

UML para modelagem de dados

Prof. Dr. Luis Mariano del Val Cura

UML

- UML foi projetado para modelagem de software.

mas.

- Diagramas de classes de UML tem sido utilizados como linguagem de modelagem de Dados.

UML vs E/R

- Conjuntos Entidade : Modelados como Classes Caixas com a lista de atributos. Diferentemente dos retângulos e ovais dos diagramas E/R.
- Relacionamentos binários como linhas entre conjuntos entidade. O nome do relacionamento aparece associados a essa linha.
- Relacionamentos com atributos: Modeladas como Classes de associação conectada à linha do relacionamento, com o nome do relacionamento e os atributos do mesmo .

Classes

- **Conjuntos Entidade** : Modelados como Classes.
- Classes: Conjuntos de objetos, com atributos e métodos.
 - PK indica um atributo como chave primária.
 - Na modelagem orientada a objetos é opcional

Exemplo: Empregados

Empregados
PK RG: string Nome: string
setNome(n) getNome() : string

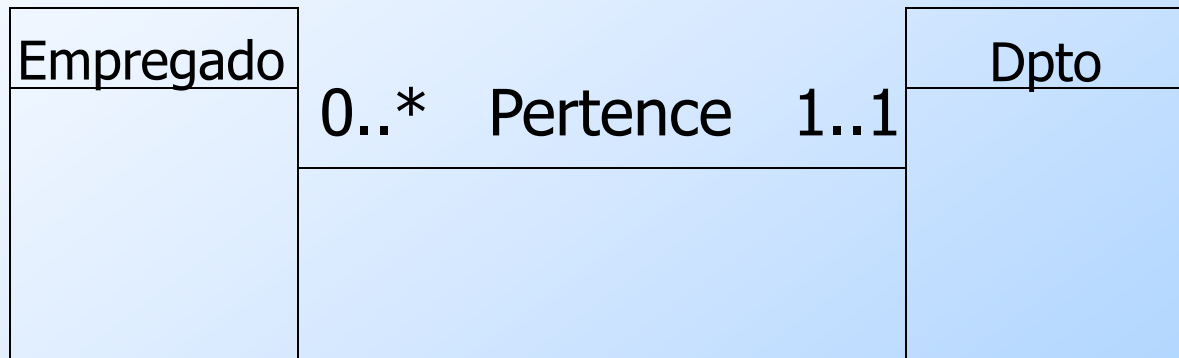
Associações

- ❑ **Relacionamentos binários** : Modelados como associações.
- ❑ Representados como linhas entre as classes (diferente dos losangos em E/R).
- ❑ Cardinalidade dos relacionamentos.
 - ❑ $m .. n$ notação mínimo e máximo.
 - ❑ $*$ = "infinito"; e.g. $1..*$ significa ao menos um
 - ❑ O nome do relacionamento aparece associado a essa linha.
 - ❑ Papeis são colocados adjacentes aos conjuntos entidade.

Associações

- A cardinalidade é colocada exatamente na posição contrária aos diagramas E/R.
- As restrições $0..*$ do lado $E2$ e $0..1$ do lado $E1$ significam que cada entidade $E2$ pode participar no máximo em um relacionamento e que $E1$ pode participar em muitos relacionamentos, i.e. o relacionamento é muitos para um de $E2$ para $E1$.
- Valores isolados como 1 ou * podem ser colocados, sendo interpretados como $1..1$, e $0..*$.

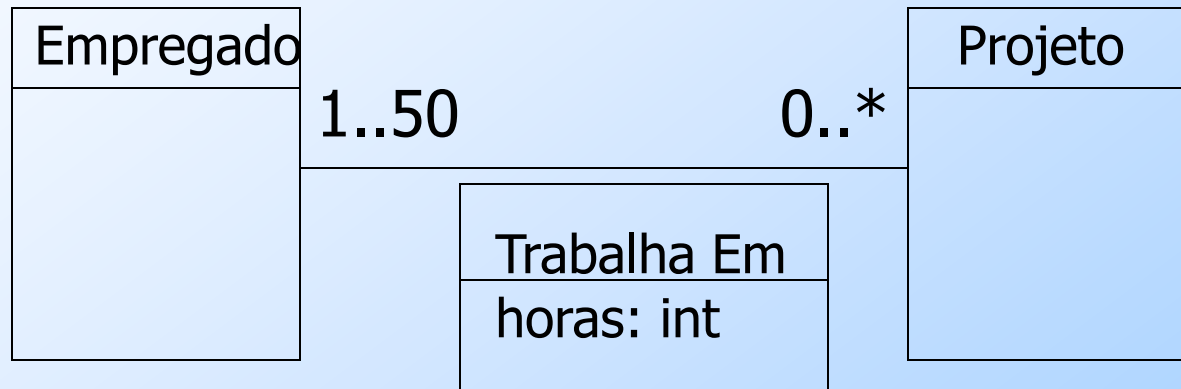
Exemplo: Associação



Classes de Associação

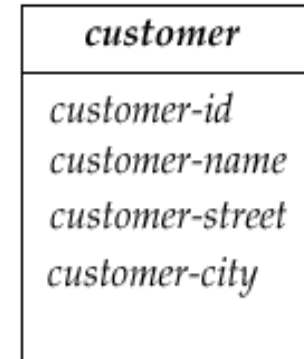
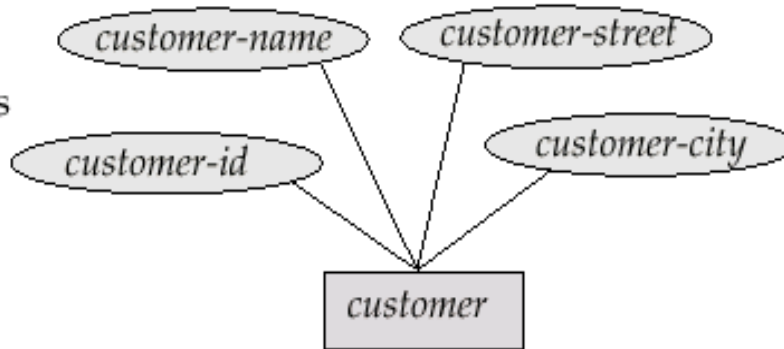
- ❑ **Relacionamentos com atributos:**
Modeladas como Classes de associação.
- ❑ Atributos associados a uma associação

Exemplo: Classe de Associação

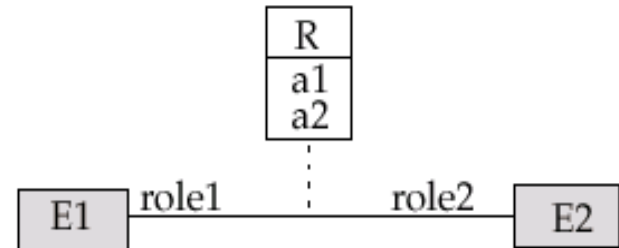
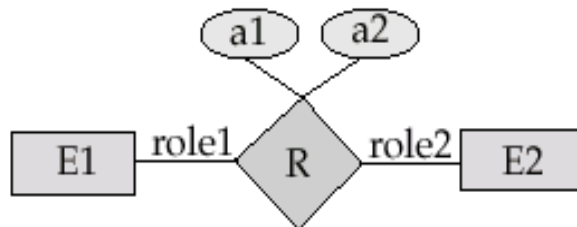
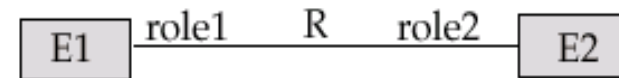
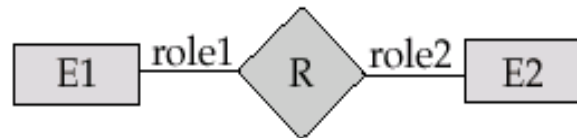


UML vs E/R

1. Entity sets and attributes



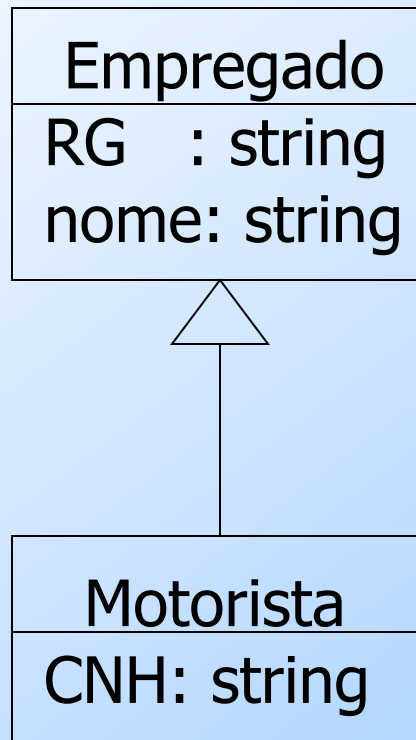
2. Relationships



Subclasses

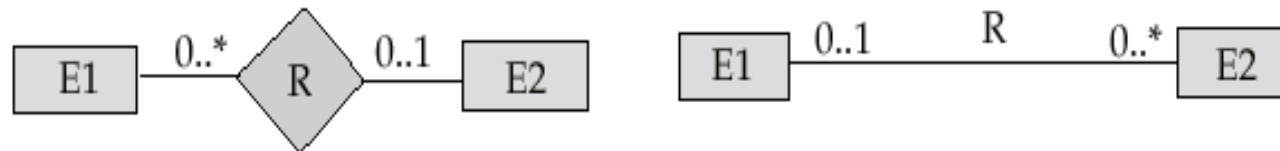
- **Especialização:** É modelada de forma similar em UML.
- Diferenças de notação para sobreposição e disjunção

Exemplo: Subclasses

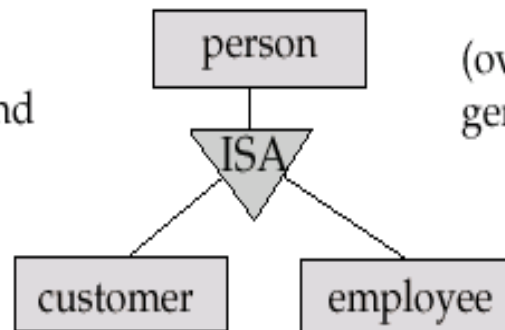


UML vs E/R

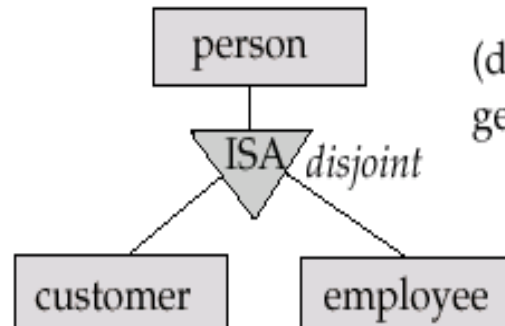
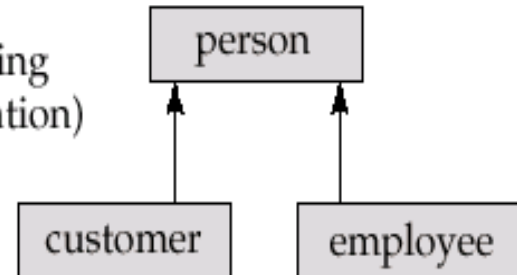
3. Cardinality constraints



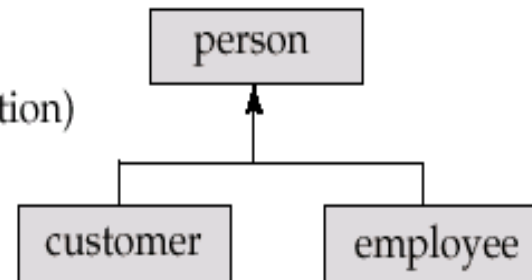
4. Generalization and Specialization



(overlapping generalization)



(disjoint generalization)

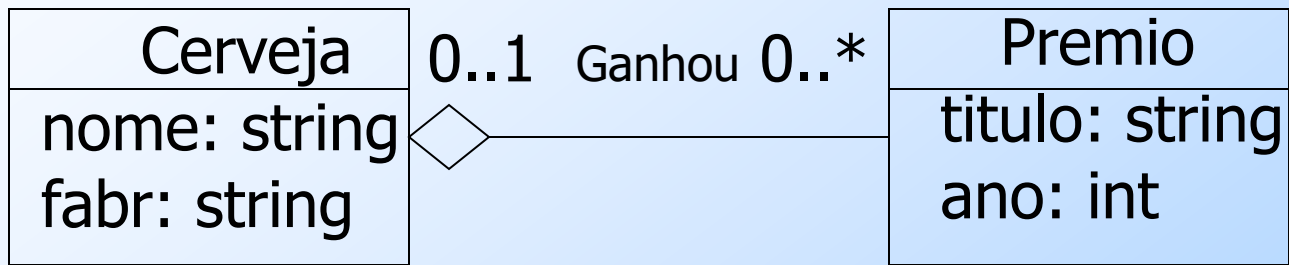


Diferenças UML E/R

Agregação

- Relacionamentos com o significado de que o objeto em um lado faz parte do objeto do outro lado
- Representado por um losango no final da linha de relacionamento.
 - Implica no modelo relacional que objetos são parte das tuplas relacionadas.

Exemplo: Agregação

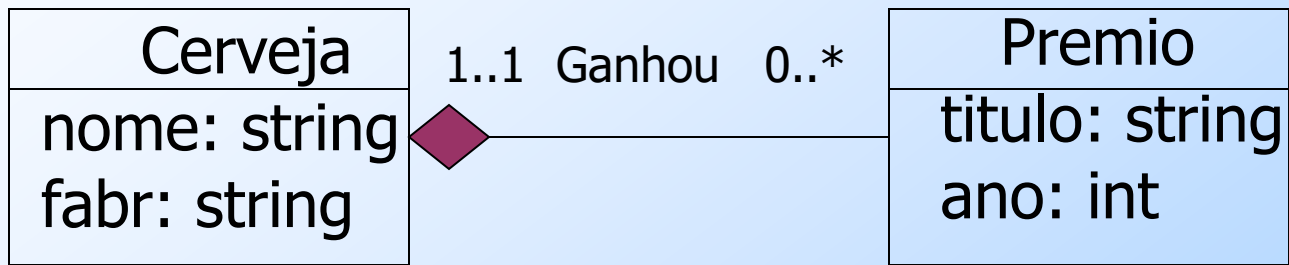


Diferenças UML E/R

Composição

- Similar a agregação com a implicação de que o objeto é necessariamente propriedade ou parte de um objeto do outro lado do relacionamento.
- Representado por um losango escuro

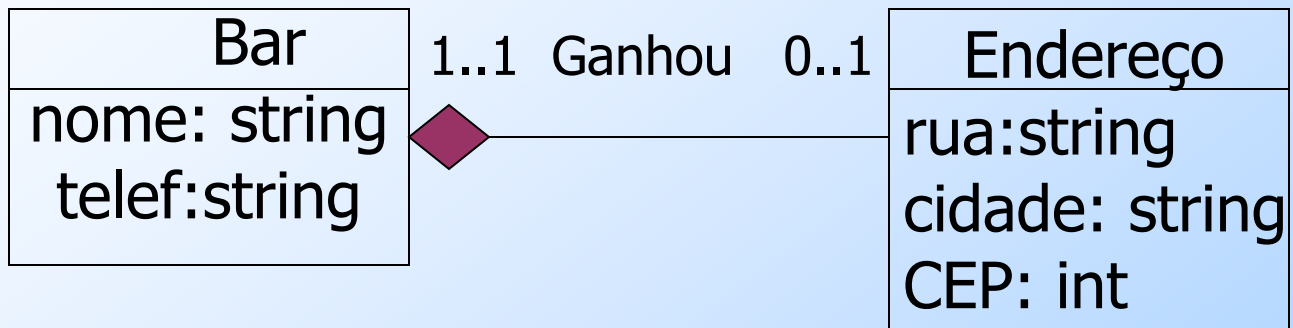
Exemplo: Composição



Diferenças. Entidade Fraca

- Não existe um conceito similar a entidade fraca em UML.
 - Em OO a chave primária é opcional. A identidade de um objeto não é definida pela chave primária.
- Pode ser modelado como composição

Example: Composition



Relacionamentos múltiplos

- Não existem em UML
- Precisam ser modelados como relacionamentos binários.

Relacionamentos múltiplos

- Um novo conjunto entidade relacionado com os outros por relacionamentos binários

