas excluídas nas restrições de nenhum participante convidado a reunião r*

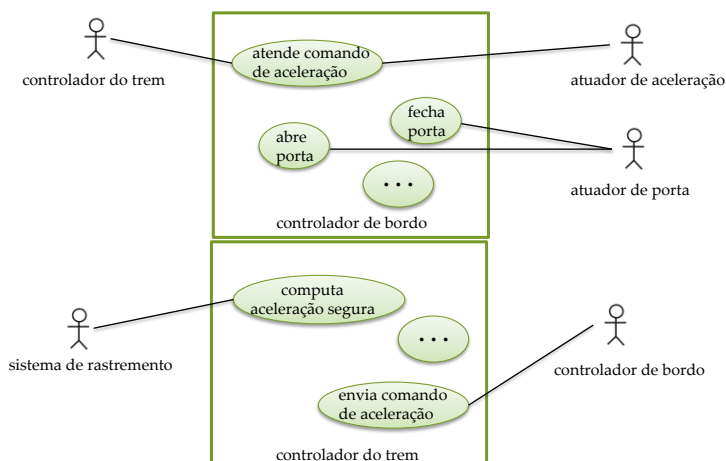
Representando Modelos de Operação no KAOS

- Gráfico anotado mostrando:
 - operações do sistema
 - links de operacionalização
 - links de entrada e saída a objetos
 - pré e pós condições de operações e operacionalizações



Representando Modelos de Operação em UML

- Retângulo inclui todas as operações de um agente
 - Operação conectada a agente externo quando existe interação



Não esqueçam!

- Operações podem operacionalizar múltiplas metas
 - *AbrePorta*
 - *Alcança[Porta fechada se velocidade medida não nula]*
 - *Alcança[Entrada e Saída Segura]*
- Metas são operacionalizadas por múltiplas operações
 - *Alcança[Porta fechada se velocidade medida não nula]*
 - *AbrePorta*
 - *FechaPorta*
 - *IniciaTrem*
- Refine metas, não operações