
Engenharia de Software

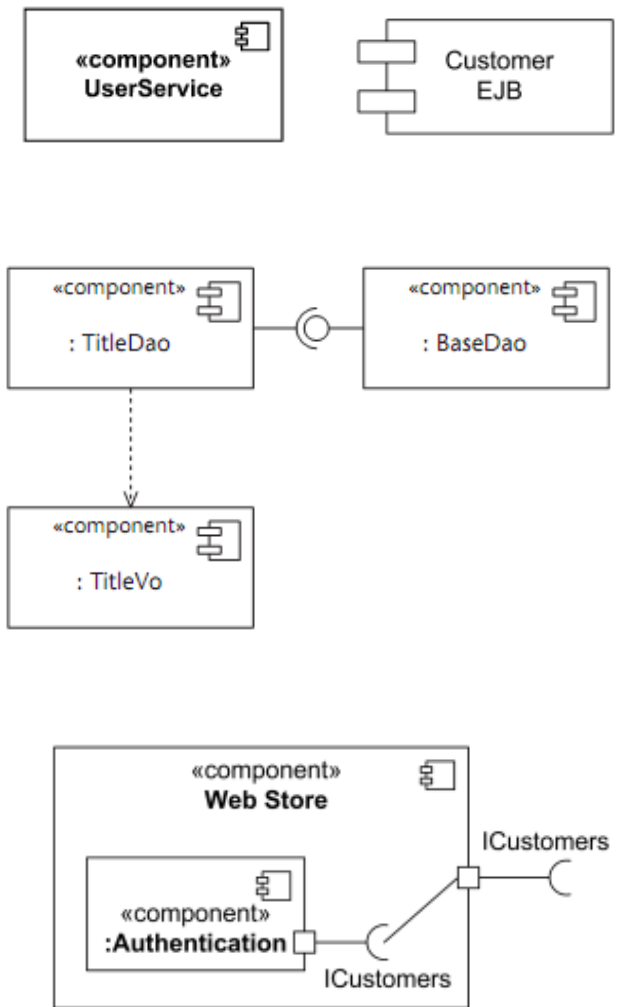
Linguagem UML

Luís Morgado

Instituto Superior de Engenharia de Lisboa
Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

Diagramas de Componentes

- Organização lógica e física
 - Descrição das partes reutilizáveis de um sistema (componentes) e das relações entre essas partes
- Componente
 - Módulo ou parte autónoma e encapsulada de um sistema que pode ser reutilizada ou substituída
 - Disponibiliza uma ou mais interfaces que permitem a interação entre componentes
 - Organização modular de um sistema
- Perspectivas de modelação
 - Perspectiva de caixa fechada (*black-box*)
 - Perspectiva de alto-nível com foco nos componentes e nas respectivas relações e configurações
 - Perspectiva de caixa aberta (*white-box*)
 - Perspectiva detalhada com foco na arquitectura interna dos componentes



Diagramas de Componentes

• Interfaces

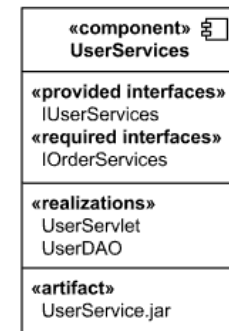
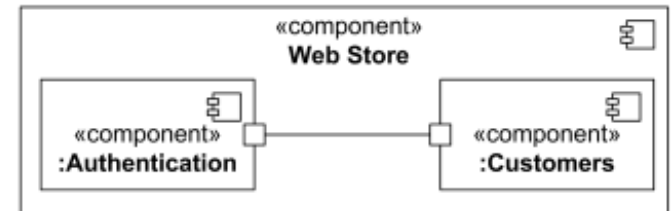
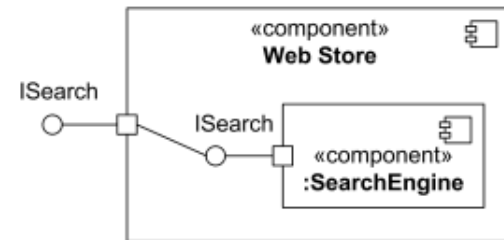
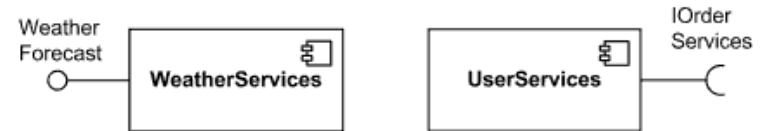
- Encapsulamento de funcionalidade
- Redução de acoplamento
 - Alterações internas num componente não se propagam a outros componentes
 - Controlo de dependências entre componentes
- **Disponibilizadas** (*provided*)
- **Requeridas** (*required*)

• Portos (*Ports*)

- Pontos de interacção com o exterior
 - Ligação a partes internas que implementam ou utilizam interfaces
 - Agrupamento de interfaces
- Conectores de delegação
 - Permitem relacionar interfaces com as partes internas que as implementam

• Compartimentos

- Forma de representação onde são indicados aspectos específicos ou artefactos necessários à disponibilização do componente



[uml-diagrams.org]

Diagramas de Componentes

Estereótipos de componentes

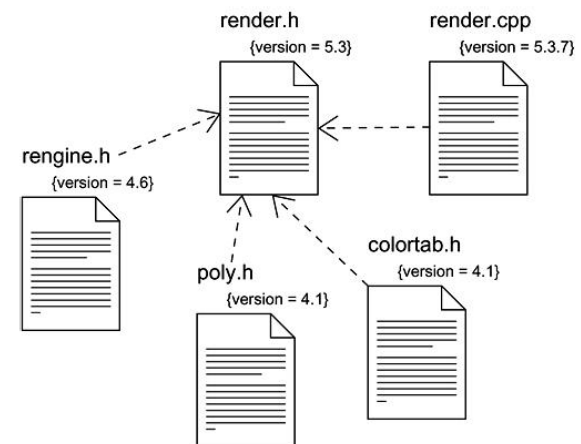
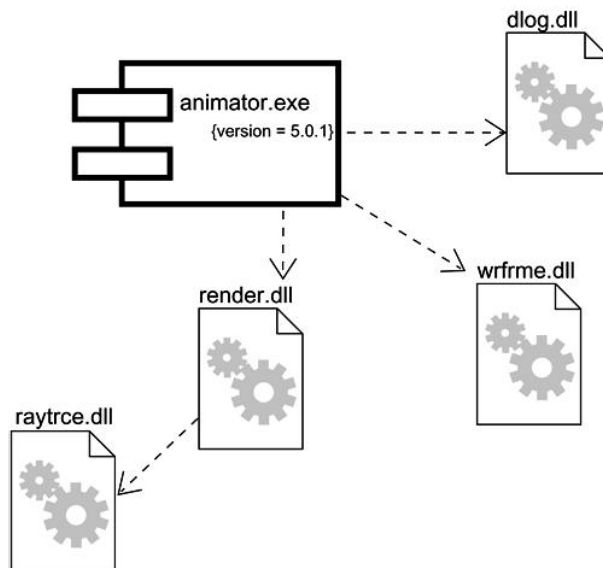
Entity	O componente representa um conceito do domínio do problema.
Specification	O componente representa uma especificação, inclui interfaces mas não as implementa.
Realization	O componente representa a realização de uma especificação. Surge em conjunto com o estereótipo «specification».
Process	Componente transaccional ou com fluxo autónomo de execução (<i>thread</i>).
Executable	Componente de software que pode ser executado num nó físico.
Subsystem	Parte de um sistema maior. É auto-contido e tipicamente maior que um componente isolado.
Service	Componente sem manutenção de estado.
Database	Base de dados.
Table	Tabela de uma base de dados.
Source code	Código fonte.
XML DTD	Regras sintáticas de um documento XML.

[UML User Guide, Booch *et al.* 1998]

Diagramas de Componentes

Utilização

- Modelação de arquitectura lógica e física
- Modelação de executáveis e bibliotecas
- Modelação de dados e documentos
- Modelação de serviços
- Modelação de código fonte
- Representação da relação entre modelos lógicos e físicos



[UML User Guide, Booch *et al.* 1998]

Diagramas de Implantação

• Perspectiva física do sistema

- Relacionam os artefactos de software com o hardware que os executa
 - Mostram uma perspectiva estática em termos da sua configuração de execução, localização física e formas de comunicação
 - Sistema descrito em termos de software e de hardware
 - Visualização da topologia de implantação do sistema
- Evoluem ao longo do ciclo de desenvolvimento
 - Identificação e exploração de dependências entre o sistema e outros sistemas no seu ambiente

• Artefactos

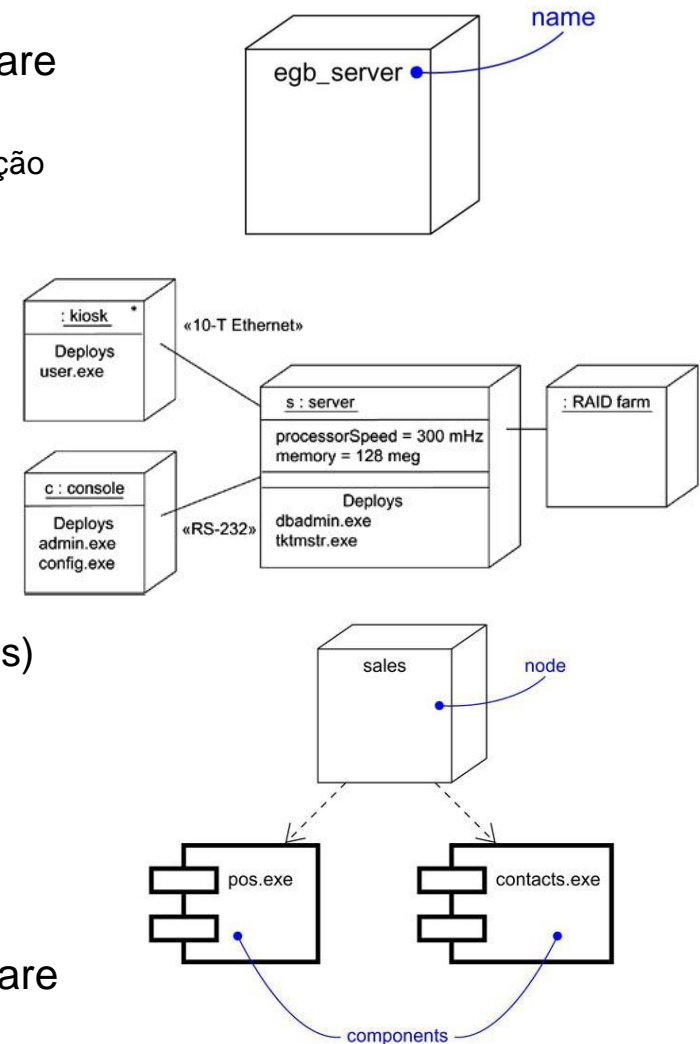
- Recursos físicos (ficheiros) que o sistema utiliza ou executa
- Tipos (podem corresponder a diferentes ficheiros físicos)

• Ligações

- Representam canais de comunicação
 - Estereótipos: HTTP, TCP/IP, RMI, JDBC, ODBC, RPC, etc.

• Nós

- Recursos físicos (entidades computacionais, hardware ou software) capazes de executar artefactos
 - Dispositivos (hardware de diferentes tipos)
 - Ambientes de execução (pode ser software, e.g. sistema operativo)



[UML User Guide, Booch *et al.* 1998]

Diagramas de Implantação

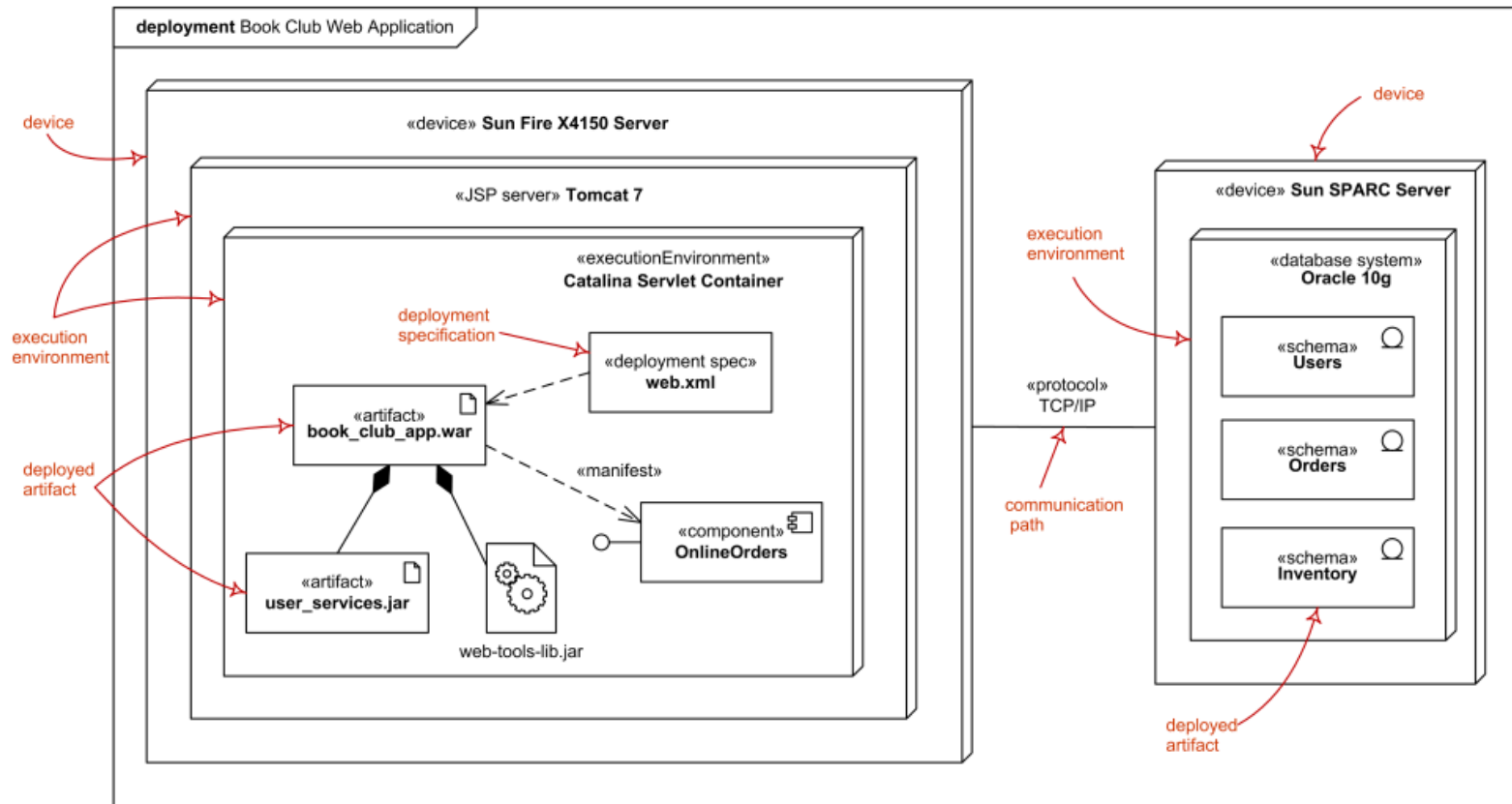
- Modelo de implantação

- Manifesto

- Descrição de configuração de componentes do sistema

- Especificação de implantação

- Tipo especial de artefacto para descrição parâmetros de execução (e.g. Informação de ligação)



[uml-diagrams.org]

Bibliografia

[Pressman, 2003]

R. Pressman, *Software Engineering: a Practitioner's Approach*, McGraw-Hill, 2003.

[Gamma et al., 1995]

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.

[Shaw & Garlan, 1996]

M. Shaw, D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.

[Vernon, 2013]

V. Vernon, *Implementing Domain Driven Design*, Addison-Wesley, 2013.

[Parnas, 1972]

D. Parnas, *On the Criteria to Be Used in Decomposing Systems into Modules*, Communications of the ACM 15-12, 1968.

[Kruchten, 1995]

F. Kruchten, *Architectural Blueprints - The "4+1" View Model of Software Architecture*, IEEE Software, 12-6, 1995.

[Burbeck, 1992]

S. Burbeck; *Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC)*, <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>, 1992

[Booch, 2004]

G. Booch, *Software Architecture*, IBM, 2004.