Estruturas de controle

- O paradigma de controle externo da aplicação deve ser escolhido
  - O paradigma diz como uma aplicação é controlada "de fora" e como interage com seus usuários e outras aplicações
  - O controle interno dos objetos da aplicação não é visível fora da aplicação
- Há vários paradigmas de controle externo

Controle orientado a procedimento

- Tem um único thread no programa
- Métodos executam chamadas para obter dados de entrada e esperam os dados
- O estado do programa pode ser guardado em variáveis locais aos procedimentos (na pilha de execução)
- Vantagem: fácil de implementar com linguagens convencionais
- Desvantagem: qualquer paralelismo inerente aos objetos deve ser mapeado a um fluxo de controle seqüencial
- Freqüentemente usado em aplicações que possuem uma interface não gráfica

Controle orientado a evento

- Um dispatcher é provido pela linguagem, SGBD, linguagem ou sistema operacional
- Métodos da aplicação são amarrados a eventos
  - Quando o evento ocorre, o método correspondente é chamado
- Todos os métodos retornam o controle para o dispatcher em vez de esperar que os dados de entrada cheguem
- O estado do programa deve ser guardado em variáveis globais já que os métodos retornam o controle (e não ficam com ele)

- Vantagem: ajuda a separar a lógica da aplicação da interface com o usuário
- Freqüentemente usado com SGBD munido de linguagem de quarta geração
- Muito usado com bancos de dados
- Quase sempre usado para controlar uma interface gráfica

Controle concorrente

- O controle fica com vários objetos autônomos, cada um respondendo a eventos
- A concorrência é provida pelo sistema operacional
  - Usando paralelismo em hardware ou não
- Observe que estamos nos referindo a controle concorrente *dentro* da aplicação e não *entre* aplicações
  - Este último é muito comum (sessões paralelas de acesso a BD, por exemplo)
- Infrequentemente usado, embora seu uso tenda a crescer com o CORBA (framework de objetos distribuídos)

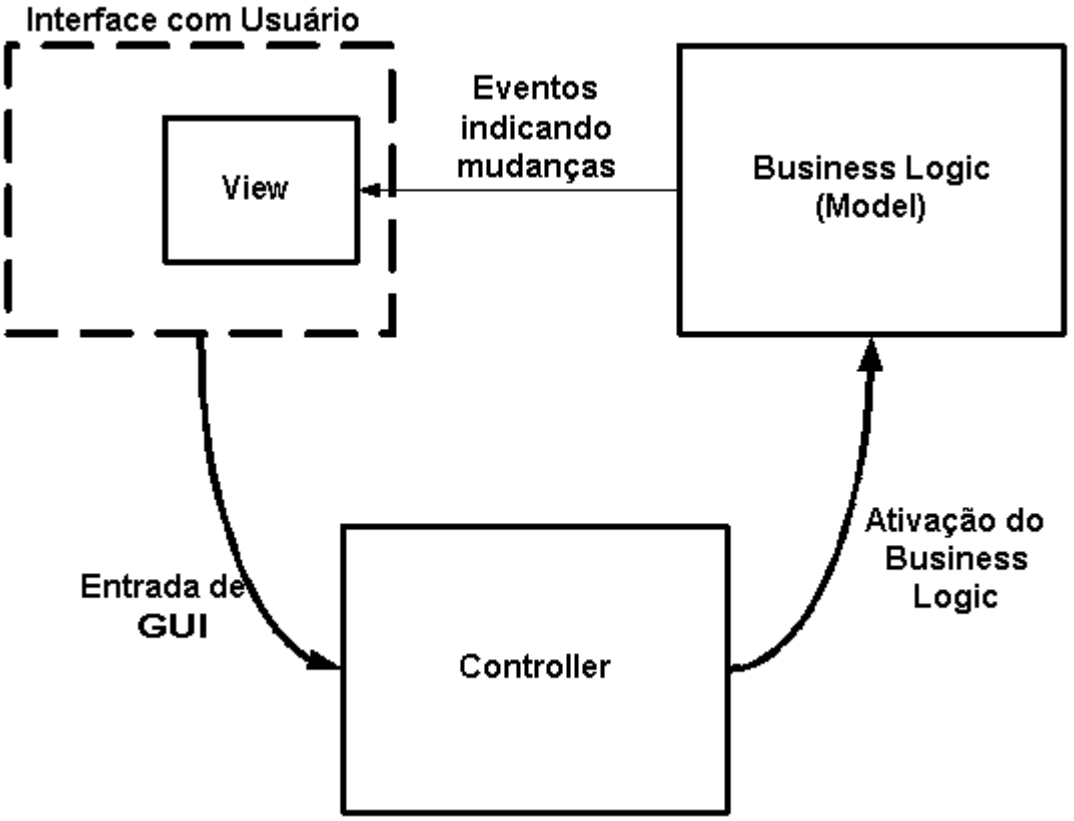
Controle declarativo

- O programa usa um fluxo de controle implícito baseado em statements declarativos
- Usado em sistemas especialistas baseados em regras

Padrões de controle

- Dois padrões de controle freqüentemente utilizados na arquitetura são:
  - Observer Pattern (ou Publish-Subscribe) para acoplar objetos ou subsistemas
    - Será descrito em outra seção
  - Model-View-Controller (MVC) para separar os objetos de domínio dos objetos de interface

- O Model é tudo que envolve o Business Logic, sem considerações de UI
- O View representa numa UI o estado corrente do Model
- O Controller é usado para alimentar o Model quando ocorrem entradas na UI
- Seré descrito em outra seção



programa