Diagramas de Interacção

Diagramas de interacção

- Mostram comunicação entre objectos
- Objectivo
  - especificar a realização de uma operação
  - especificar a realização de um *use case*
- Dois tipos:
  - Diagramas de sequência
  - Diagramas de colaboração

Diagramas de sequência e de coloboração

- Ambos especificam a mesma informação
- Cada um foca aspectos diferentes
  - **Diagrama de sequência é "orientado" no tempo**
    - ❖mostra graficamente a ordem das mensagens
    - ❖não mostra como se obtém o objecto receptor
  - **Diagrama de colaboração é "orientado" no espaço**
    - ❖mostra relações estáticas e dinâmicas entre objectos
    - ❖a ordem das mensagens dada explicitamente
    - ❖tempo não é uma dimensão

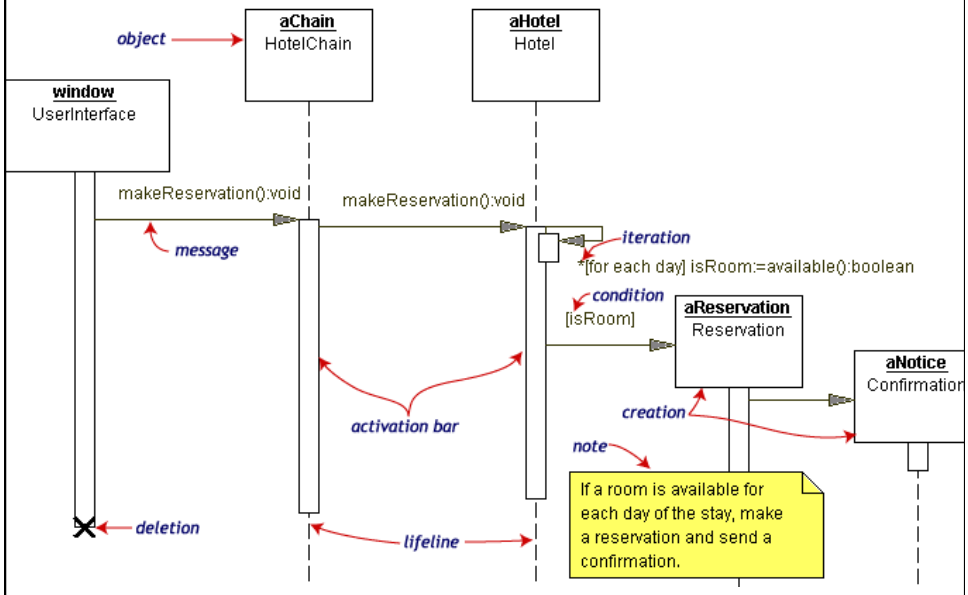
Diagramas de sequência

- Mostra a comunicação (troca de mensagens) entre objectos necessária para a execução de um caso de uso
- Fonte: informação disponível nos *use cases* e noutros modelos disponíveis (por exemplo, diagramas de actividade).
- Como um *use case* descreve todas as vertentes de uma dada funcionalidade (incluindo casos de erro e excepção), podemos optar por construir um diagrama de sequência por cenário.

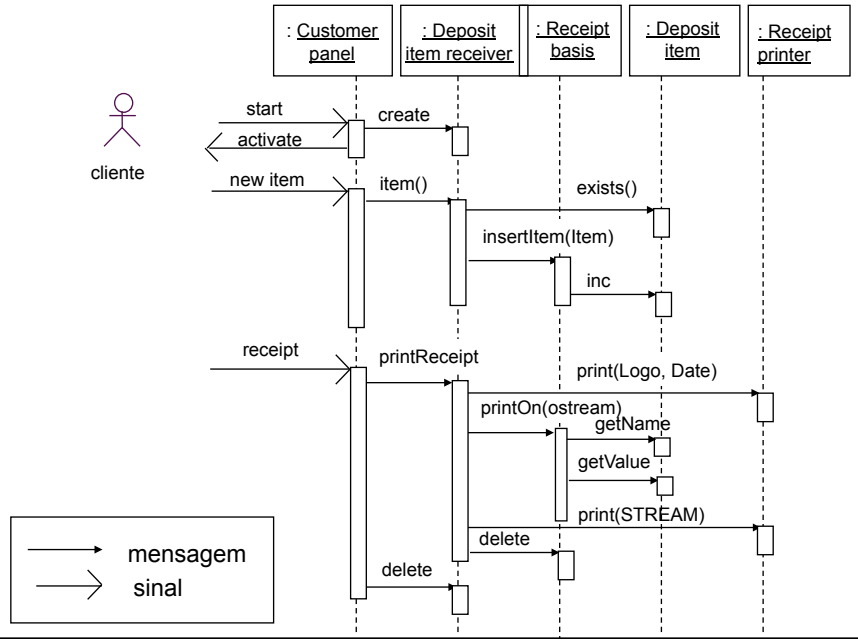
Diagramas de sequência: notação

- Objectos (rectângulos) organizados ao longo do eixo X
- Linha de vida de um objecto: linha tracejada que representa a existência de um objecto num período de tempo
- Mensagens (setas), ordenadas temporalmente, no eixo Y
- Fluxo de controlo execução: rectângulo estreito que mostra o período de tempo desde que o objecto recebe a mensagem até que fornece uma resposta (tempo de execução de uma operação);
  - este tempo pode incluir tempos de execução de operações subordinadas.

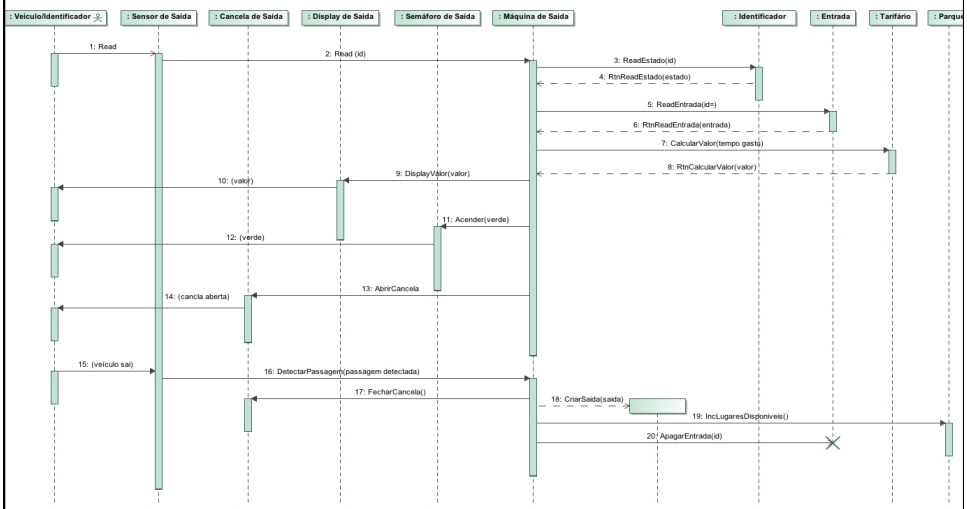
Diagrama sequência: reserva quarto de hotel



Exemplo: máquina de reciclagem



DS para o cenário "veículo autorizado sai do parque"



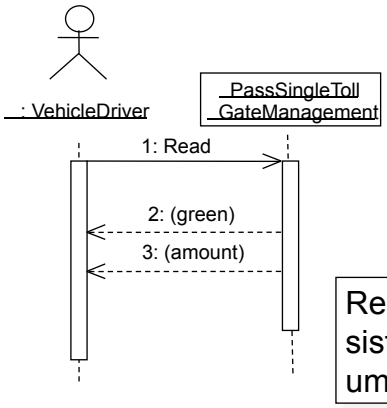
- Leading object?
- Separation of Concerns

Problemas na construção do DS

- Nem sempre é fácil construir um DS
  - Objectos de controlo estão envolvidos sempre que detectamos que um objecto de interface lidera/controla a execução;
  - O DS pode construir-se por partes, usando os três tipos de objectos como "guia" de construção.
- Algumas regras de construção:
  - A primeira mensagem é sempre enviada por um actor
  - A primeira mensagem é sempre recebido por um objecto de interface
  - Adicionar um objecto de controlo sempre que o objecto de interface se torne o "decisor"

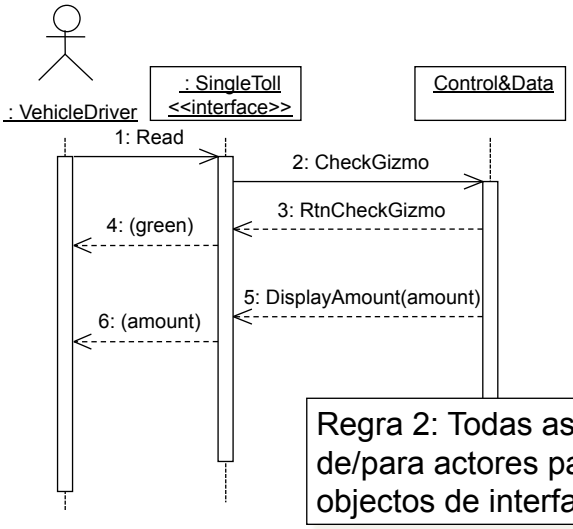
DS, passo a passo

Exemplo: veículo passa numa portagem de ponto único (e.g., ponte 25 Abril).



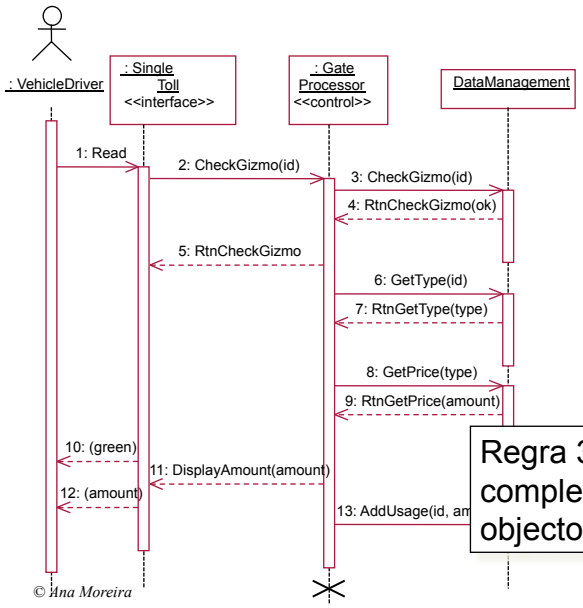
Regra 1: ao iniciar, o sistema é visto como uma caixa preta.

DS com um objecto de interface



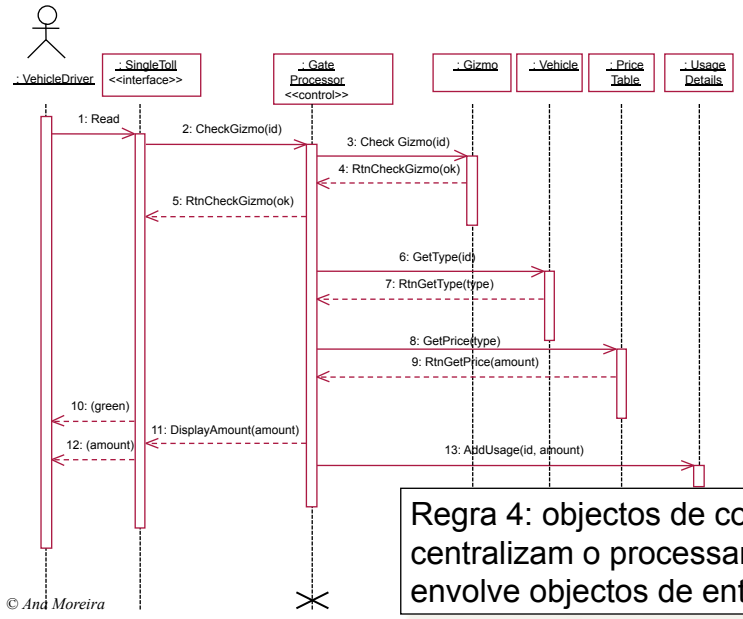
Regra 2: Todas as mensagens de/para actores passam por objectos de interface.

DS com objectos de interface e controlo



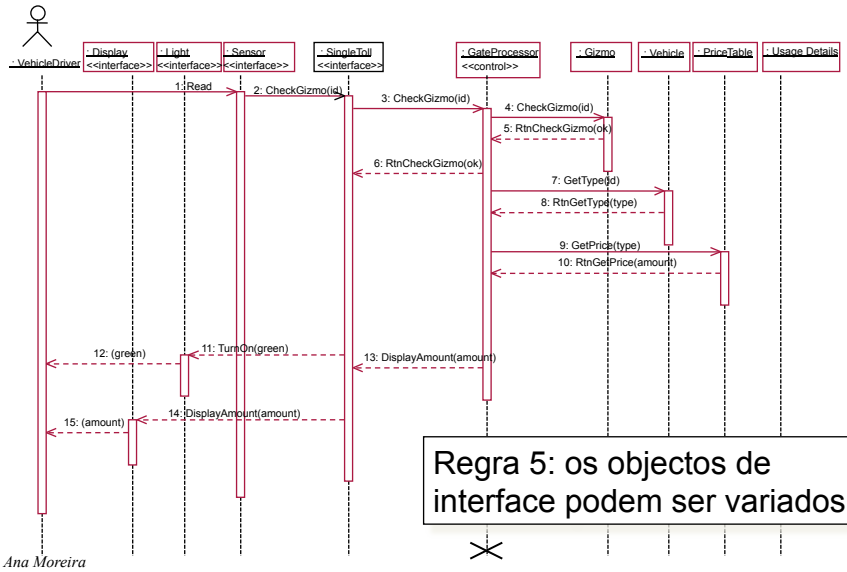
Regra 3: para funcionalidades complexas precisamos de objectos de controlo.

DS com objectos de interface, controlo e entidade



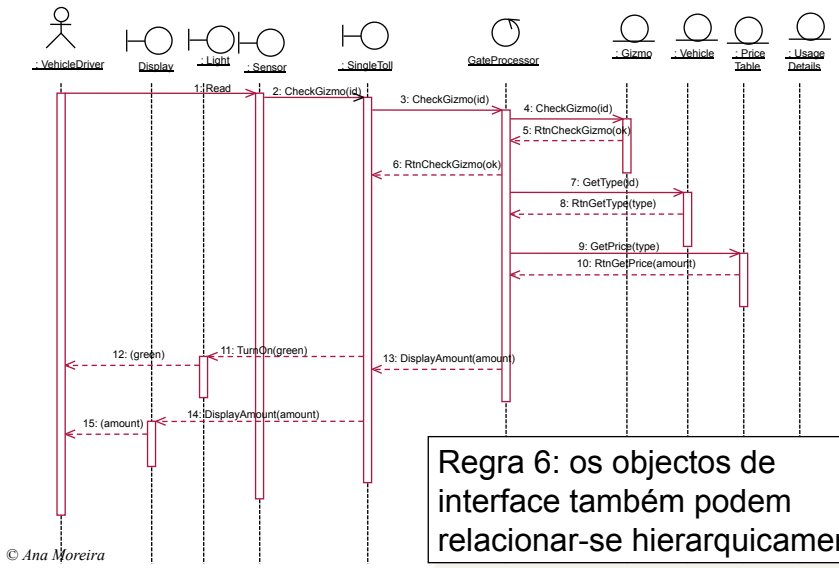
Regra 4: objectos de controlo centralizam o processamento que envolve objectos de entidade.

DS com os componentes da portagem

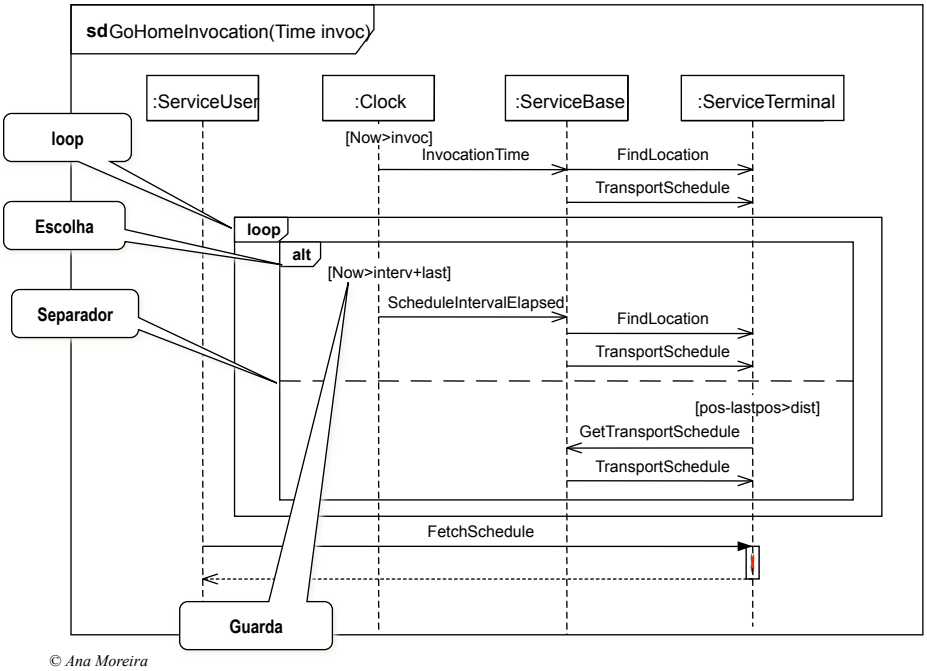


Regra 5: os objectos de interface podem ser variados.

DS: notação alternativa



Regra 6: os objectos de interface também podem relacionar-se hierarquicamente.



Combined Fragment Types (1 of 2)

- Alternatives (**alt**)
  - choice of behaviors – at most one will execute
  - depends on the value of the guard ("else" guard supported)
- Option (**opt**)
  - Special case of alternative
- Break (**break**)
  - Represents an alternative that is executed instead of the remainder of the fragment (like a break in a loop)
- Parallel (**par**)
  - Concurrent (interleaved) sub-scenarios
- Loop (**loop**)
  - Optional guard: [<min>, <max>, <Boolean-expression>]
  - No guard means no specified limit

Combined Fragment Types (2 of 2)

- Negative (**neg**)
  - Identifies sequences that must not occur
- Critical Region (**region**)
  - Traces cannot be interleaved with events on any of the participating lifelines
- Assertion (**assert**)
  - Only valid continuation

Dos diagramas de actividades aos DS

- A estrutura básica do DS pode extrair-se do diagrama de actividades.
- Em particular:
  - Actividade :: Operação em classe
  - Transição :: Mensagem
  - Branching-Merge :: Alt
  - Fork-Join :: Par
  - (Ciclo: transição para trás) :: Loop

Dos diagramas de Sequência aos Diagramas de Classes

- O diagrama de classes de domínio deve completar-se com a informação dos diagramas de sequência.
- Em particular:
  - Linha de Vida :: Classe
    - Interface, controlo, (entidade)
  - Mensagem :: Operação
  - Mensagem e argumentos :: Associação
  - Mensagem :: Dependência

Complementos ao Diagrama de Classes

Diagrama de classes (parcial) resultante

Exemplo: veículo passa numa portagem de ponto único (e.g., ponte 25 Abril).

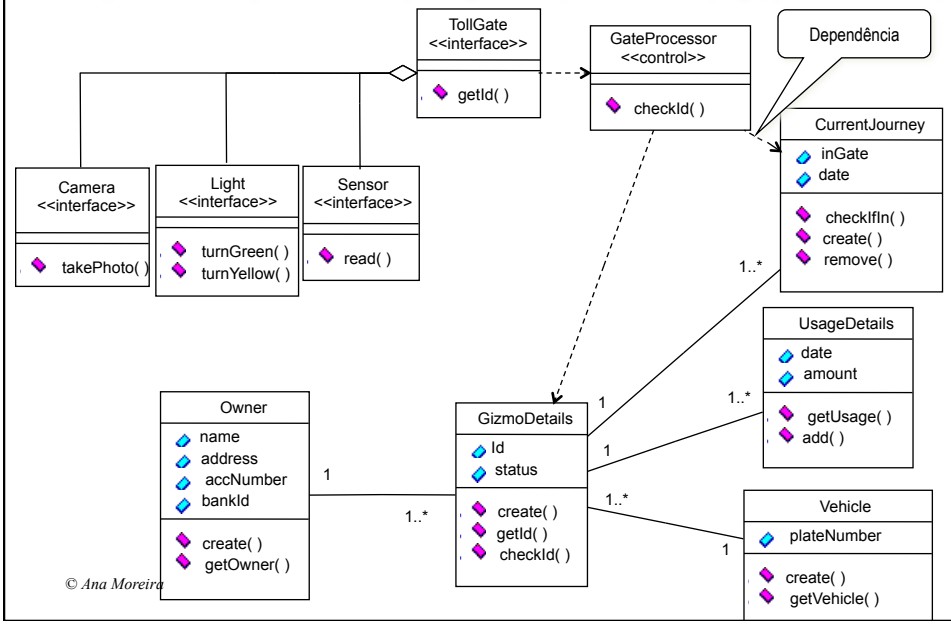


Diagrama de Colaboração

Diagrama de Colaboração

- Um diagrama de colaboração é um diagrama de interacção que enfatisa a organização estrutural dos objectos que enviam e recebem mensagens.
- Ele mostra o conjunto de objectos, ligações (*links*) entre estes objectos, e mensagens enviadas e recebidas por esses objectos.
- Diagramas de colaboração são úteis para ilustrar a visão dinâmica do sistema.

Diagrama de Colaboração

- Oferece uma segunda forma de mostrar a sequência na qual os eventos ocorrem
- Os objectos são mostrados em rectângulos ligados por linhas que indicam *links* entre eles
- Números indicam a ordem na qual as operações são executadas
- Os números são escritos junto dos nomes das mensagens e uma seta indica o sentido do fluxo

Diagrama de Colaboraçã: exemplo simples

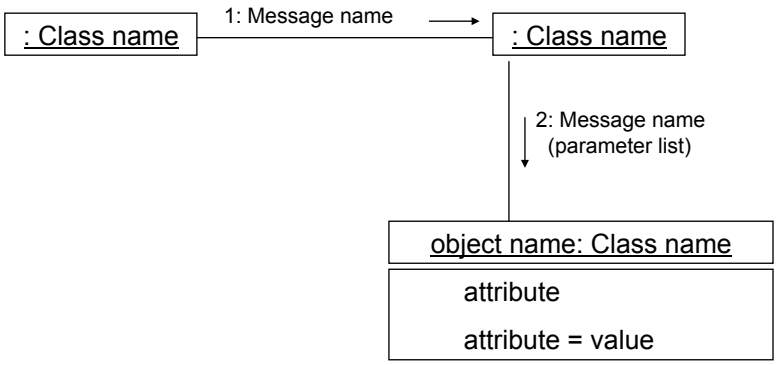


Diagrama de colaboração (cont)

- A numeração associada às mensagens podem representar aninhamento.
  - Por exemplo mensagem 1.1 é a 1ª mensagem aninhada na mensagem 1, 1.2 é a segunda e assim sucessivamente.
- Equivalência semântica com os DS: pode converter-se automaticamente um Diagrama de Sequência num Diagrama de Colaboração

Diagrama de colaboração
reserva de quarto de hotel
(com aninhamento de mensagens)

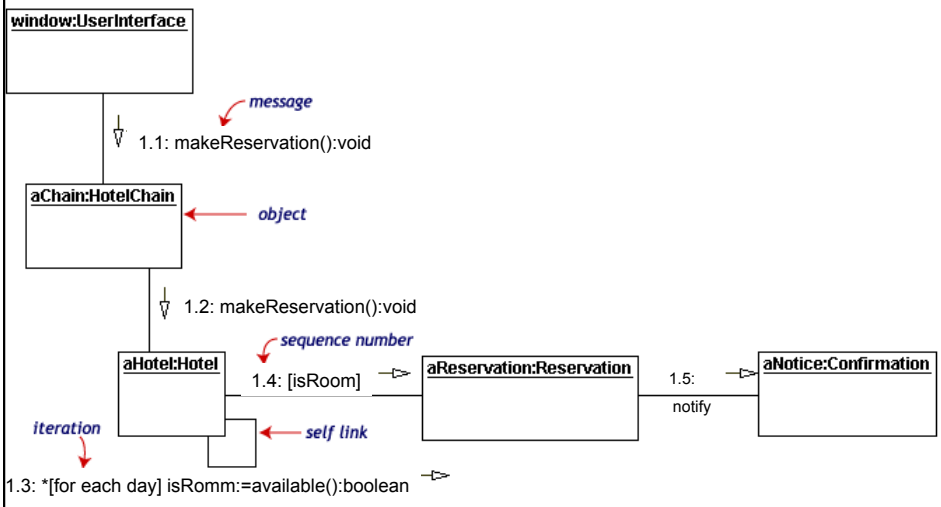
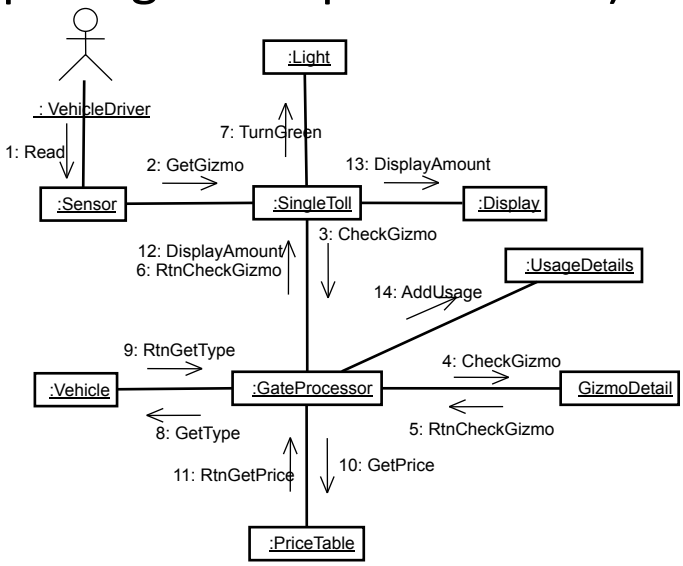


Diagrama de colaboração (portagem de ponto único)



Colaboração vs Sequência

- Os diagramas de sequência
  - lêem-se e compreendem-se rapidamente
  - a ordem das mensagens é clara e muito intuitiva
  - oferecem estruturas próprias para representar ciclos, concorrência, alternativas, etc
  - mas requerem mais disciplina na sua construção
- Os diagramas de colaboração
  - exigem menos disciplina na sua construção (não precisamos de conhecer o local certo onde o objecto deve ficar)
  - a ordem das mensagens pode ser adiconada à posteriori
  - mas são mais pobres nas estruturas que oferecem