

OTES12 – Tópicos Avançados em Engenharia de Software

**Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – DCC**

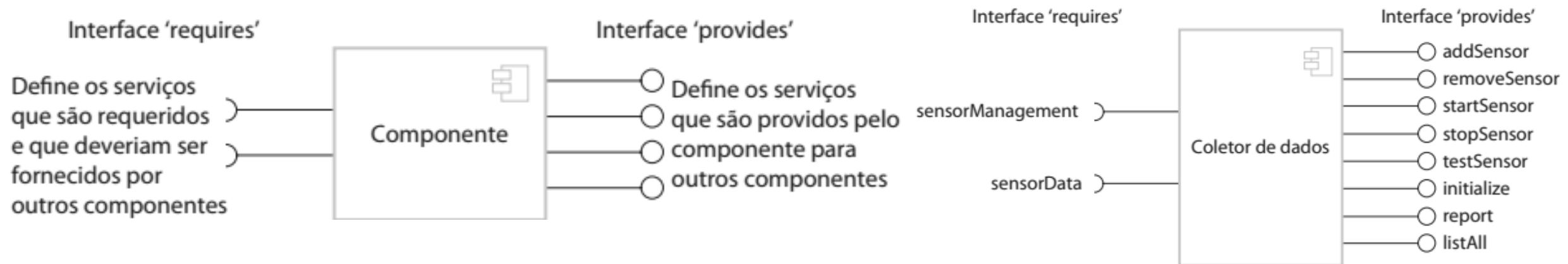
Prof. Dr. William Alberto Cruz Castañeda

2020/2

[Engenharia de Software baseada em Componentes]

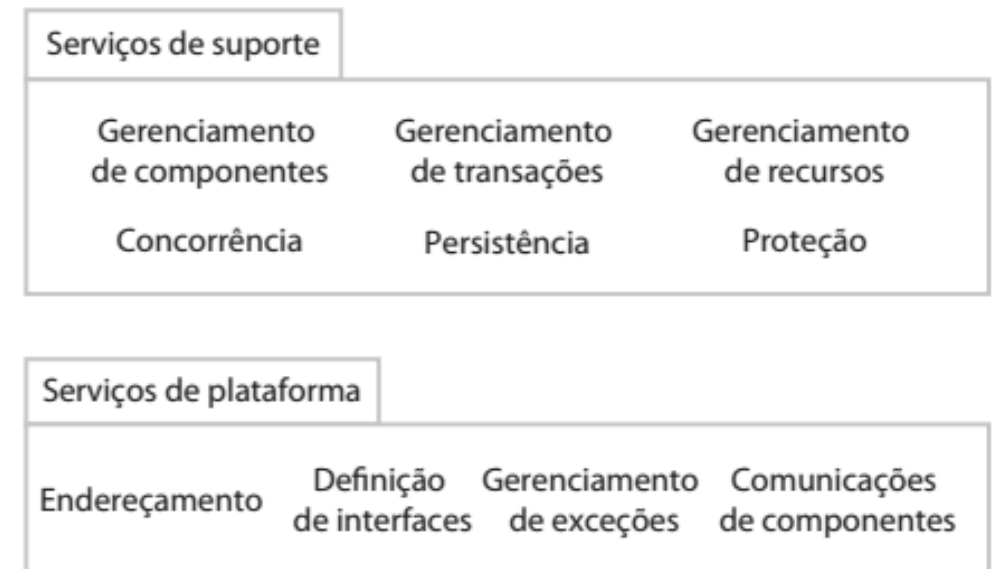
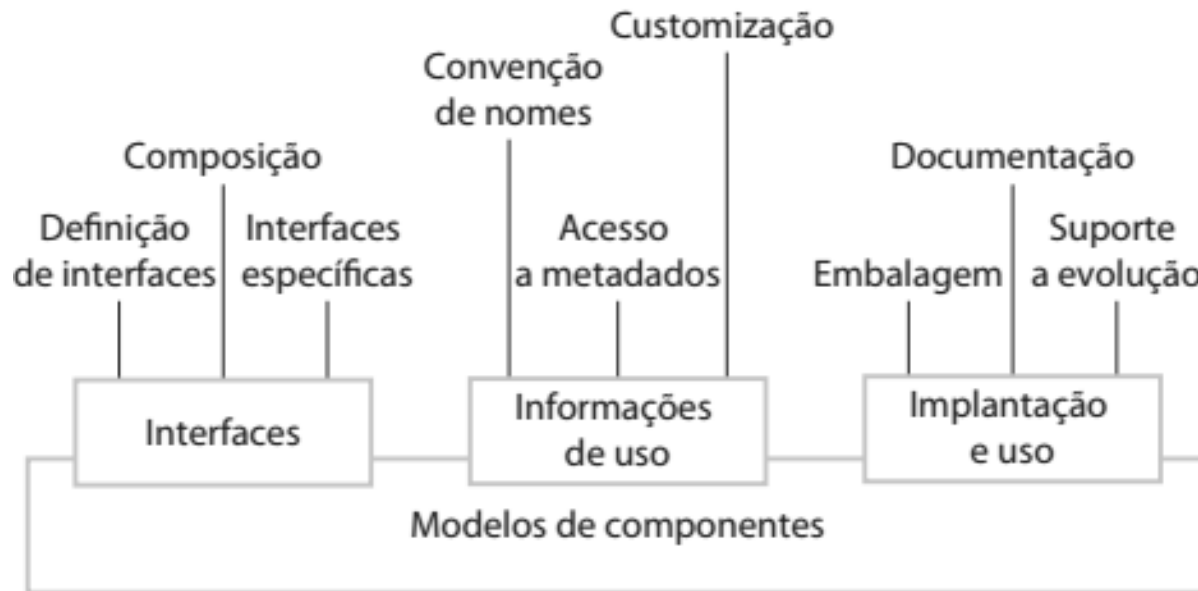
Componente de software

Unidade de composição com interfaces contratualmente especificadas e apenas dependências de contexto explícitas. Um componente de software pode ser implantado de forma independente e está sujeito a ser composto por parte de terceiros.



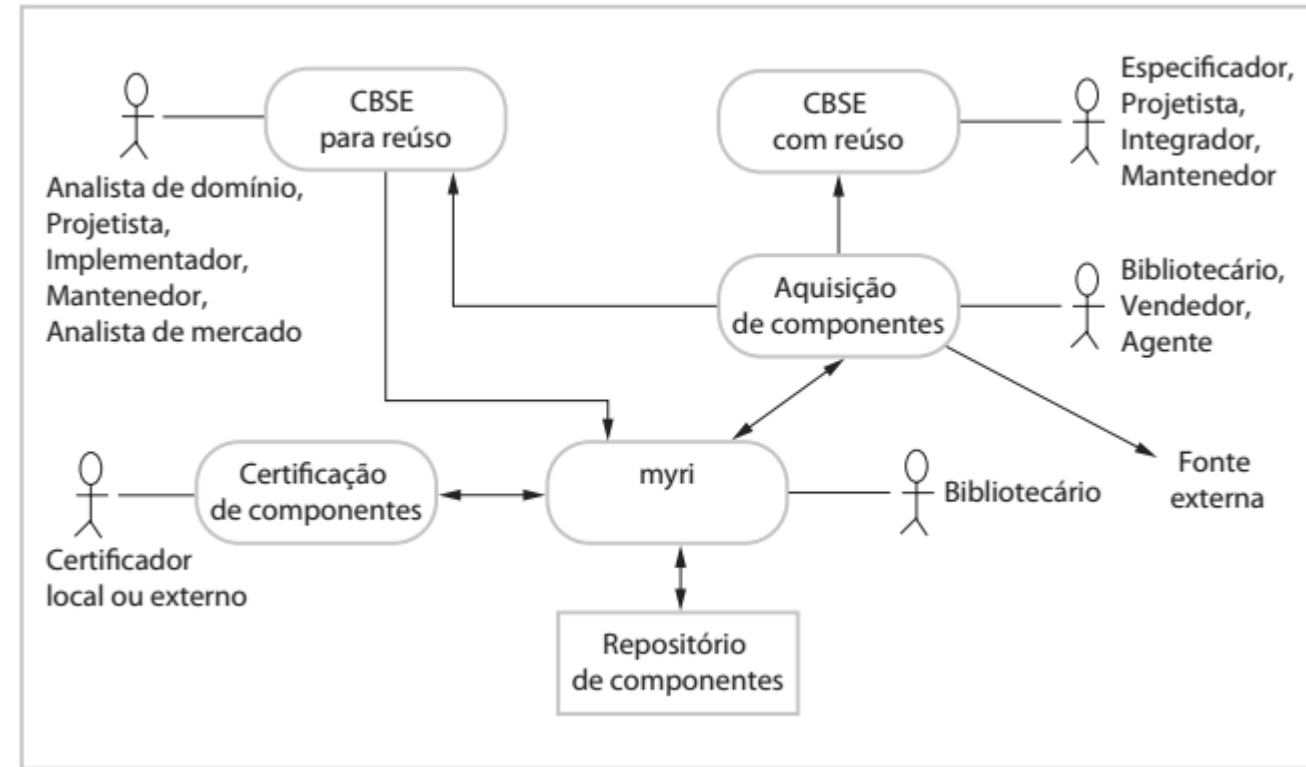
Modelos de Componentes

- Definição de normas para a implementação, documentação e implantação de componentes.
- Normas servem para garantir que os componentes podem interoperar.



Processo de Engenharia de Software baseada em Componentes

- 1. Desenvolvimento para reuso** – processo interessado no desenvolvimento de componentes ou serviços que serão reusados em outras aplicações. Envolve generalizar os componentes existentes.
- 2. Desenvolvimento com reuso** – processo de desenvolvimento de novas aplicações usando componentes e serviços existentes.

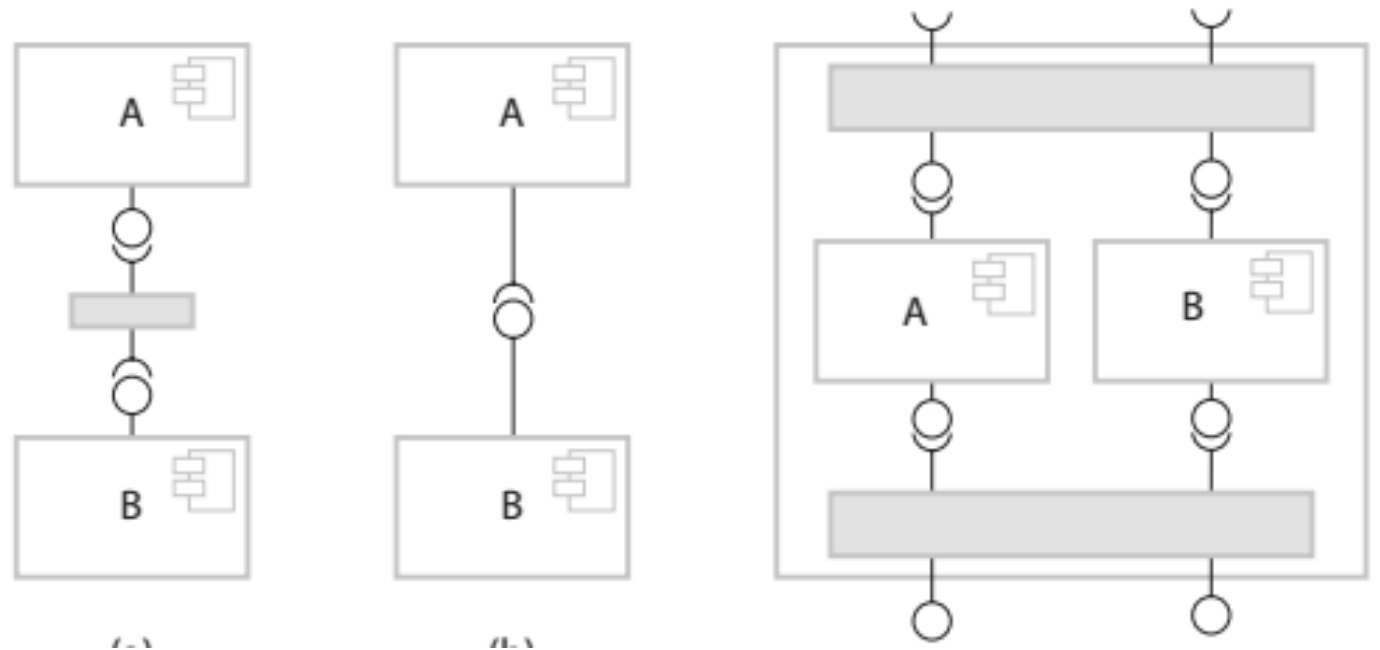


Composição de Componentes

Processo de integração de componentes uns com os outros e com o componente especialmente escrito “*glue code*” para criar um sistema ou outro componente

Podem ocorrer três tipos de incompatibilidades:

1. **Incompatibilidade de parâmetro;**
2. **Incompatibilidade de operação;**
3. **Incompletude de operação;**



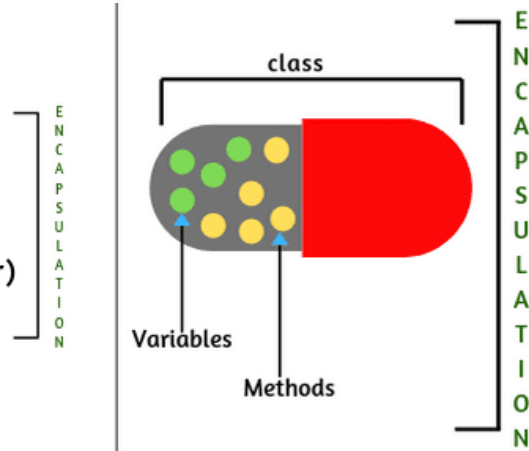
Exercícios:

- Projete a interface *provides* e a interface *requires* de um componente reusável que possa ser usado para representar um aluno em sistema de registro escolar de gestão de alunos.
- Projete as interfaces de componentes que podem ser usadas em um sistema para uma sala de controle de emergência. Você deve projetar interfaces para um componente de registro de chamadas que registre as chamadas realizadas e um componente de descoberta do veículo que, dado um código postal (CEP) e um incidente, encontre o veículo adequado mais próximo a ser despachado para o incidente.

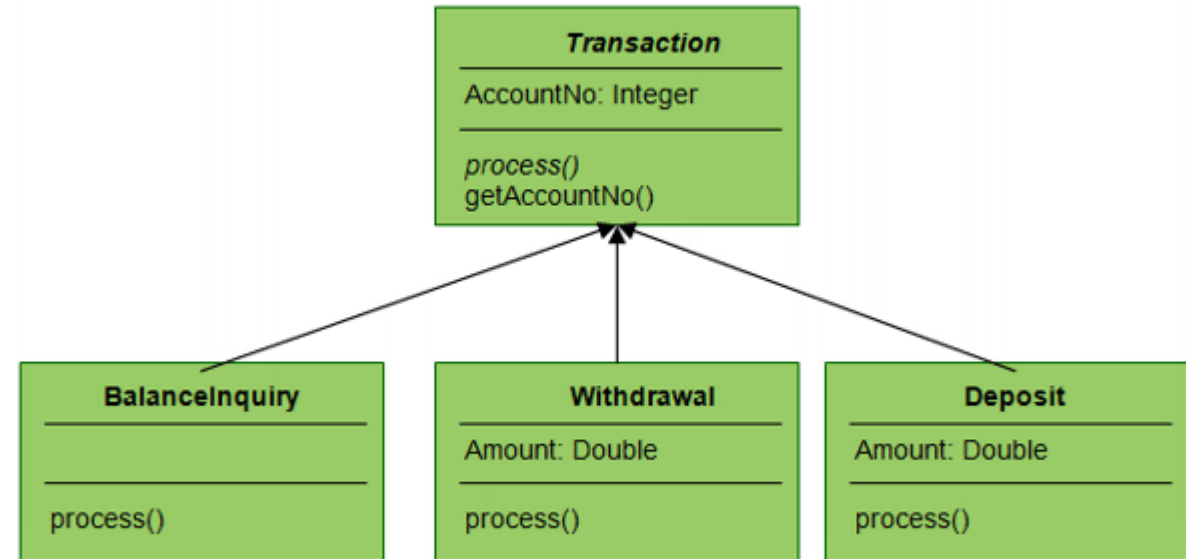
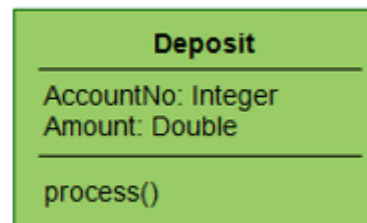
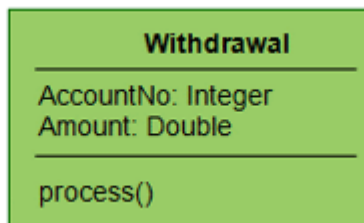
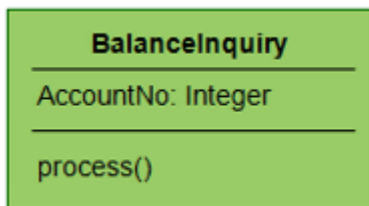
[Estratégias de Reuso em Projetos de Software]

Encapsulamento

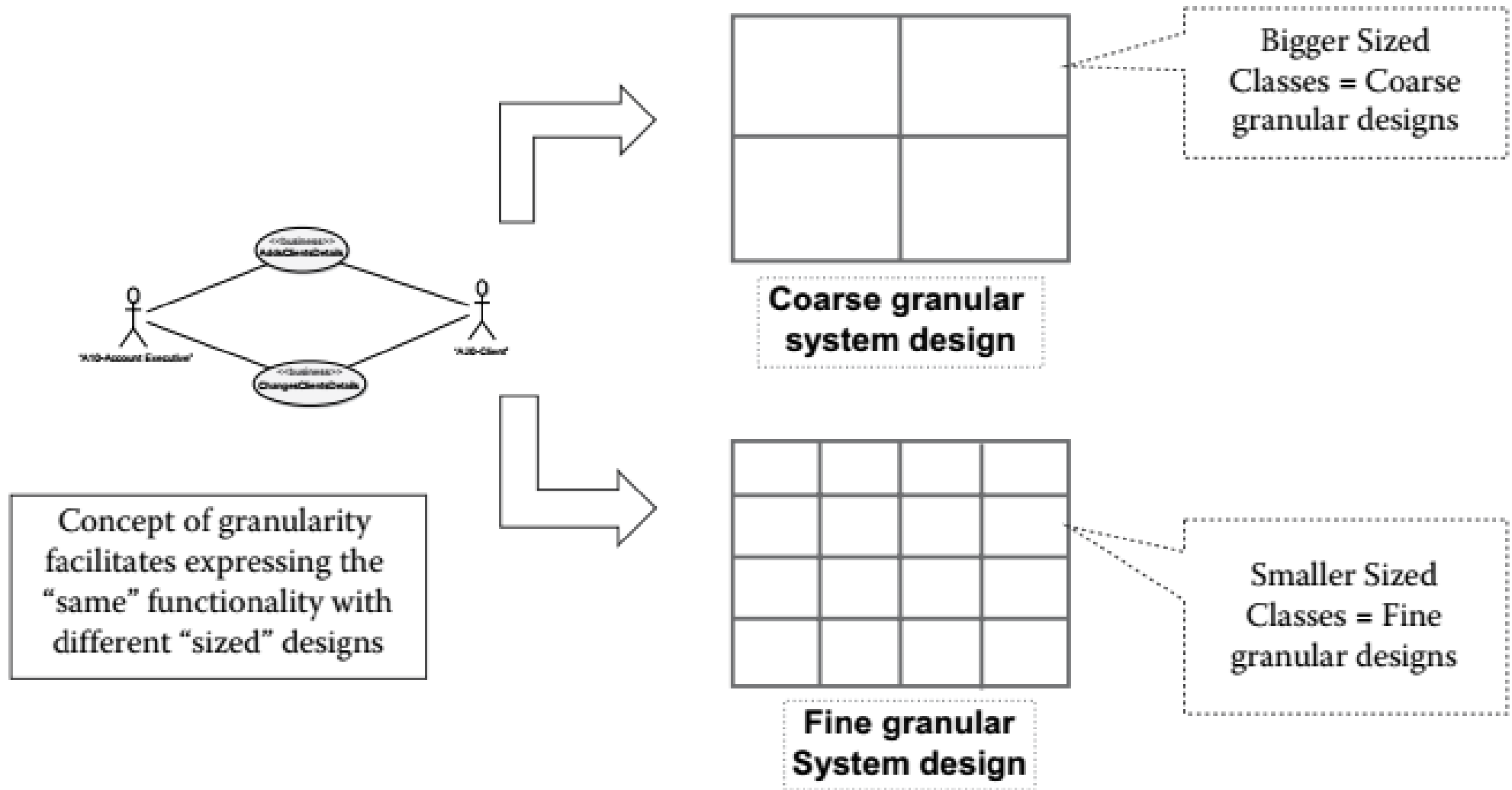
```
class
{
    data members
    +
    methods (behavior)
}
```



Generalização



Granularidade

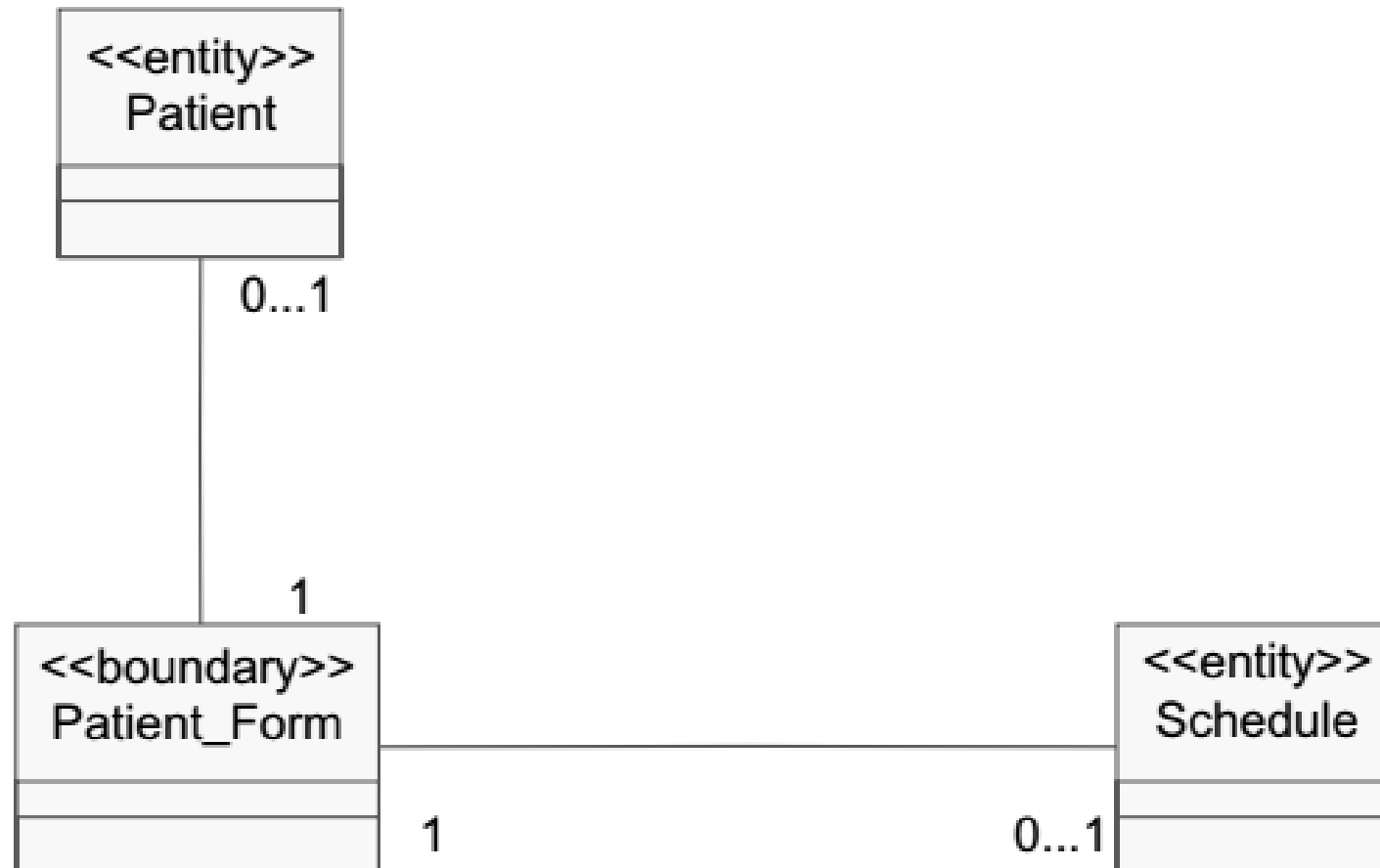


Robustez

- Abordagem que garante a capacidade de uma parte de uma solução mudar sem afetar o restante do sistema.
- Maior a dependência entre as classes → menos robusto;
- Menor dependência entre as classes → o sistema é considerado robusto;



Identificando a falta de robustez



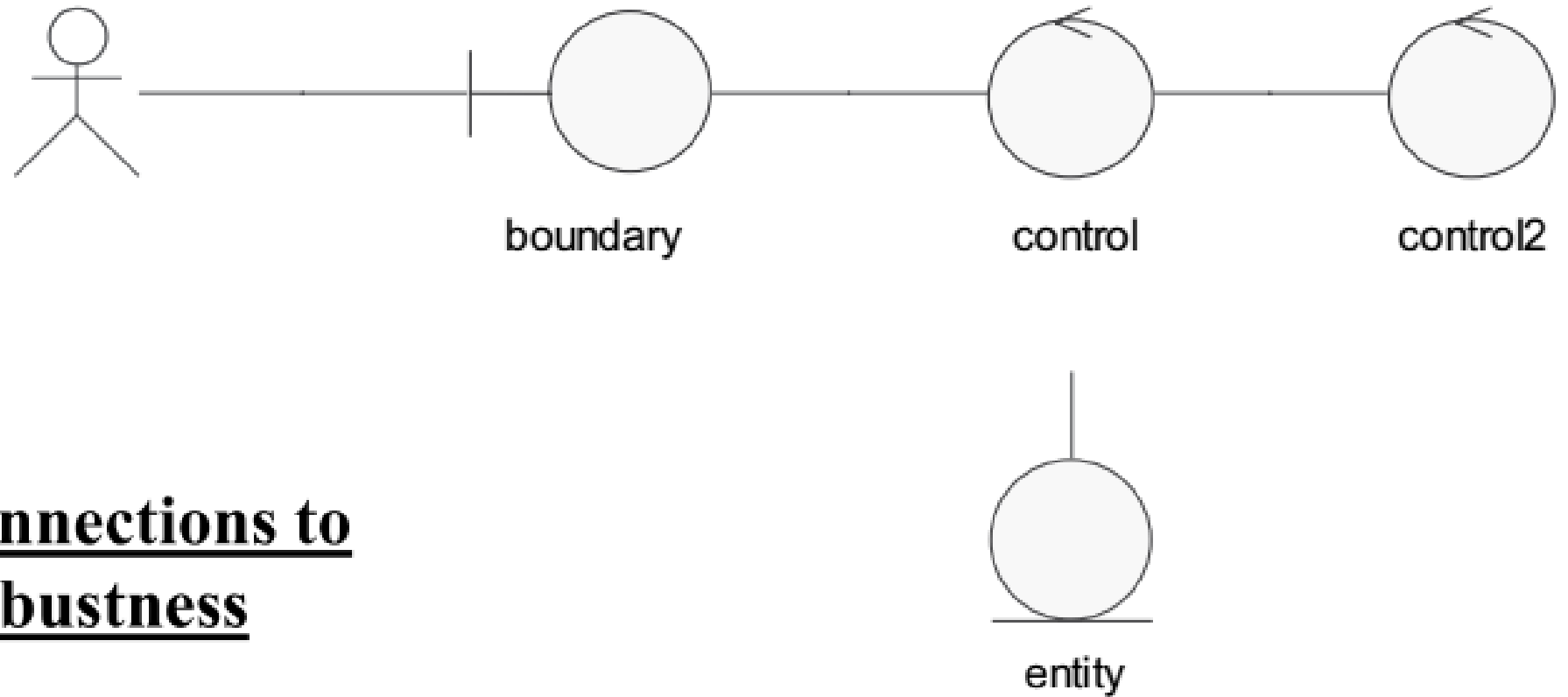
Regras

Ao aplicar robustez, as seguintes regras são incorporadas nos projetos:

- Classes **Boundary** não podem associar-se umas às outras;
- Classes **Entity** não podem associar-se umas às outras;
- Classes **Boundary** e **Entity** podem associar-se com classes **Control**;
- Classes **Control** podem associar-se umas com as outras;

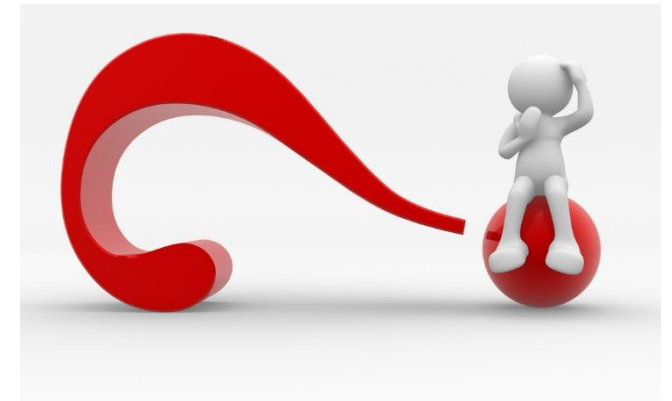
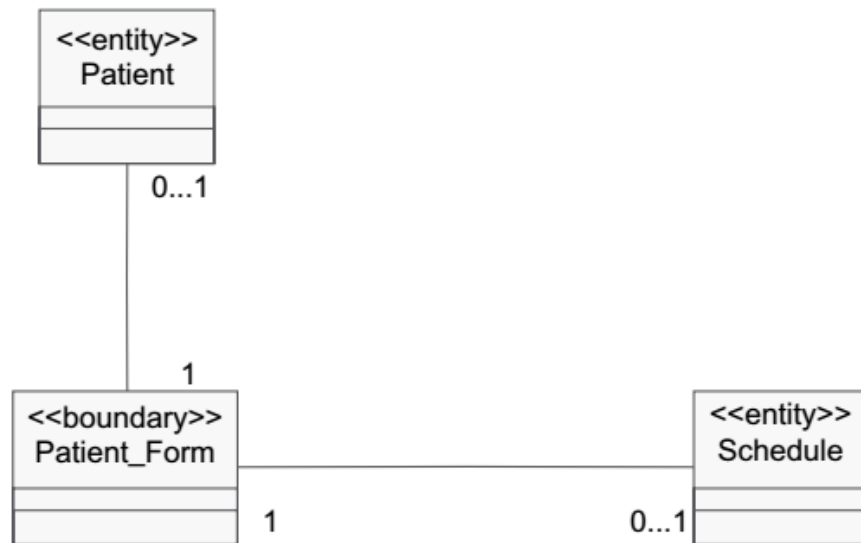


Estendidas para o relacionamento << table >> e << entity >>



Allowable Connections to ensure Robustness

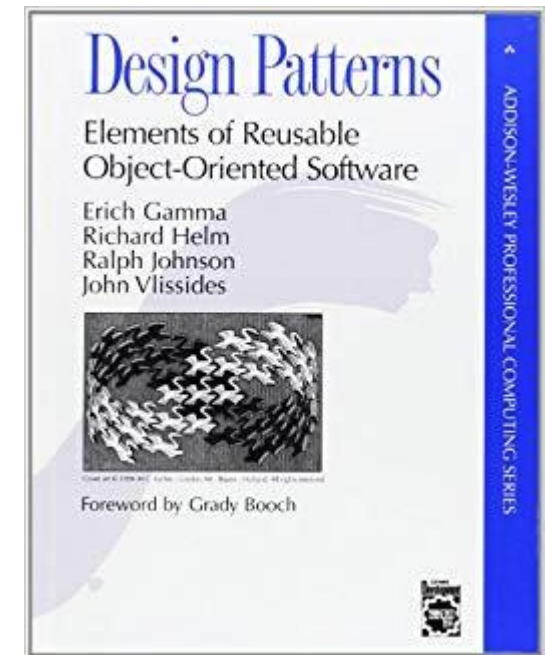
Incorporando Robustez

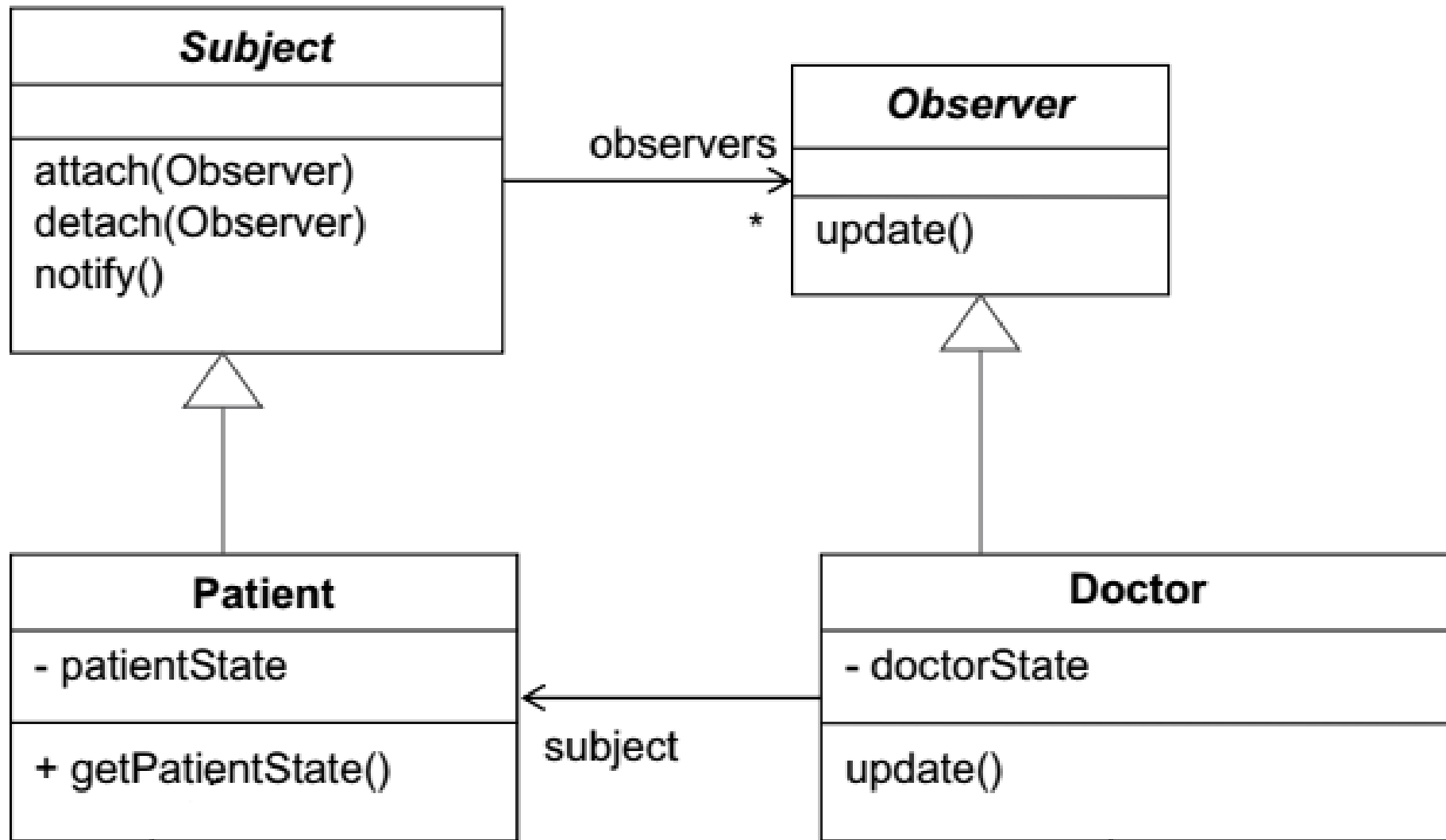


Padrões de Projeto

Descrito pelos seguintes quatro elementos essenciais:

- Nome do padrão – descreve o problema, suas soluções e consequências;
- Descreve quando aplicar o padrão;
- Descreve os elementos que compõem o design, seus relacionamentos, responsabilidades e colaborações;
- Consequências são os resultados da aplicação do padrão;





Exercício Padrão de Projetos

OTES12 – Tópicos Avançados em Engenharia de Software

**Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – DCC**

Prof. Dr. William Alberto Cruz Castañeda

2020/2