

## Estilos Arquiteturais

## Estilos Arquiteturais

- A arquitetura de um sistema pode aderir a um ou mais estilos arquiteturais
  - Um estilo define os tipos de elementos que podem aparecer em uma arquitetura e as regras que regem a sua interconexão
- Esses estilos podem simplificar o problema de definição de arquiteturas de sistema.
- A maioria dos sistemas de grande porte adere a vários estilos
- Estilos arquiteturais = “modelos arquiteturais”

## Modelos Arquiteturais

- São usados para documentar um projeto de arquitetura.
- Modelos estáticos de estrutura que mostram os principais componentes do sistema.
- Um modelo dinâmico de processo que mostra a estrutura de processo do sistema.
- Um modelo de interface que define as interfaces de subsistemas.
- Modelos de relacionamentos, tal como um modelo de fluxo de dados, que mostra os relacionamentos dos subsistemas.
- Um modelo de distribuição que mostra como subsistemas são distribuídos pelos computadores.

## Organização de sistema

- Reflete a estratégia básica que é usada para estruturar um sistema.
- Três estilos de organizações são amplamente usados:
  - O estilo de repositório de dados compartilhados;
  - Estilo de serviços e servidores compartilhados;
  - Estilo de máquina abstrata ou em camadas.

## Modelo de repositório

- Os subsistemas devem trocar dados. Isso pode ser feito de duas maneiras:
  - Os dados compartilhados são mantidos em um banco de dados central ou repositório e podem ser acessados por todos os subsistemas;
  - Cada subsistema mantém seu próprio banco de dados e passa dados explicitamente para outros subsistemas.
- Quando grandes quantidades de dados são compartilhadas, o modelo de repositório de compartilhamento é o mais usado.

## Arquitetura de conjunto de ferramentas CASE

Figura 11.2  
Arquitetura de um conjunto de ferramentas CASE integradas.



## Características de modelo de repositório

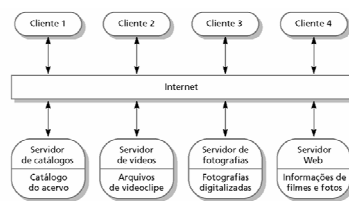
- Vantagens
  - É uma maneira eficiente de compartilhar grandes quantidades de dados;
  - Os subsistemas não necessitam saber como os dados são produzidos pelo gerenciamento centralizado, por exemplo, *backup*, proteção, etc.
  - Um modelo de compartilhamento é publicado como o esquema de repositório.
- Desvantagens
  - Os subsistemas devem estar de acordo com um modelo de dados do repositório. E, inevitavelmente, um compromisso;
  - A evolução de dados é difícil e dispendiosa;
  - Não há escopo para políticas específicas de gerenciamento;
  - Dificuldade para distribuir de forma eficiente.

## Modelo cliente-servidor

- É um modelo distribuído de sistema que mostra como dados e processamento são distribuídos por uma variedade de componentes.
- Estabelece servidores independentes que fornecem serviços específicos, tais como impressão, gerenciamento de dados, etc.
- Estabelece clientes que acessam esses serviços.
- Clientes e servidores normalmente se comunicam através de uma rede
  - Diversas tecnologias de comunicação são possíveis

## Biblioteca de filmes e fotografias

**Figura 11.3**  
Arquitetura de um sistema de acervo de filmes e fotografias.



## Características de cliente-servidor

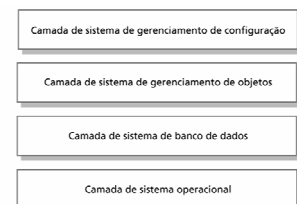
- Vantagens
  - A distribuição de dados é direta;
  - Faz uso efetivo dos sistemas em rede. Pode solicitar hardware mais barato;
  - É fácil adicionar novos servidores ou atualizar servidores existentes.
- Desvantagens
  - Nenhum modelo de dados compartilhado e, dessa forma, os subsistemas usam diferentes organizações de dados; por isso, a troca de dados pode ser ineficiente.
  - Gerenciamento redundante em cada servidor;
  - Nenhum registro central de nomes e serviços – pode ser difícil descobrir quais servidores e serviços estão disponíveis.

## Modelo de máquina abstrata (em camadas)

- Usado para modelar o interfaceamento dos subsistemas.
- Organiza o sistema em um conjunto de camadas (ou máquinas abstratas), cada uma das quais fornecendo um conjunto de serviços.
- Apóia o desenvolvimento incremental dos subsistemas em camadas diferentes. Quando uma camada de interface muda, somente a camada adjacente é afetada.
- Contudo, é freqüentemente artificial estruturar sistemas dessa maneira.
  - Modelos de camadas relaxados

## Sistema de gerenciamento de versões

**Figura 11.4**  
Modelo em camadas de um sistema de gerenciamento de versões.

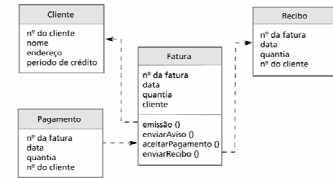


## Modelos de objetos

- Estruturar o sistema em um conjunto de objetos fracamente acoplados com interfaces bem definidas.
- A decomposição orientada a objetos está relacionada à identificação de classes de objetos, aos seus atributos e às suas operações.
- Quando implementados, os objetos são criados a partir dessas classes e um tipo de controle é usado para coordenar as operações de objetos.

## Sistema de processamento de faturas

Figura 11.5  
Modelo de objeto de um sistema de processamento de faturas.



## Vantagens do modelo de objetos

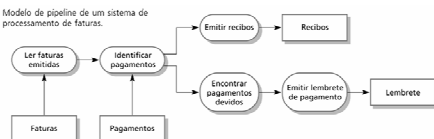
- Objetos não são firmemente acoplados e, desse modo, sua implementação pode ser modificada sem afetar outros objetos.
- Os objetos podem refletir entidades do mundo real.
- Linguagens de implementação orientada a objeto são amplamente usadas.
- Contudo, mudanças de interface de objeto podem causar problemas e entidades complexas podem ser difíceis de representar como objetos.

## Pipelining orientado a funções

- Transformações funcionais processam suas entradas para produzir saídas.
- Pode ser chamado de estilo de duto (pipe) e filtro (como no *shell* UNIX)
- Variações dessa abordagem são muito comuns. Quando as transformações são sequenciais, isso é um modelo sequencial em lotes, que é extensivamente usado em sistemas de processamento de dados.
- Não é realmente adequado para sistemas interativos.

## Sistema de processamento de faturas

Figura 11.6  
Modelo de pipeline de um sistema de processamento de faturas.



## Vantagens do modelo de *pipeline*

- Apóia reuso de transformações.
- Organização intuitiva para comunicação com stakeholders.
- É fácil adicionar novas transformações.
- É relativamente simples implementar como sistema concorrente ou sequencial.
- Contudo, requer um formato comum para a transferência de dados ao longo do pipeline e o apoio a interações baseadas em eventos é difícil.

## Modelos de controle

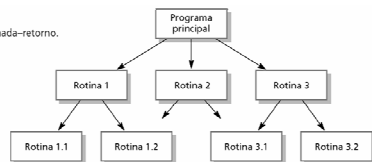
- Diferente do modelo de decomposição de sistema, os modelos de controle estão relacionados ao fluxo de controle entre subsistemas
- Controle centralizado
  - Um subsistema tem responsabilidade global pelo controle, e inicia e pára outros sistemas.
- Controle baseado em eventos
  - Cada subsistema pode responder a eventos gerados externamente a partir de outros subsistemas ou do ambiente do sistema.

## Controle centralizado

- Um subsistema de controle é responsável pelo gerenciamento da execução de outros subsistemas.
- Modelo chamada-retorno
  - É o modelo de subrotina *top-down* onde o controle inicia no topo de uma hierarquia de subrotina, e se move para baixo da hierarquia. É aplicável a sistemas seqüenciais.
- Modelo de gerenciador
  - É aplicável a sistemas concorrentes. Um componente de sistema controla a parada, o início e a coordenação de outros processos de sistema. Pode ser implementado em sistemas seqüenciais como uma declaração 'Case'.

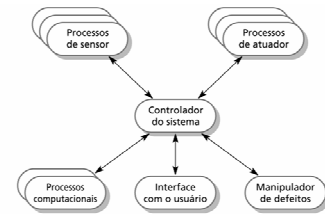
## Modelo chamada-retorno

**Figura 11.7**  
Modelo de controle chamada-retorno.



## Controle de sistema tempo real

**Figura 11.8**  
Modelo de controle centralizado para um sistema de tempo real.



## Sistemas orientados a eventos

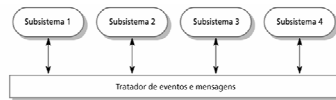
- Dirigidos por eventos gerados externamente onde o *timing* dos eventos está fora do controle dos subsistemas que processam o evento.
- Dois modelos dirigidos a eventos principais
  - Modelos *Publisher/Subscriber*. Um evento é transmitido a todos os subsistemas. Qualquer subsistema programado para manipular esse evento pode responder a ele.
  - Modelos orientados a interrupções. Usado em sistemas de tempo real onde as interrupções são detectadas por um tratador de interrupções e passadas por algum outro componente para processamento.
- Outros modelos dirigidos a eventos incluem sistemas de planilhas e de produção.

## Modelo Publisher/Subscriber

- É efetivo na integração de subsistemas em computadores diferentes em uma rede.
- Subsistemas registram um interesse em eventos específicos. Quando estes ocorrem, o controle é transferido para o subsistema que pode tratar o evento.
- A política de controle não é embutida no tratador de eventos e mensagens. Os subsistemas decidem sobre os eventos de seu interesse.
- Contudo, os subsistemas não sabem se um evento será tratado e nem quando será.

## Publiser/Subscriber

**Figura 11.9**  
Modelo de controle baseado em broadcast seletivo.

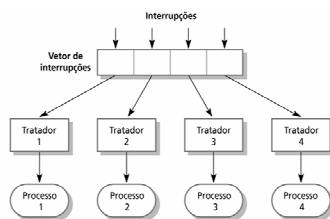


## Sistemas dirigidos a interrupções

- É usado em sistemas de tempo real onde a resposta rápida para um evento é essencial.
- Existem tipos de interrupções conhecidos com um tratador definido para cada tipo.
- Cada tipo é associado à uma localização da memória, e uma chave de hardware causa a transferência de controle para seu tratador.
- Permite respostas rápidas, mas é complexo para programar e difícil de validar.

## Controle dirigido a interrupções

**Figura 11.10**  
Modelo de controle orientado a interrupções.

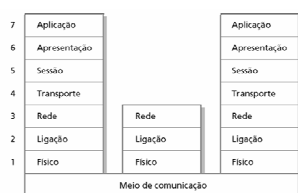


## Arquiteturas de referência

- Os modelos de referência são derivados de um estudo de domínio de aplicação ao invés de sistemas existentes.
- Pode ser usado como base para a implementação de sistemas ou comparar sistemas diferentes.
  - Atua como um padrão contra o qual os sistemas podem ser avaliados.
- O modelo OSI é um modelo de camadas para sistemas de comunicação.

## Modelo de referência OSI

**Figura 11.11**  
Arquitetura de modelo de referência OSI.



## Modelo CASE de referência

- Serviços de repositório de dados
  - Armazenamento e gerenciamento de itens de dados.
- Serviços de integração de dados
  - Gerenciamento de grupos de entidades.
- Serviços de gerenciamento de tarefas
  - Definição e aprovação de modelos de processo.
- Serviços de mensagens
  - Comunicação ferramenta-ferramenta e ferramenta-ambiente.
- Serviços de interface de usuário
  - Desenvolvimento de interface de usuário.

## O modelo de referência ECMA

**Figura 11.12**

Arquitetura de referência  
ECMA para ambientes CASE.

