



Técnica de Projeto

Diagrama de Atividades




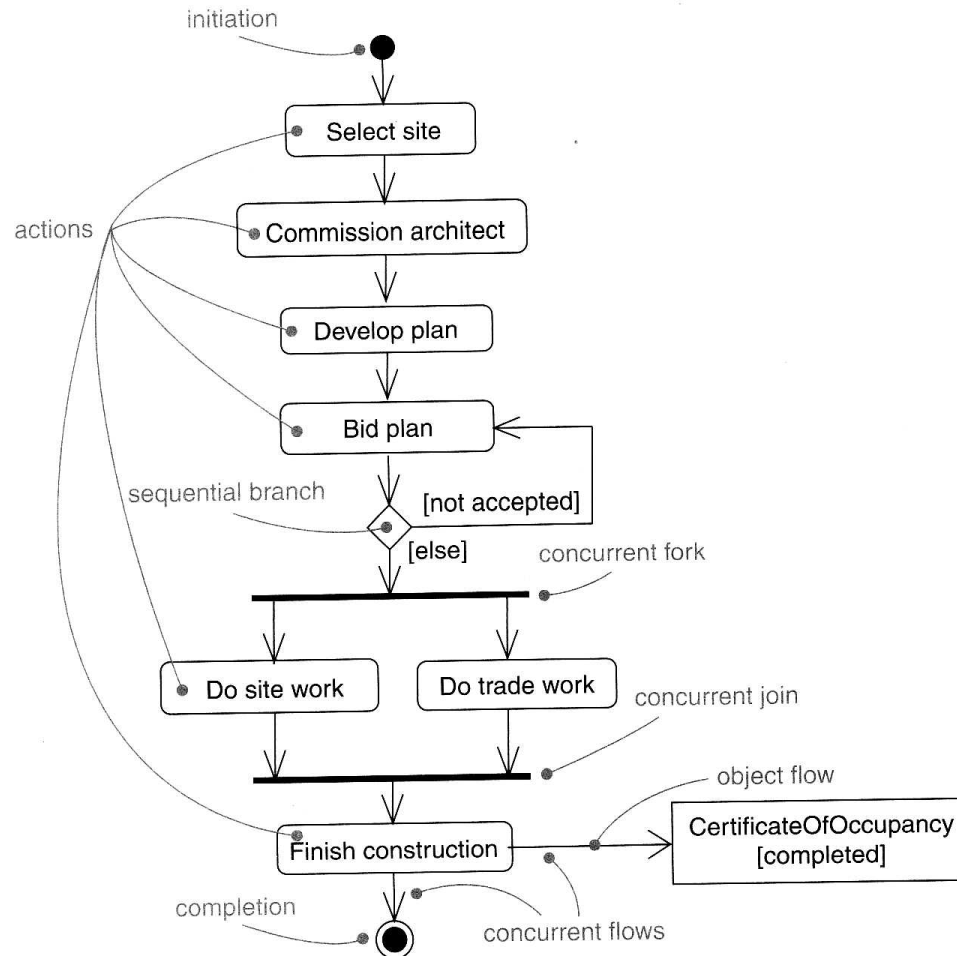


Diagrama de Atividades

- Usado para modelar os aspectos dinâmicos dos sistemas.
- Os diagramas de atividades geralmente são usados para modelar processos de negócios, workflows, fluxos de dados e algoritmos complexos.
- Um diagrama de atividade é essencialmente um flowchart (diagrama de fluxo) que enfatiza as atividades que são executadas, mostrando o fluxo de controle de atividade para atividade.
- Graficamente, um diagrama de atividade é uma coleção de nodos e arcos.

Diagrama de Atividades

→ Exemplo de modelagem de um processo de construção de uma casa





Notação

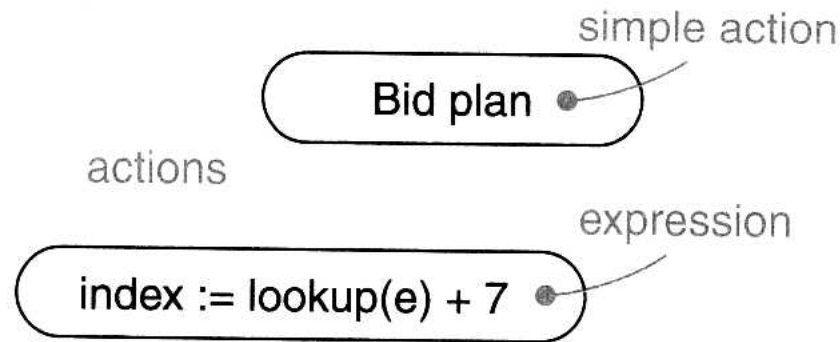
Diagramas de atividades contêm:

- Ações
- Nodos de Atividade
- Fluxos de Controle (seleção, forking e joining, raia)
- Fluxos de Objetos

Notação - Ação

AÇÕES: são computações atômicas.

Exemplos: avaliar alguma expressão que atribui o valor de um atributo ou que retorna algum valor; chamar uma operação de um objeto; enviar um sinal a um objeto; criar ou destruir um objeto; etc.



→ Uma ação é representada por uma caixa com cantos arredondados, contendo uma expressão

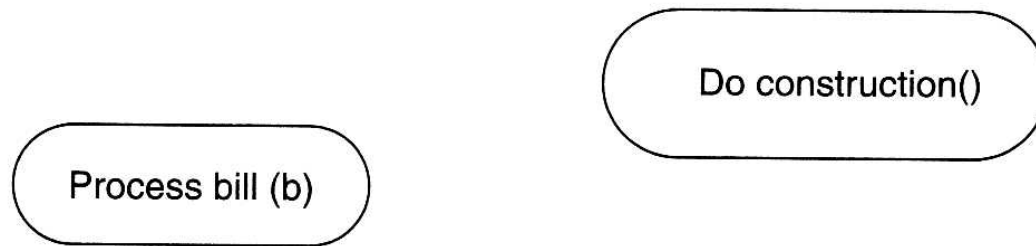
Ação

Características:

- Ações não podem ser decompostas.
 - ➔ Uma parte da ação não pode ser executada sozinha, ou toda a ação é executada ou nada é executado.
- A UML não define uma linguagem para expressar uma ação.
 - ➔ Pode ser usado um texto estruturado ou a sintaxe de uma linguagem de programação.

Notação - Fluxo de Atividade

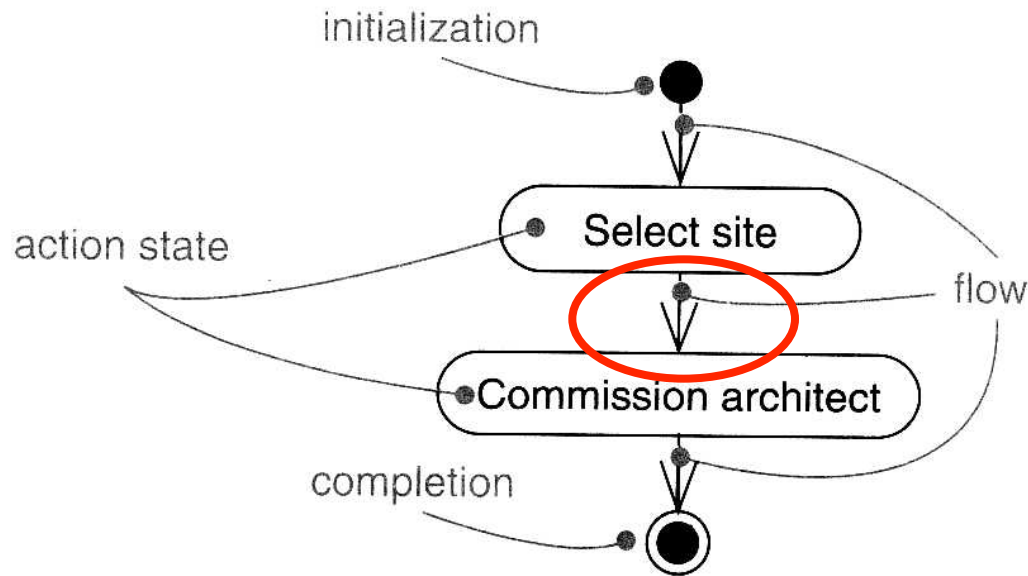
NODO DE ATIVIDADE: é o agrupamento aninhado de ações ou outros nodos de atividade.



- Não existe distinção na notação de ação e nodo de atividade.
- ➔ Uma ação pode ser vista como um caso especial de nodo de atividade.

Notação - Fluxo de Controle

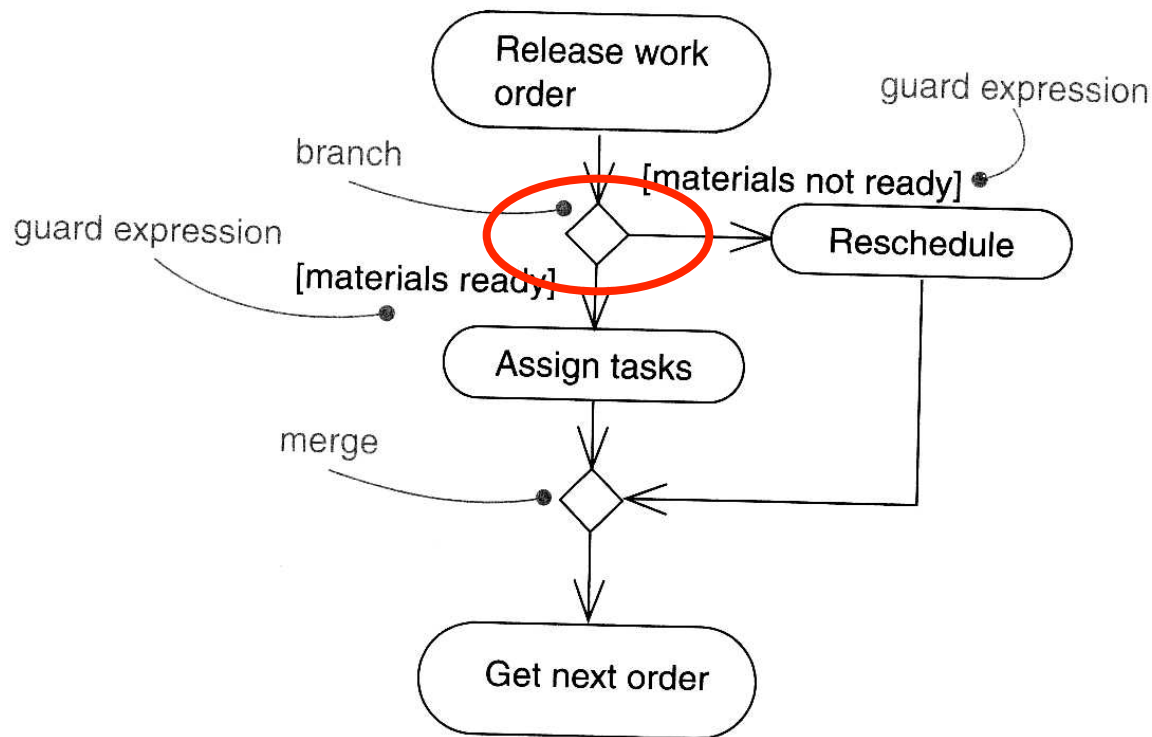
FLUXO DE CONTROLE: quando uma ação ou um nodo de atividade completa a execução, o fluxo de controle passa imediatamente para a próxima ação ou nodo de atividade.



➔ O fluxo de controle é representado por uma seta simples.

Notação - Seleção

SELEÇÃO (BRANCHING): um diagrama de atividades pode incluir uma seleção, que especifica caminhos alternativos que são tomados de acordo com alguma expressão booleana.



➔ A palavra else pode ser usada na transição de saída.



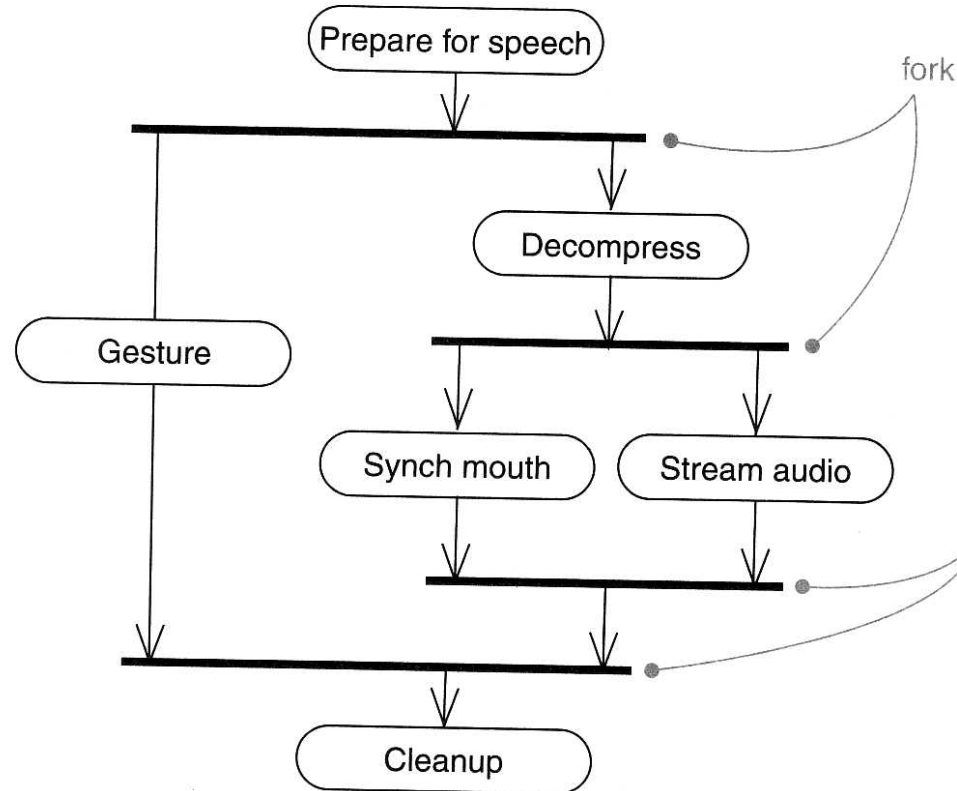
Notação - Forking e Joining

FORKING E JOINING: um diagrama de atividades pode incluir fluxos que são concorrentes.

Para representar o forking e o joining são usadas barras de sincronização (linhas verticais ou horizontais).

Notação - Forking e Joining

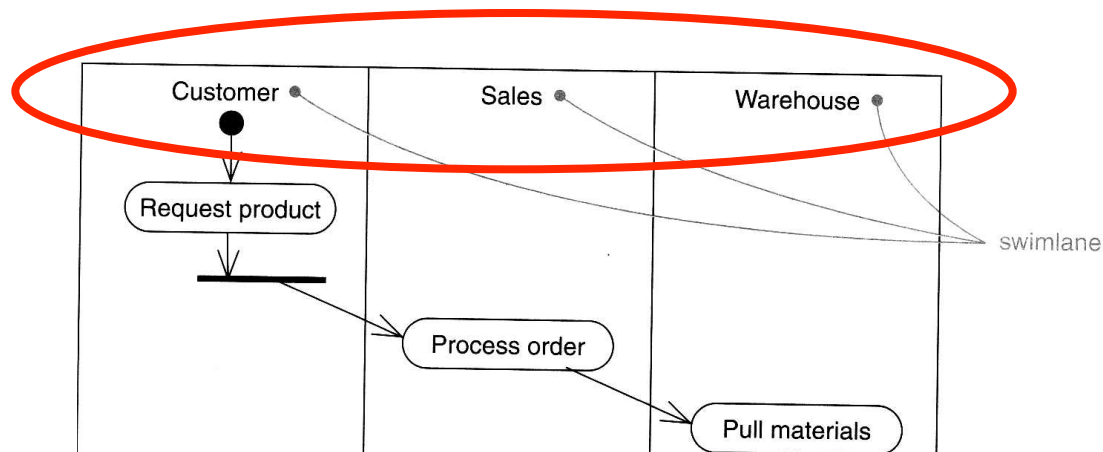
Exemplo - Dispositivo que imita a fala e gestos humanos.



➔ O número de fluxos que saem de um fork deve combinar com o número de fluxos que entram no correspondente join.

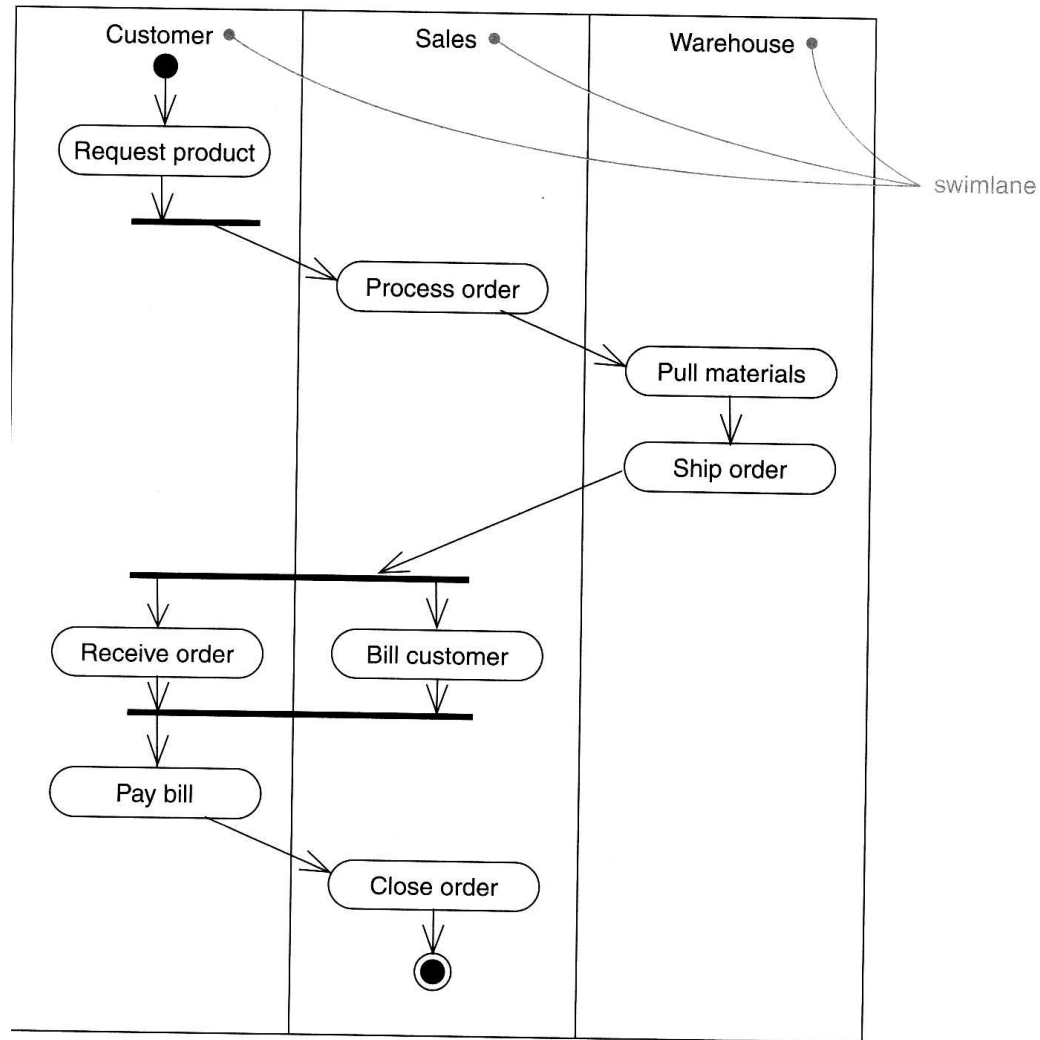
Notação – Swinlanes (Raias)

SWIMLANE (RAIAS): um diagrama de atividades pode particionar as atividades em grupos.



- Cada grupo é chamado de swimlane e especifica um conjunto de atividades que compartilham alguma propriedade organizacional.

Swimlanes





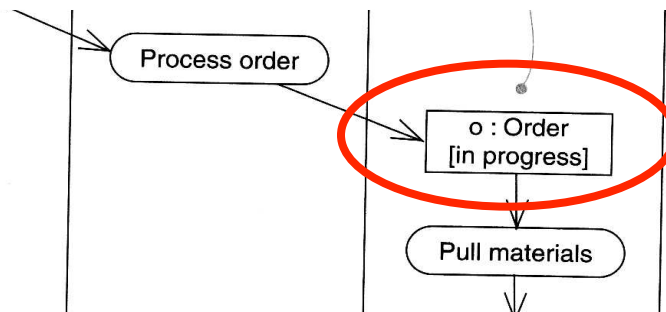
Swimlanes

Comentários:

- Cada swimlane tem um nome único dentro do seu diagrama.
- Swimlanes não tem nenhuma semântica específica.

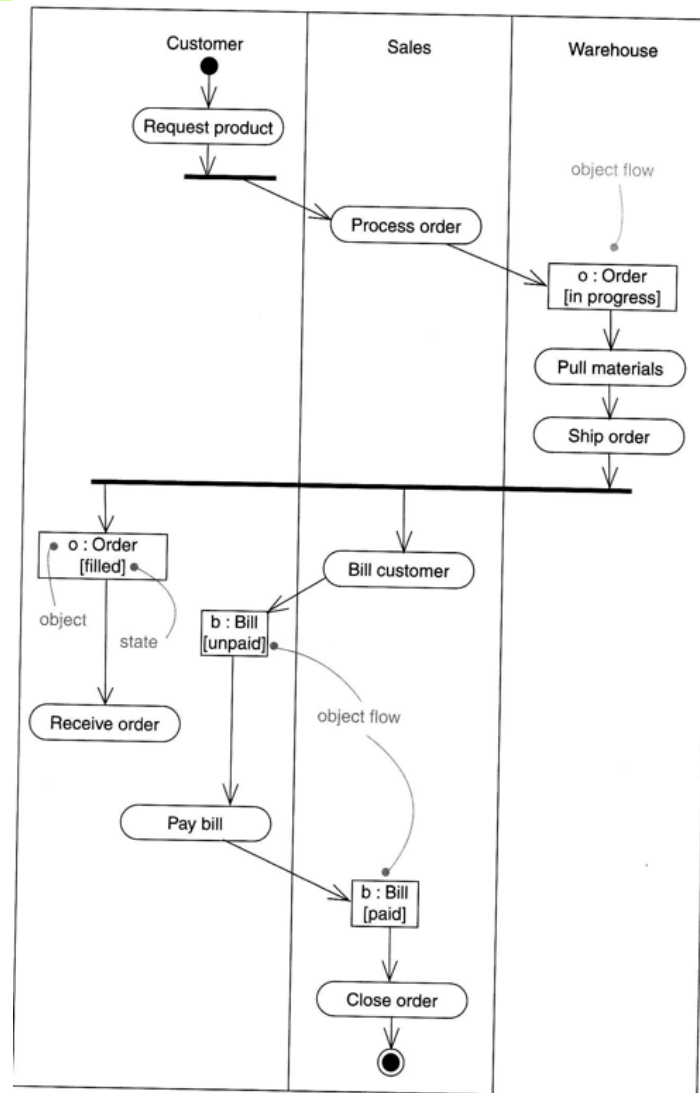
Notação - Fluxo de Objeto

FLUXO DE OBJETO: os objetos que estão envolvidos em um diagrama de atividades podem ser colocados no diagrama e conectados por setas às ações que os produzem e os consomem.



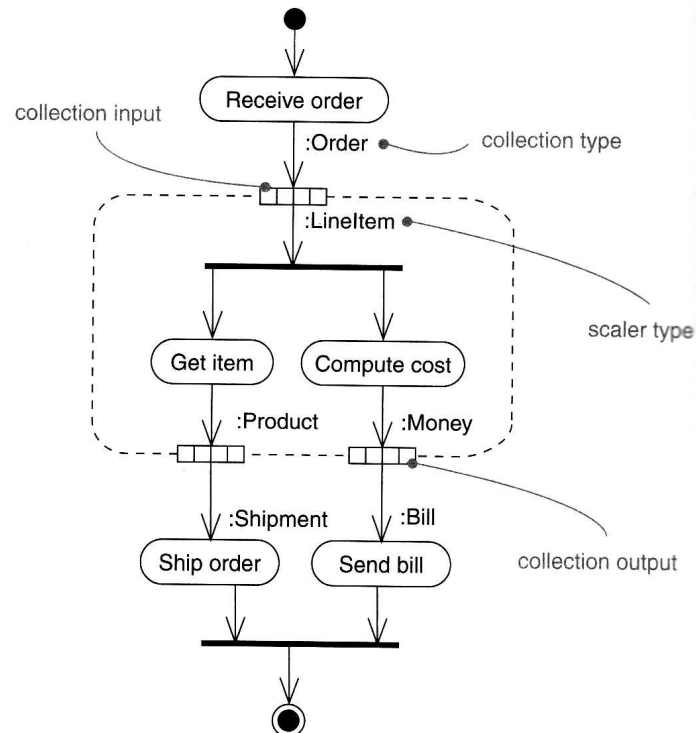
- Não é necessário desenhar o fluxo de controle entre as ações conectadas pelos fluxos de objetos.
- O estado de um objeto pode ser mostrado adicionando seu nome entre colchetes.

Fluxo de Objeto



Notação - Região de Expansão

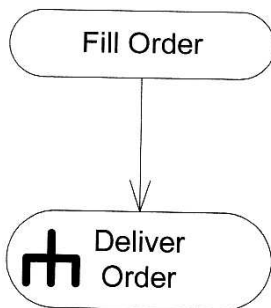
REGIÃO DE EXPANSÃO: representa um fragmento do modelo de atividade que é executado sob os elementos de uma lista ou conjunto.



- Uma linha tracejada é desenhada ao redor da região no diagrama.
- As entradas e saídas destas regiões são coleções de valores.

Notação - Rake

RAKE: Para expandir um diagrama de atividade em outro diagrama de atividade utilize o símbolo rake (ancinho).



símbolo rake

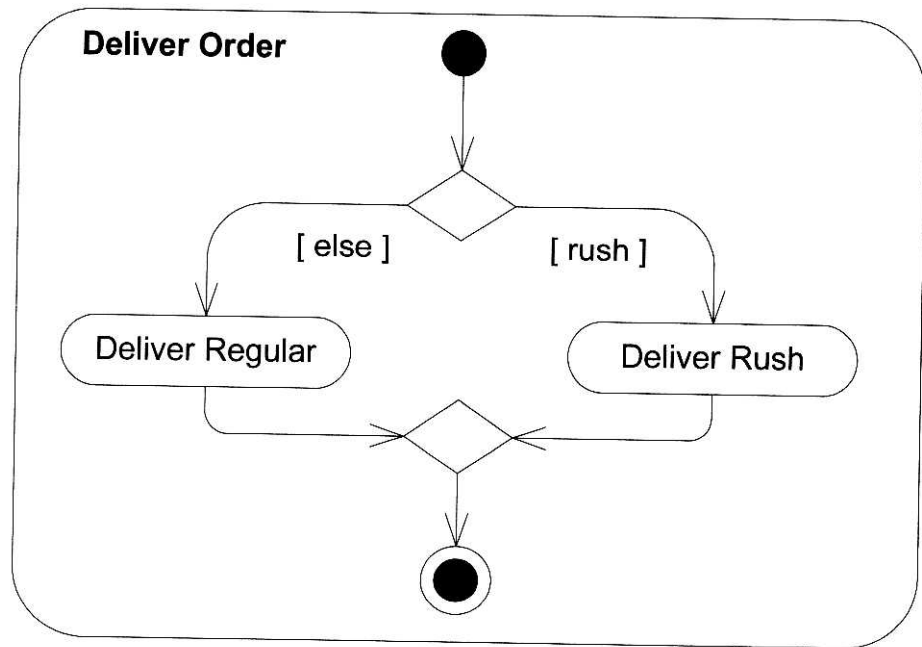


Diagrama de Atividades

- Usado para modelar algum aspecto dinâmico de um sistema: sistema como um todo, um subsistema, um caso de uso, uma operação ou uma classe.

- Geralmente, diagramas de atividades são usados para modelar:

Workflows e processos de negócios: foca nas atividades vistas pelos atores que colaboram com o sistema.

Operações: usado como um flowchart para modelar os detalhes de uma computação.

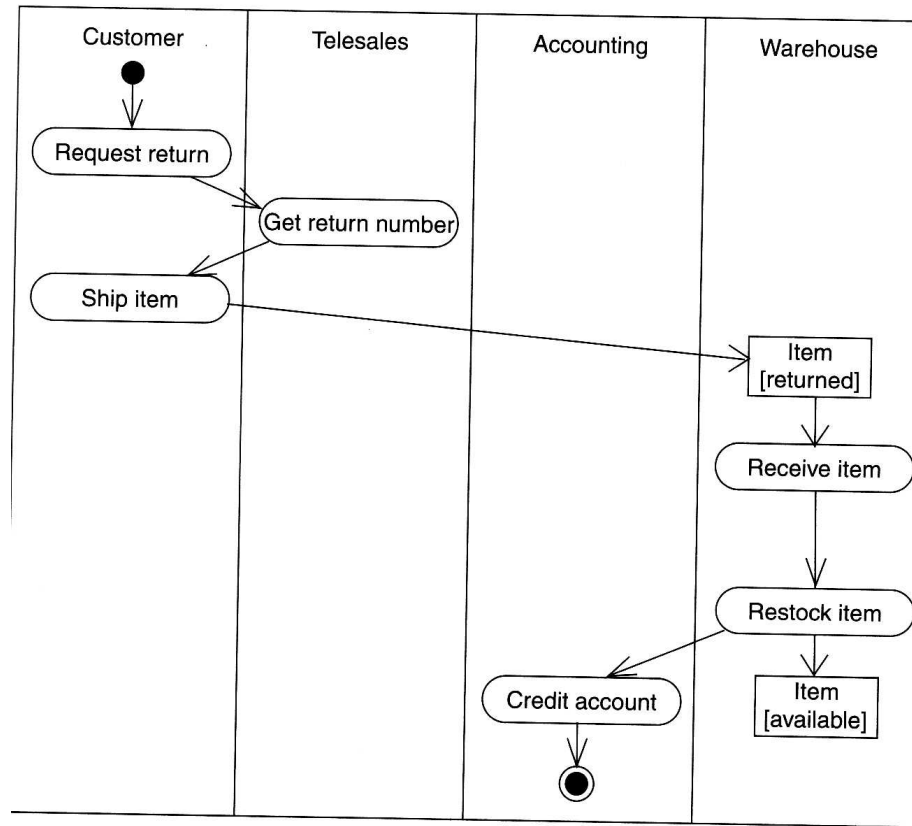



Modelando um Workflow

- Estabeleça o foco do workflow.
- Crie uma swimlane para organizações de negócio importantes.
- Identifique as pré-condições do estado inicial e as pós-condições do estado final, para ajudar a definir os limites do workflow.
- Inicie no estado inicial, especificando as ações que são executadas.
- Ações complicadas ou que aparecem várias vezes são colocadas em um diagrama de atividades separado.
- Desenhe os fluxos que conectam as ações e nodos de atividades, na seguinte sequência: fluxos sequencias, seleção, e forking e joining.
- Se existirem valores de objetos importantes envolvidos no workflow, inclua-os no diagrama.

Modelando um Workflow

Workflow da devolução de um item



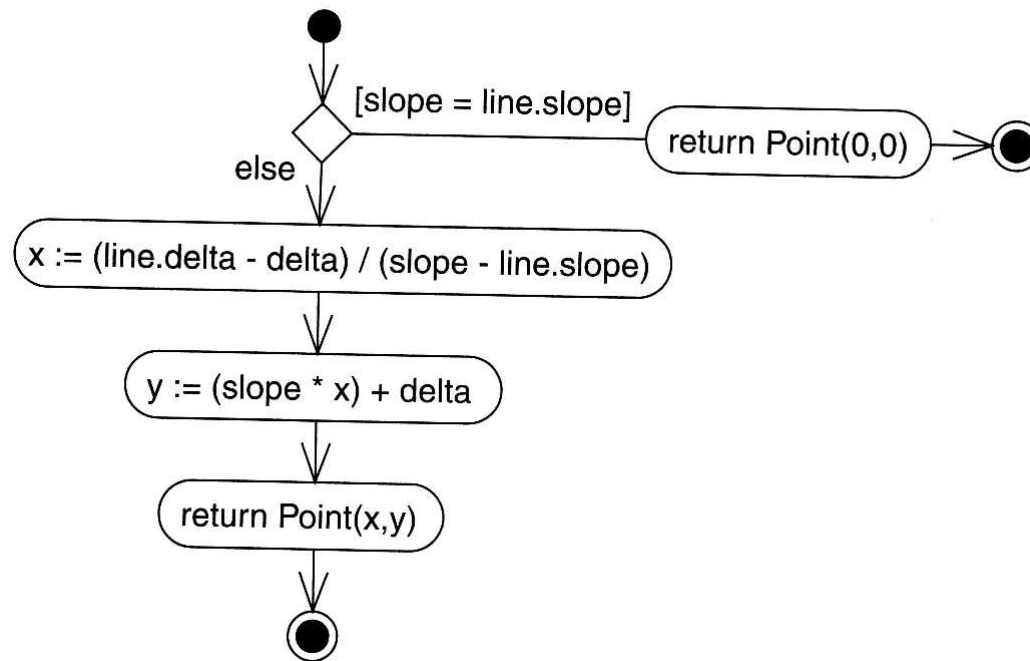


Modelando uma Operação

- Identifique as abstrações envolvidas na operação: parâmetros da operação, atributos da classe da operação e as classes envolvidas.
- Identifique as pré-condições do estado inicial e as pós-condições do estado final, e também as invariantes da classe durante a operação.
- Inicie com o estado inicial, especificando as ações que são executadas.
- Use seleção quando necessário para especificar caminhos condicionais e iteração.
- Use forking e joining quando necessário para especificar fluxos paralelos de controle.

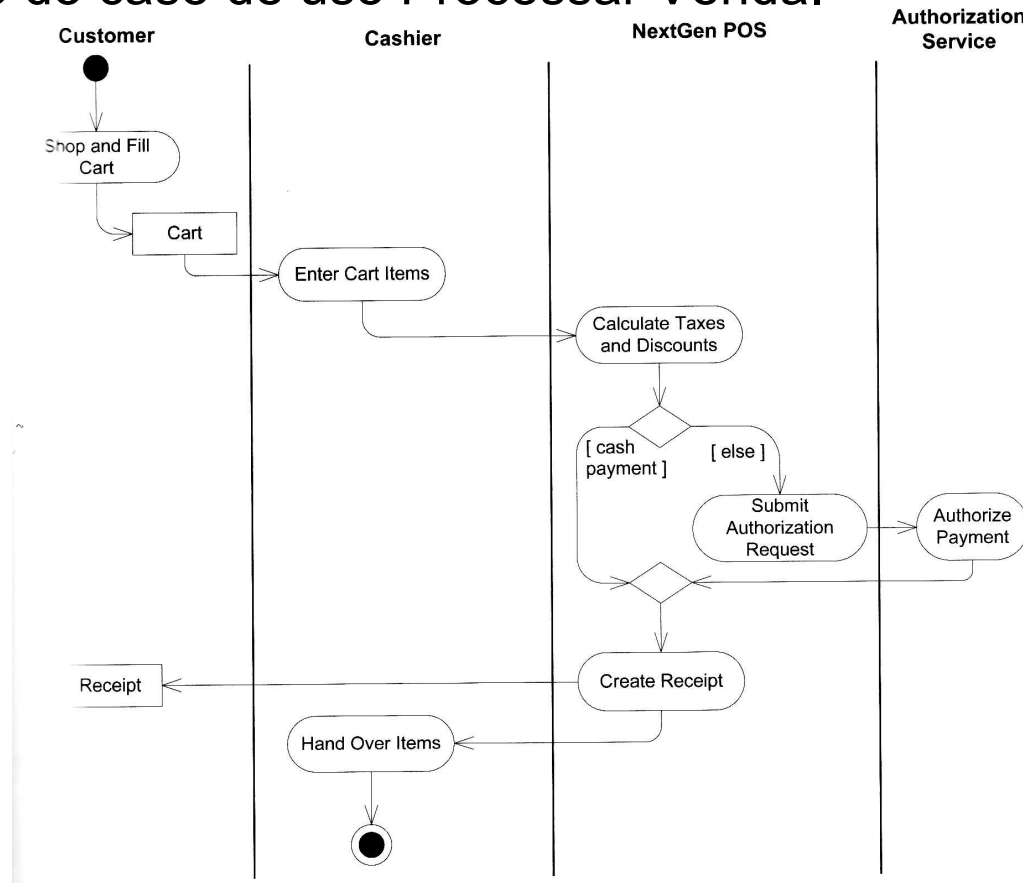
Modelando uma Operação

Operação intersecção da classe Linha



Modelando um Caso de Uso

Processo do caso de uso Processar Venda.



- Na realidade este diagrama não precisaria ser criado, pois o texto do caso de uso é suficiente.



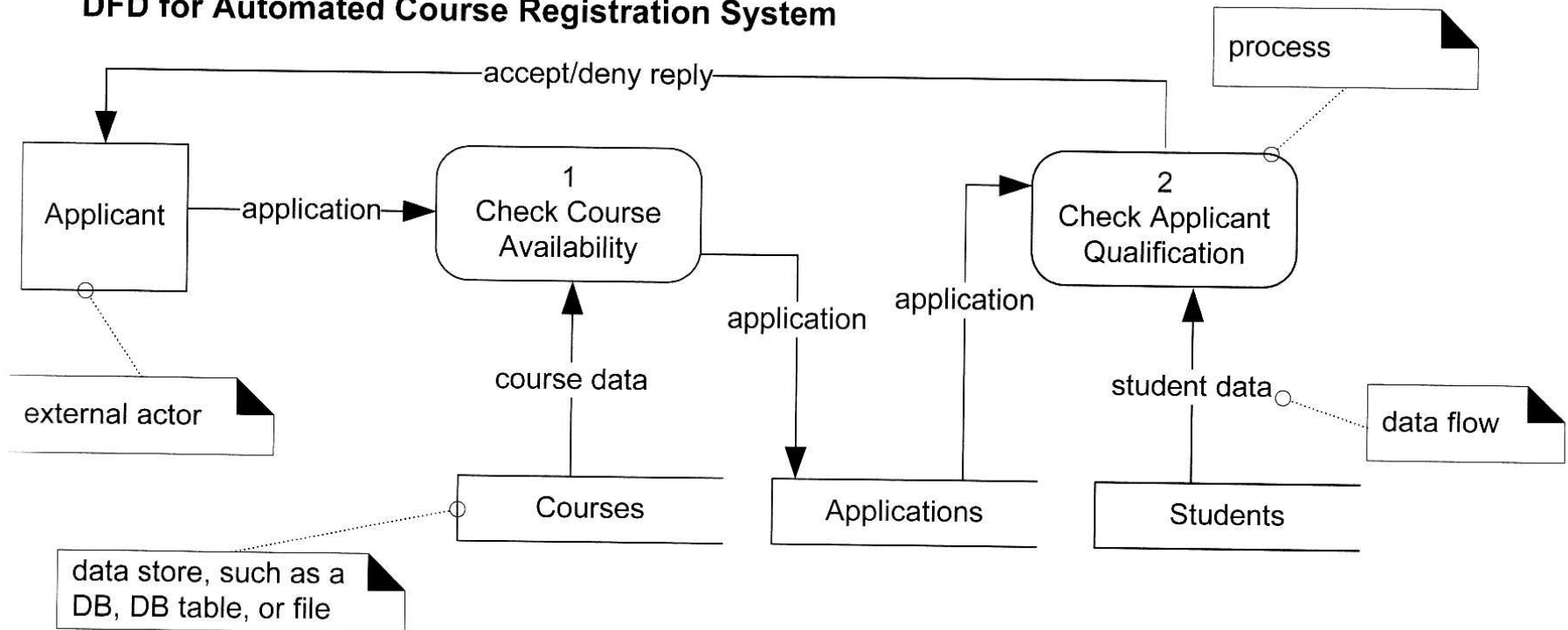
Comentário

- Diagramas de atividades são úteis para processos bem complexos, usualmente envolvendo várias partes. Os textos dos casos de uso são suficientes para processos simples.

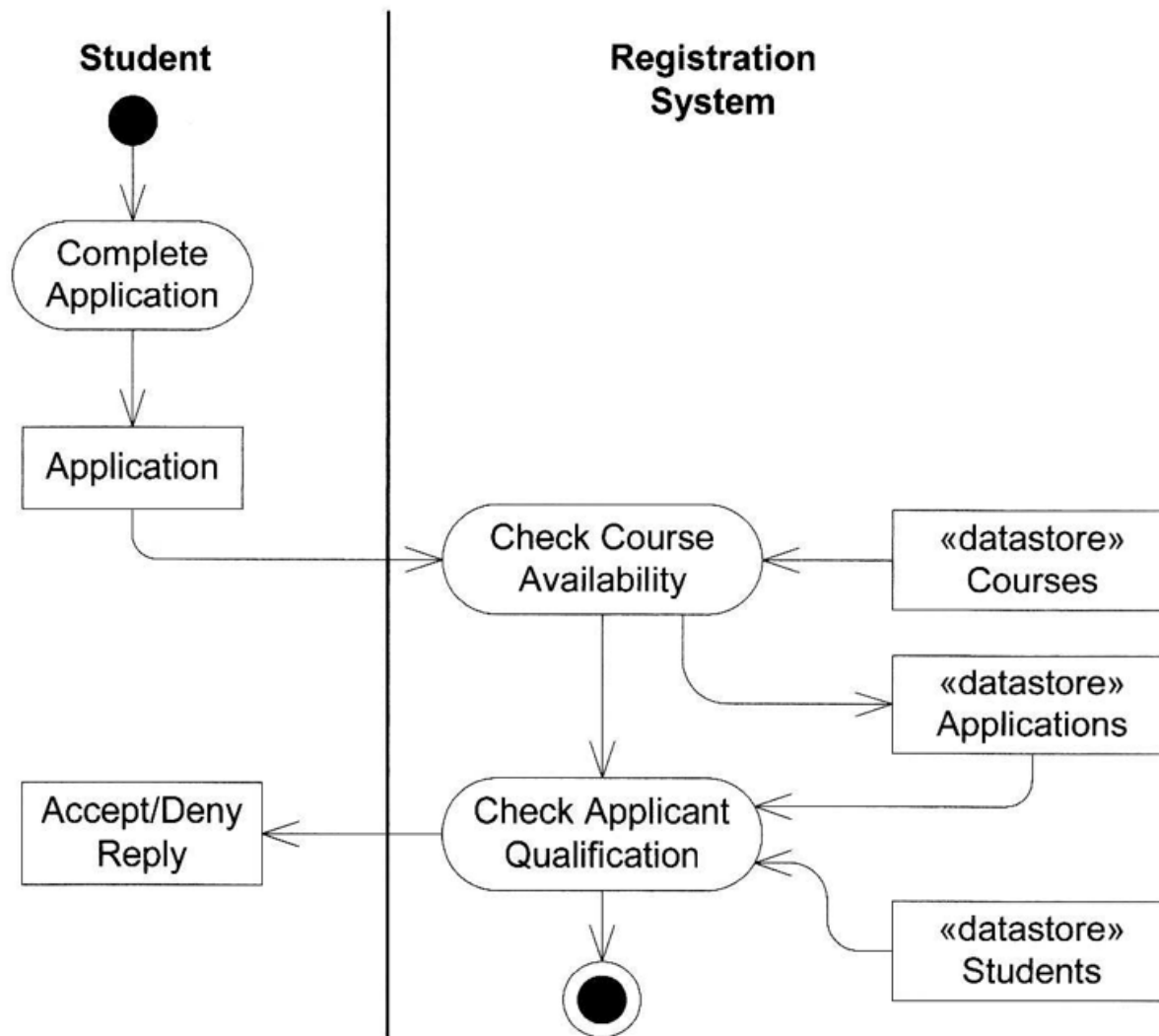
Modelagem de Fluxo de Dados

- Os diagramas de atividades também podem ser usados para modelar o fluxo de dados, substituindo a notação tradicional de DFD.

DFD for Automated Course Registration System



Modelagem de Fluxo de Dados



Exercícios

Exercício 1: Modele o seguinte caso de uso com um diagrama de atividades.

Caso de Uso Submeter um artigo

1. Para submeter um artigo, o autor fornece os seguintes dados: nome do artigo, autores do artigo (nome e instituição de origem), resumo e os tópicos (dentre os tópicos da conferência) nos quais o artigo se encaixa.
2. O sistema confere se todos os dados foram fornecidos e se não existe nenhum outro artigo com o mesmo nome e o mesmo conjunto de autores.
3. O autor faz o upload do artigo.
4. O sistema gera um identificador para o artigo e o apresenta ao autor.

Exercícios

Exercício 2: Modele o método a seguir com um diagrama de atividades.

```
public class ListBoxChoice extends MultiChoice {  
    JawsList list;  
    Vector choices;  
  
    public JPanel getUI() {  
        JPanel p = new JPanel();  
        list = new JawsList(choices.size());  
        list.setMultipleMode(true);  
        p.add(list);  
        for (int i=0; i< choices.size(); i++)  
            list.add((String)choices.elementAt(i));  
        return p;  
    }  
}
```

Exercícios

Exercício 3: Modele o método a seguir com um diagrama de atividades.

```
public class TimeSwimData extends SwimData {  
    protected Vector swimmers;  
  
    public TimeSwimData(String filename) {  
        String s = "";  
        swimmers = new Vector();  
        InputFile f = new InputFile(filename);  
        s = f.readLine();  
        while (s != null) {  
            swimmers.addElement(new Swimmer(s));  
            s = f.readLine();  
        }  
        f.close();  
    }  
}
```

Exercícios

Exercício 4: Modele o método a seguir com um diagrama de atividades.

```
public class ContadorPalavras {  
  
    public TabelaHash contaPalavras (String frase){  
        TabelaHash tabela = new TabelaHash(13);  
        StringTokenizer st = new StringTokenizer (frase, " ", false);  
        while (st.hasMoreTokens()){  
            String token = st.nextToken();  
            Integer ocorrencia = (Integer) tabela.retorna(token);  
            if (ocorrencia != null){  
                int numOcorrencia = ocorrencia.intValue();  
                tabela.remove(token);  
                tabela.insere(token, new Integer(numOcorrencia+1));    }  
            else{  
                tabela.insere(token, new Integer(1));    }  
        }  
        return tabela; }  
}
```