Engenharia de Software

Modelos de Dinâmica – Parte 3

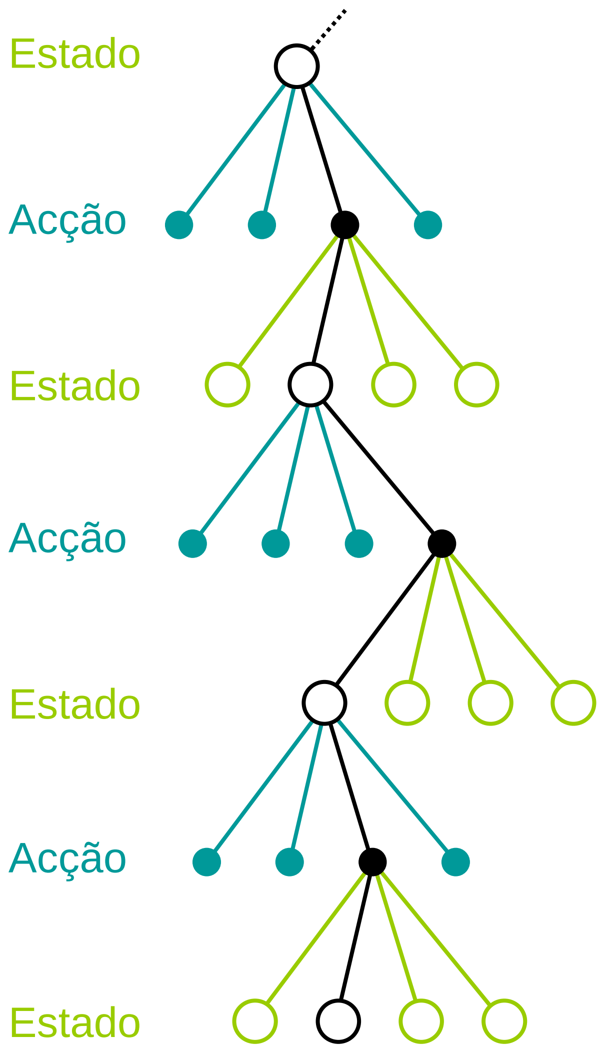
Luís Morgado

Instituto Superior de Engenharia de Lisboa

Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

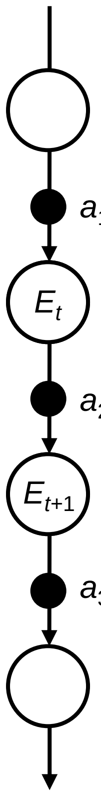
Modelos de Dinâmica

Evolução no espaço de estados

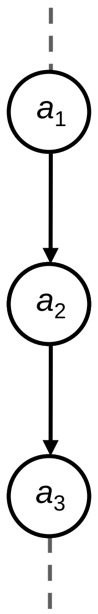


Modelos

Foco nos estados

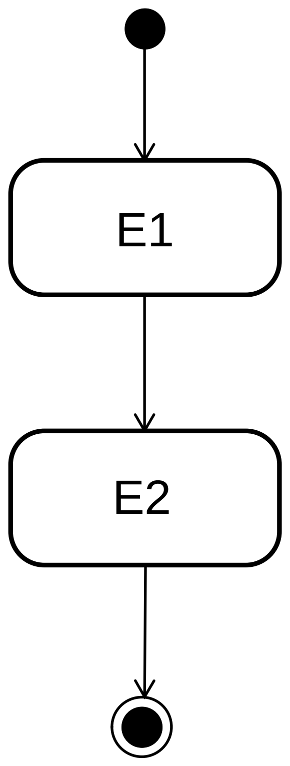


Foco nas acções



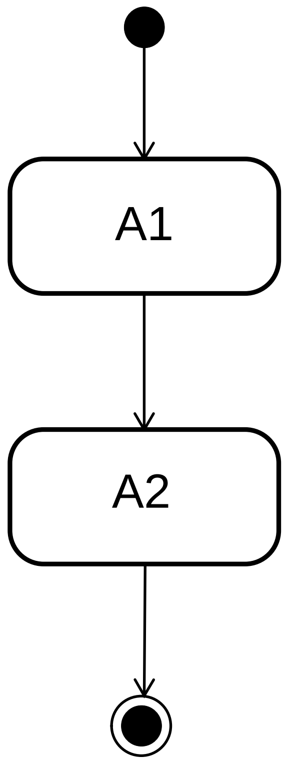
Modelos de Dinâmica

Diagrama de Transição de Estado



Semântica declarativa
"O que acontece…"

Diagrama de Actividade



Semântica imperativa
"Como acontece…"

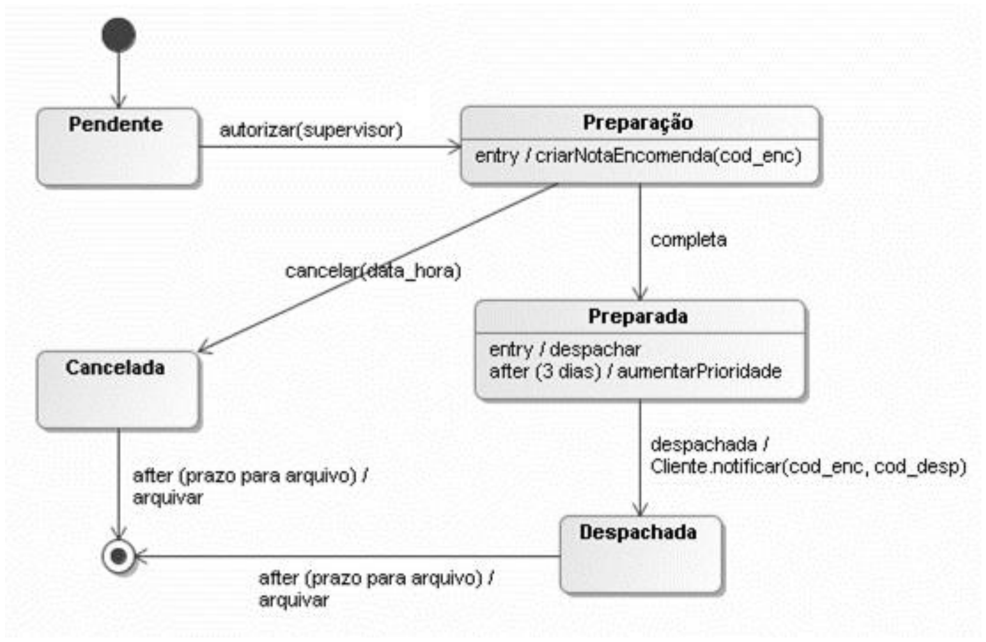
Sintaxe idêntica – Semântica diferente

Exemplo: Processamento de encomendas

Diagramas de Transição de Estado

Semântica declarativa

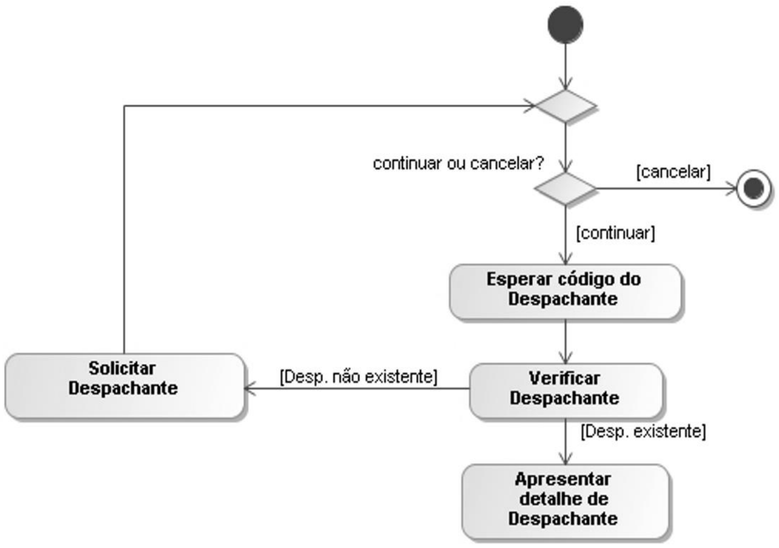
"*O que acontece …*"


Pendente
autorizar(supervisor)
Preparação
entry / criarNotaEncomenda(cod_enc)
completa
cancelar(data_hora)
Preparada
entry / despachar
after (3 dias) / aumentarPrioridade
Cancelada
despachada / Cliente.notificar(cod_enc, cod_desp)
after (prazo para arquivo) / arquivar
Despachada
after (prazo para arquivo) / arquivar

Diagramas de Actividade

Semântica imperativa

"*Como acontece …*"

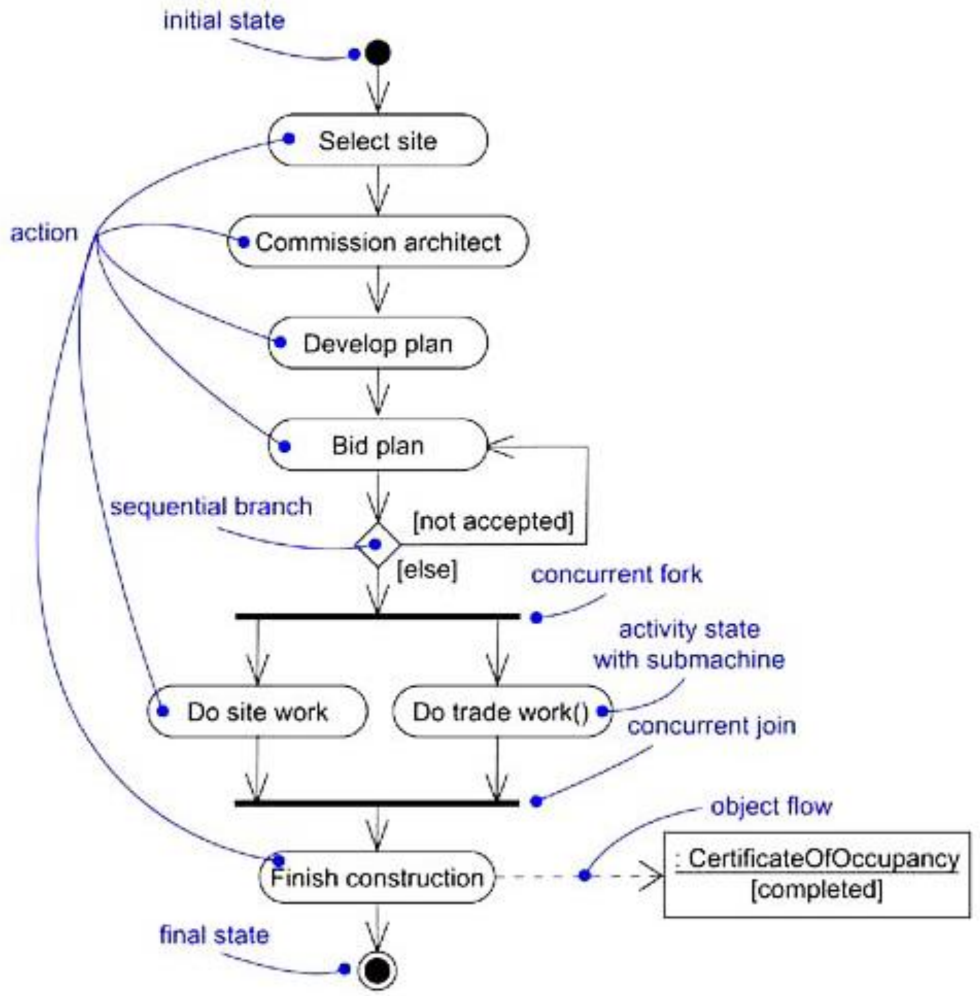


Diagramas de Actividade

An *activity diagram* shows the flow from activity to activity.

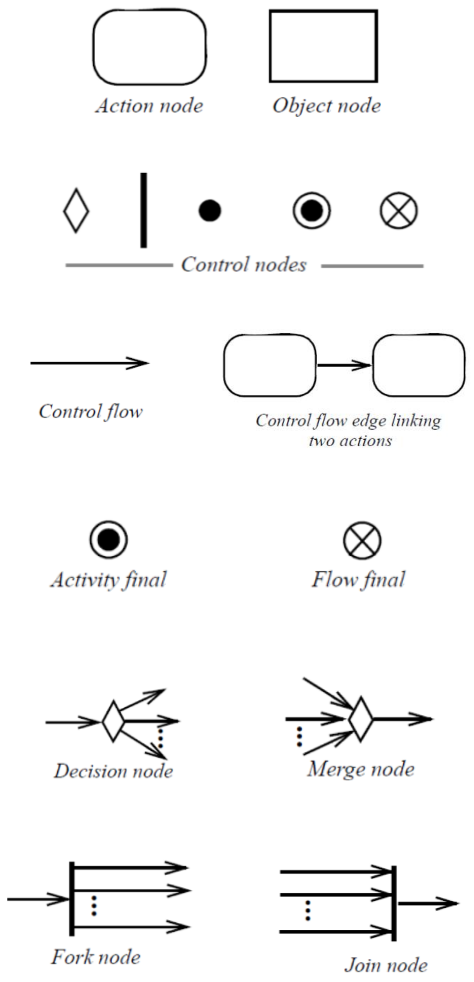
An activity is an ongoing nonatomic execution within a state machine. Activities ultimately result in some *action,* which is made up of executable atomic computations that result in a change in state of the system or the return of a value. Actions encompass calling another operation, sending a signal, creating or destroying an object, or some pure computation, such as evaluating an expression.

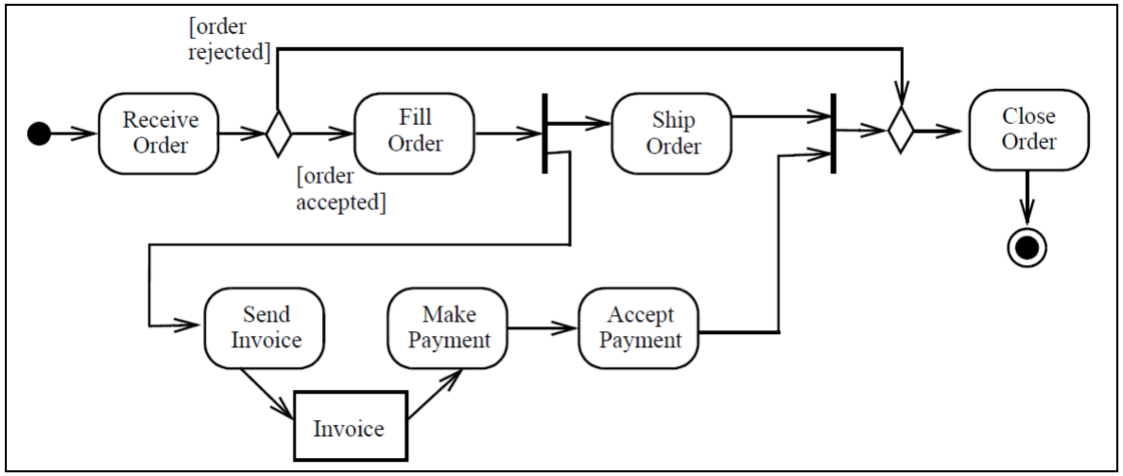
Graphically, an activity diagram is a collection of vertices and arcs.

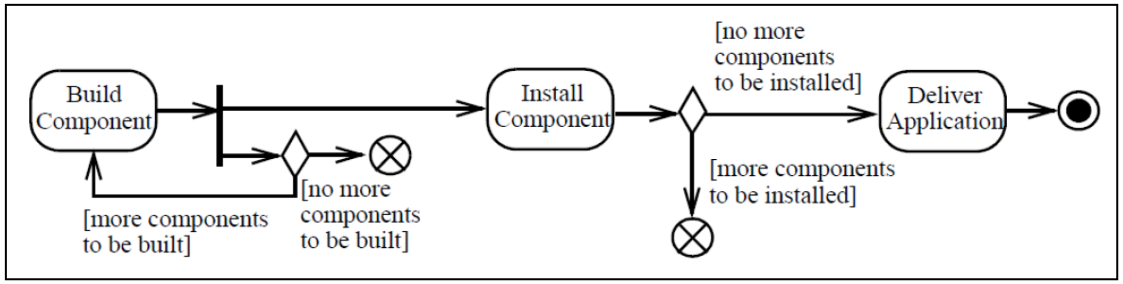


[UML User Guide, Booch *et al.* 1998]

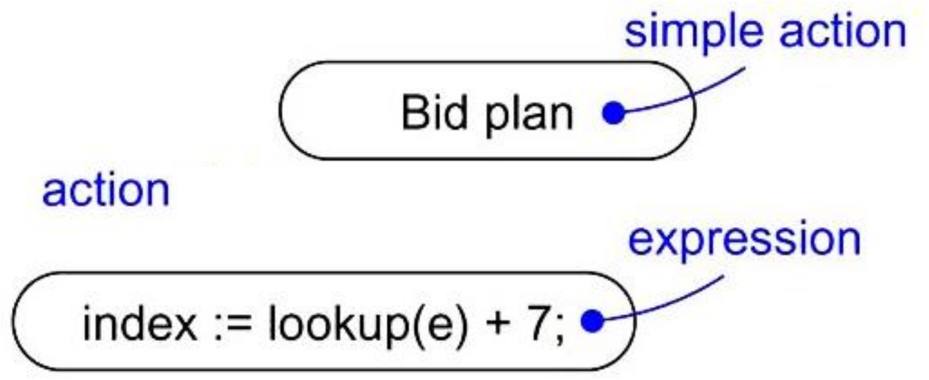
Diagramas de Actividade





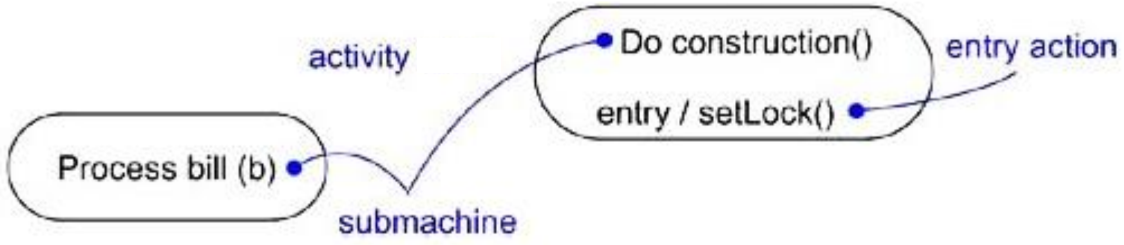

Build Component
[more components to be built]
[no more components to be built]
Install Component
[no more components to be installed]
[more components to be installed]
Deliver Application

Acções e Actividades



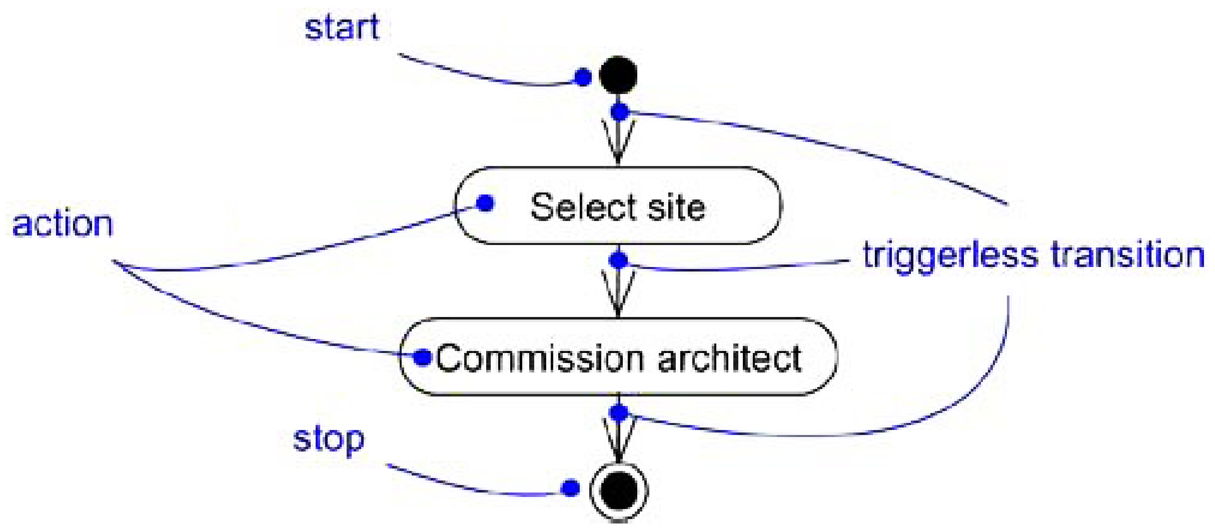
Note

The UML does not prescribe the language of these expressions. Abstractly, you might just use structured text; more concretely, you might use the syntax and semantics of a specific programming language.



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

Transições

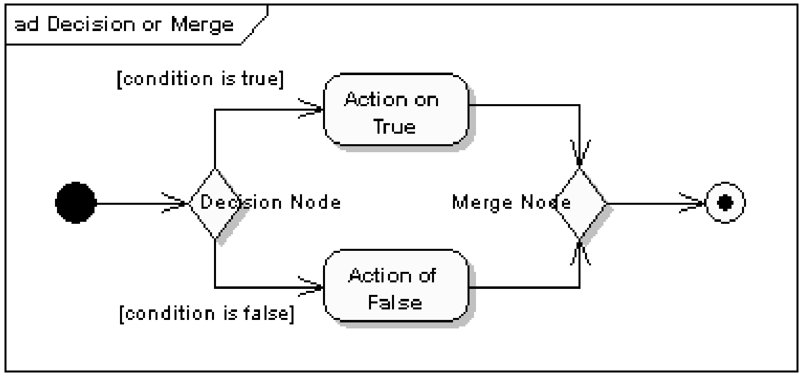


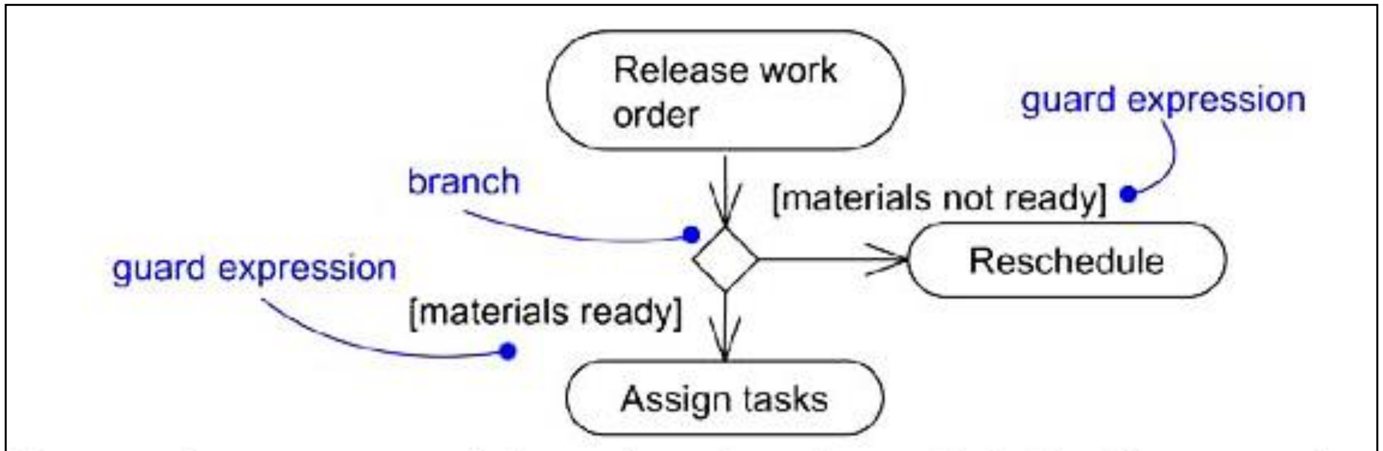
Note

Semantically, these are called triggerless, or completion, transitions because control passes immediately once the work of the source state is done. Once the action of a given source state completes, you execute that state's exit action (if any). Next, and without delay, control follows the transition and passes on to the next action or activity state. You execute that state's entry action (if any), then you perform the action or activity of the target state, again following the next transition once that state's work is done. This flow of control continues indefinitely (in the case of an infinite activity) or until you encounter a stop state.

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

Decisões

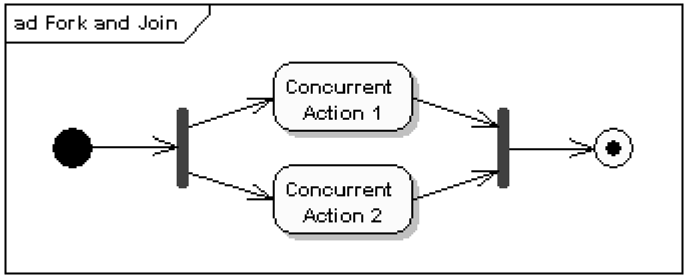




As a convenience, you can use the keyword `else` to mark one outgoing transition, representing the path taken if no other guard expression evaluates to true.

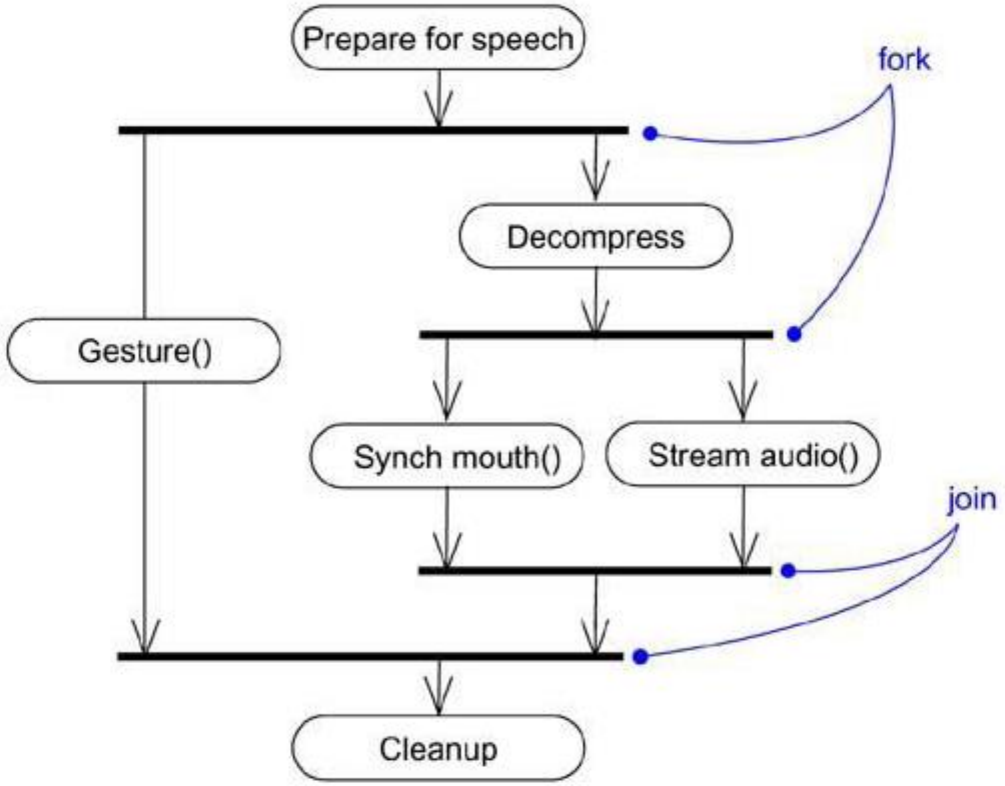
[UML User Guide, Booch *et al.* 1998]

Bifurcação / Reunião



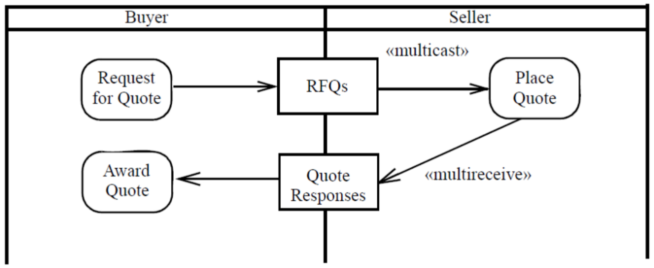
Note
Joins and forks should balance, meaning that the number of flows that leave a fork should match the number of flows that enter its corresponding join. Also, activities that are in parallel flows of control may communicate with one another by sending signals.

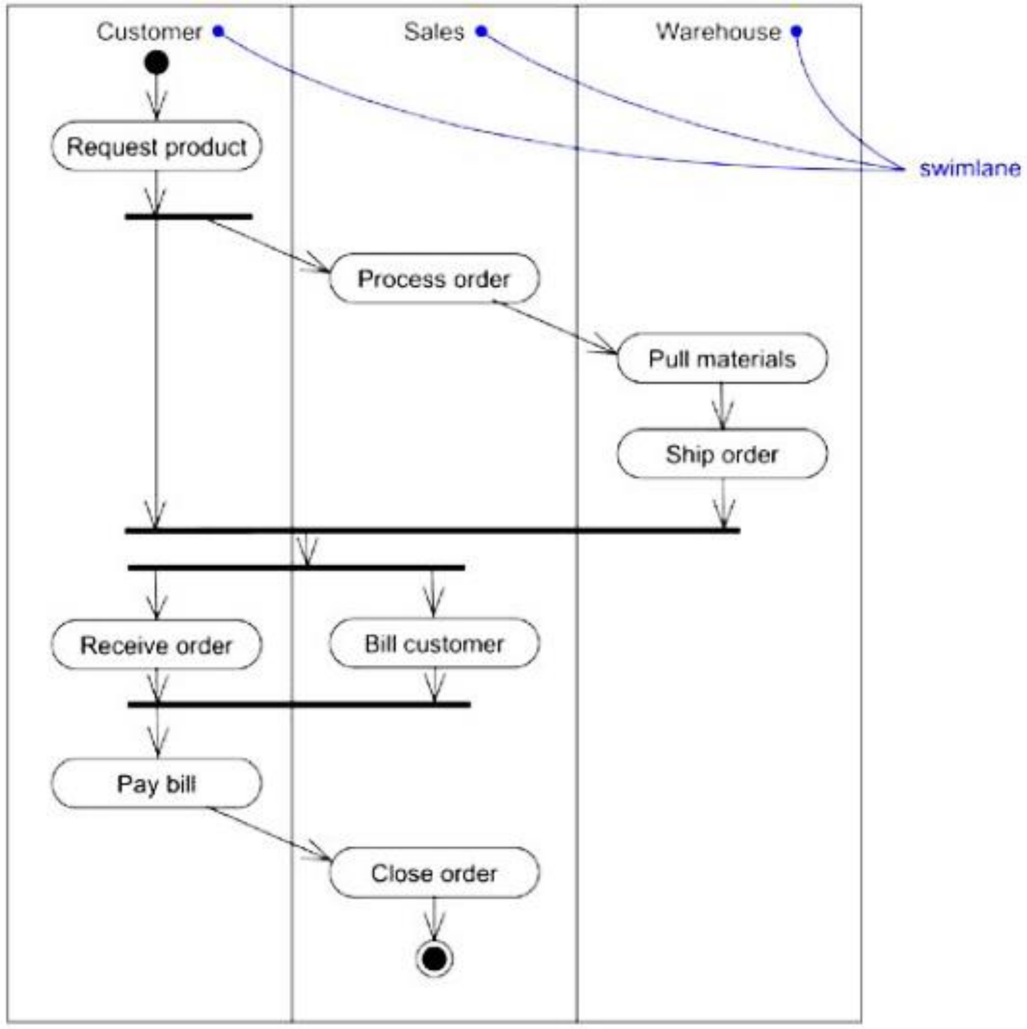
[UML User Guide, Booch *et al.* 1998]



Partições (*Swimlanes*)

A swimlane represents some real-world entity. Each swimlane represents a high-level responsibility for part of the overall activity of an activity diagram, and each swimlane may eventually be implemented by one or more classes.



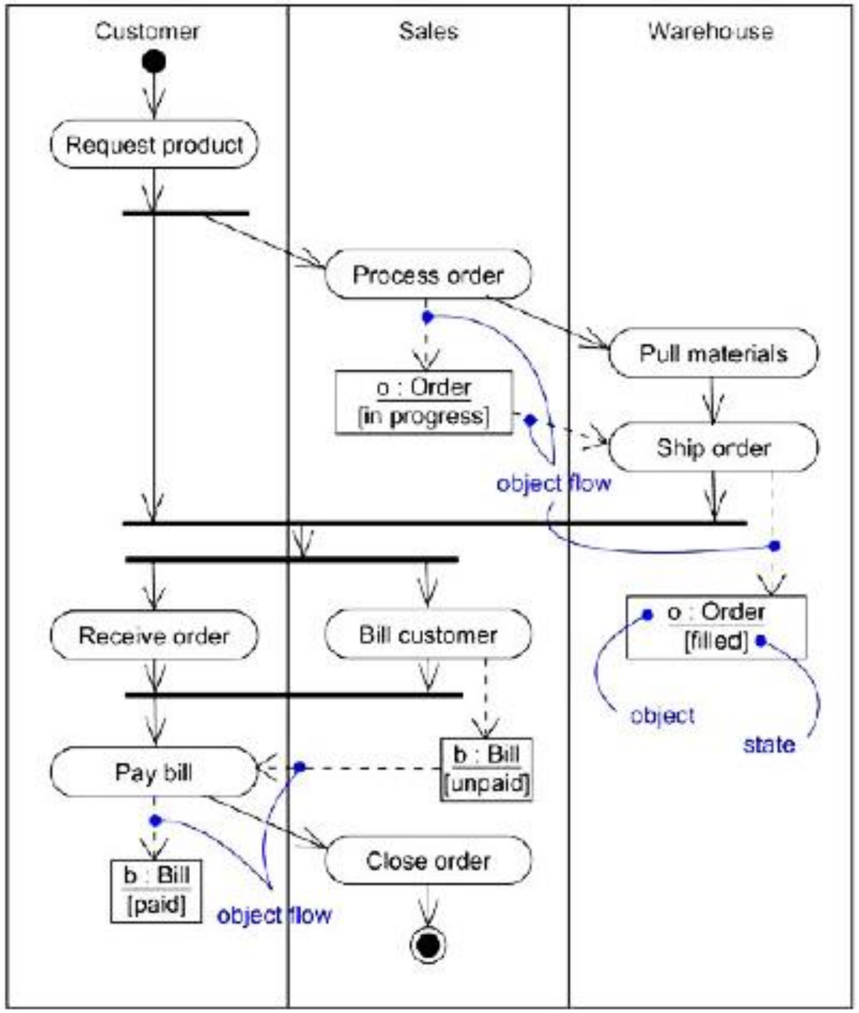


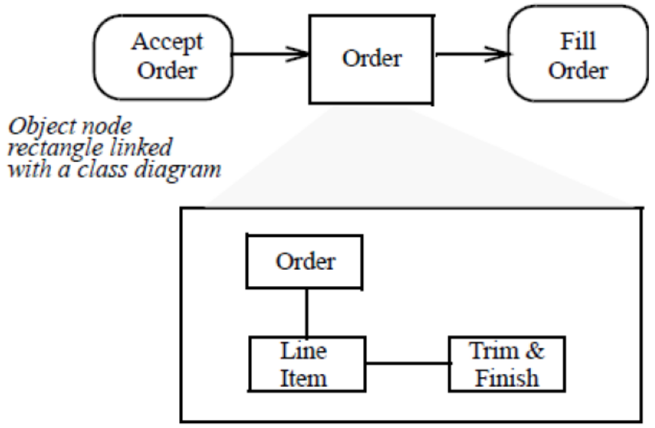
[UML User Guide, Booch *et al.* 1998]

Fluxo de Objectos

Objects may be involved in the flow of control associated with an activity diagram.

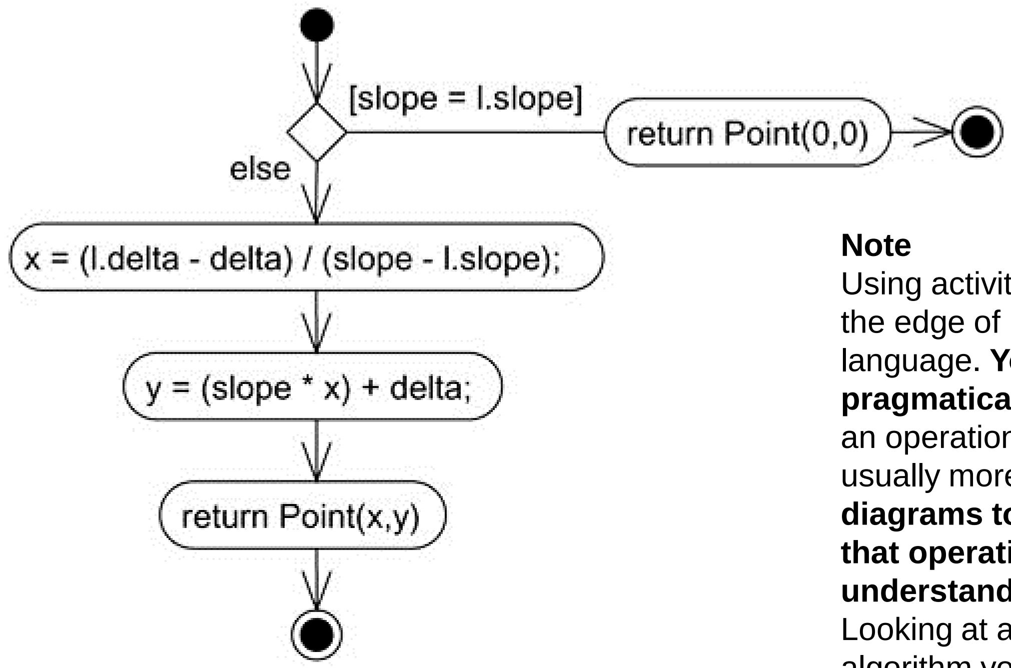
In addition to showing the flow of an object through an activity diagram, you can also **show how its role, state and attribute values change**. The state of an object is represented by naming its state in brackets below the object's name. The value of an object's attributes is represented by rendering them in a compartment below the object's name.





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

Modelação de Operações

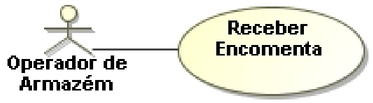


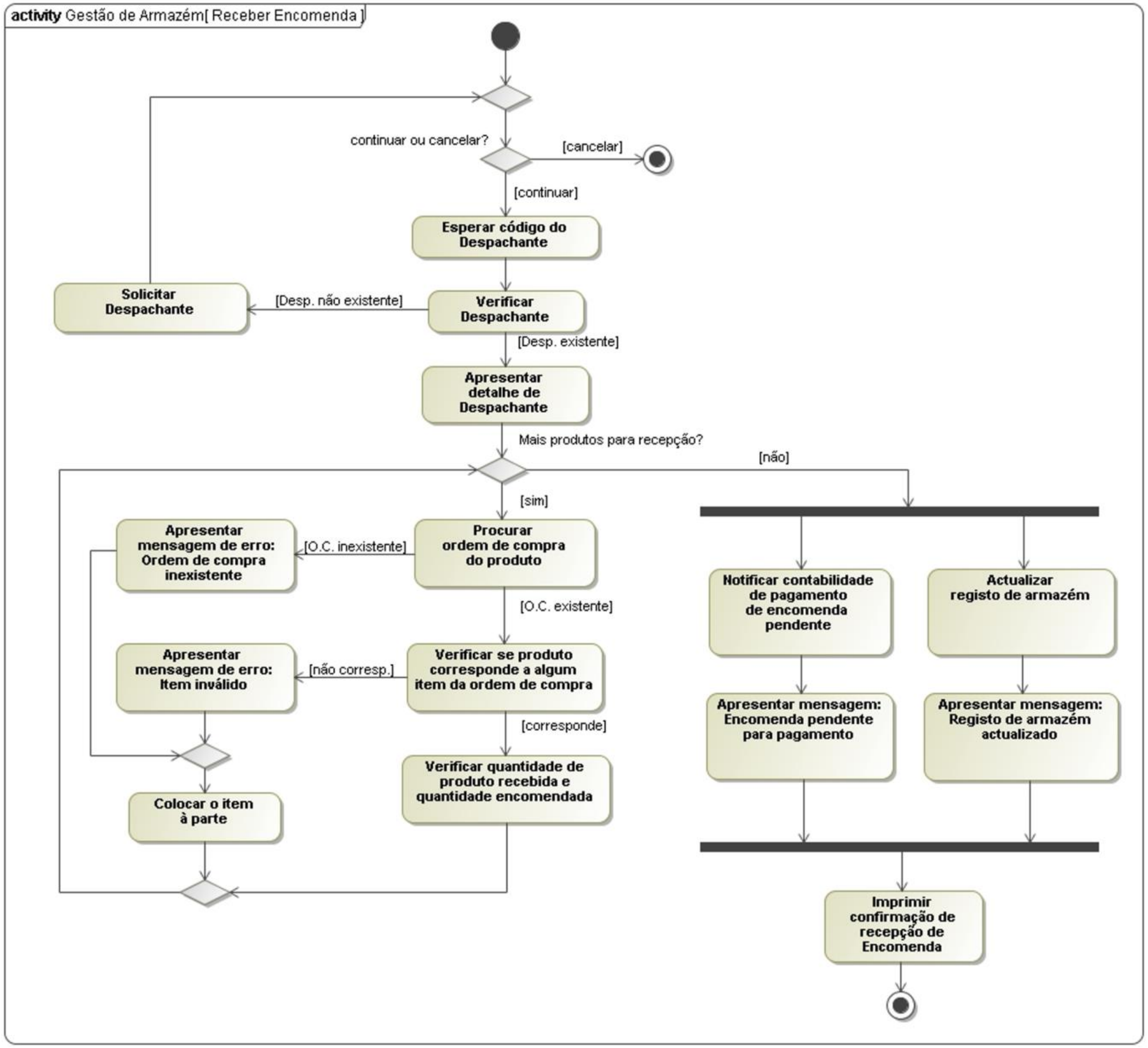
Note
Using activity diagrams to flowchart an operation lies on the edge of making the UML a visual programming language. **You can flowchart every operation, but pragmatically, you won't want to**. Writing the body of an operation in a specific programming language is usually more direct. **You will want to use activity diagrams to model an operation when the behavior of that operation is complex and therefore difficult to understand just by staring at code**.
Looking at a flowchart will reveal things about the algorithm you could not have seen just by looking at the code.

```
Point Line::intersection (l : Line) {
   if (slope == l.slope) return Point(0,0);
   int x = (l.delta - delta) / (slope - l.slope);
   int y = (slope * x) + delta;
   return Point(x, y);
}
```

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

Modelação de Processos



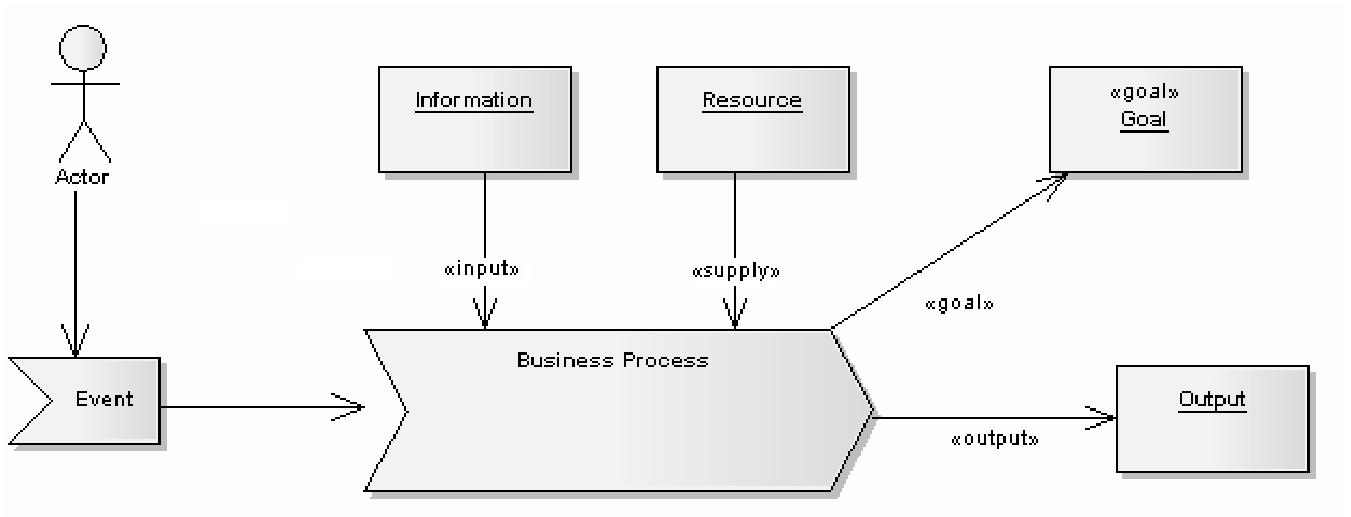


activity Gestão de Armazém[Receber Encomenda]
continuar ou cancelar?
[cancelar]
[continuar]
Esperar código do Despachante
Solicitar Despachante
[Desp. não existente]
Verificar Despachante
[Desp. existente]
Apresentar detalhe de Despachante
Mais produtos para recepção?
[não]
[sim]
Apresentar mensagem de erro: Ordem de compra inexistente
[O.C. inexistente]
Procurar ordem de compra do produto
[O.C. existente]
Apresentar mensagem de erro: Item inválido
[não corresp.]
Verificar se produto corresponde a algum item da ordem de compra
[corresponde]
Verificar quantidade de produto recebida e quantidade encomendada
Colocar o item à parte
Notificar contabilidade de pagamento de encomenda pendente
Actualizar registo de armazém
Apresentar mensagem: Encomenda pendente para pagamento
Apresentar mensagem: Registo de armazém actualizado
Imprimir confirmação de recepção de Encomenda

Processo de Negócio

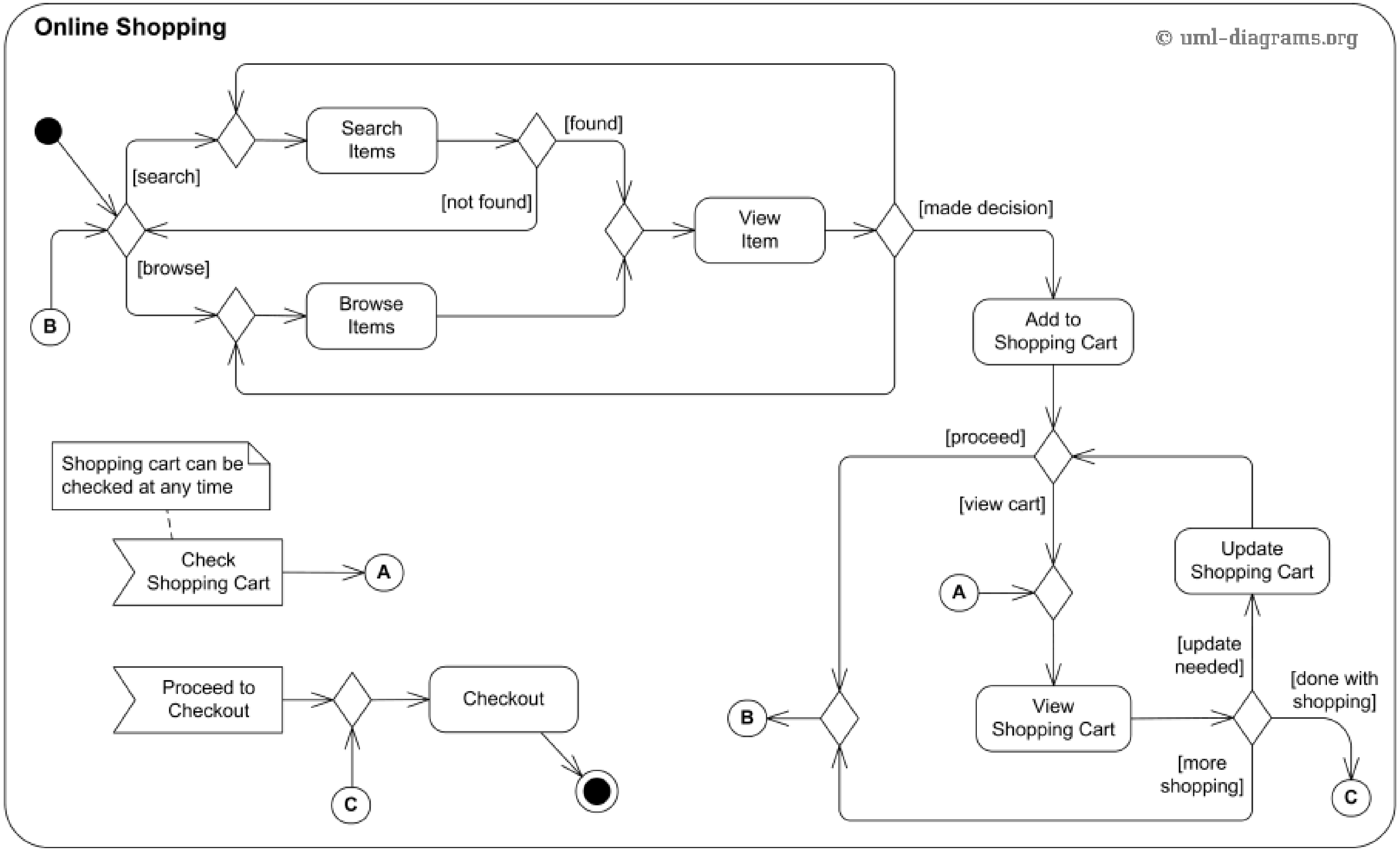
Conjunto de atividades relacionadas, organizadas de modo a produzir um produto ou serviço com um objectivo específico

- Objectivo
- Entradas
- Saídas
- Eventos
- Recursos utilizados
- Actividades realizadas numa ordem específica



[Sparx Systems, 2015]

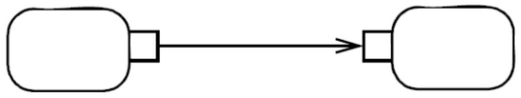
Modelação de Processos



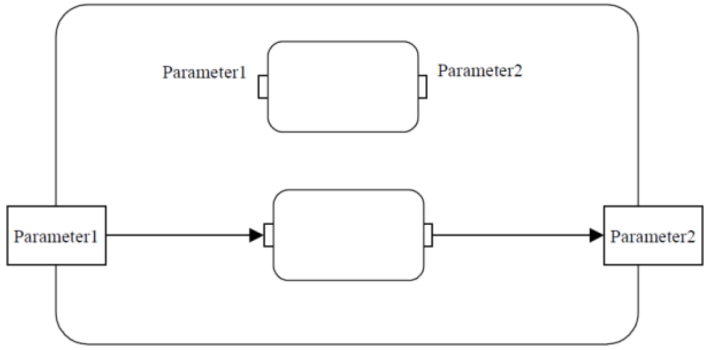
Parâmetros de Actividades

Representação de fluxo de informação entre actividades

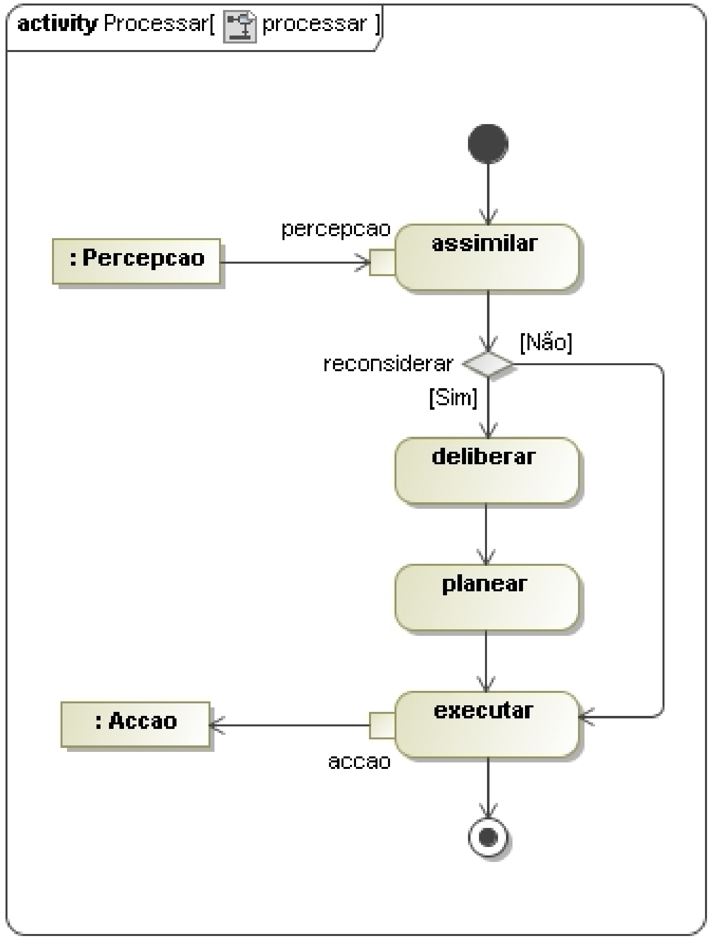
Parâmetros de entrada e de saída



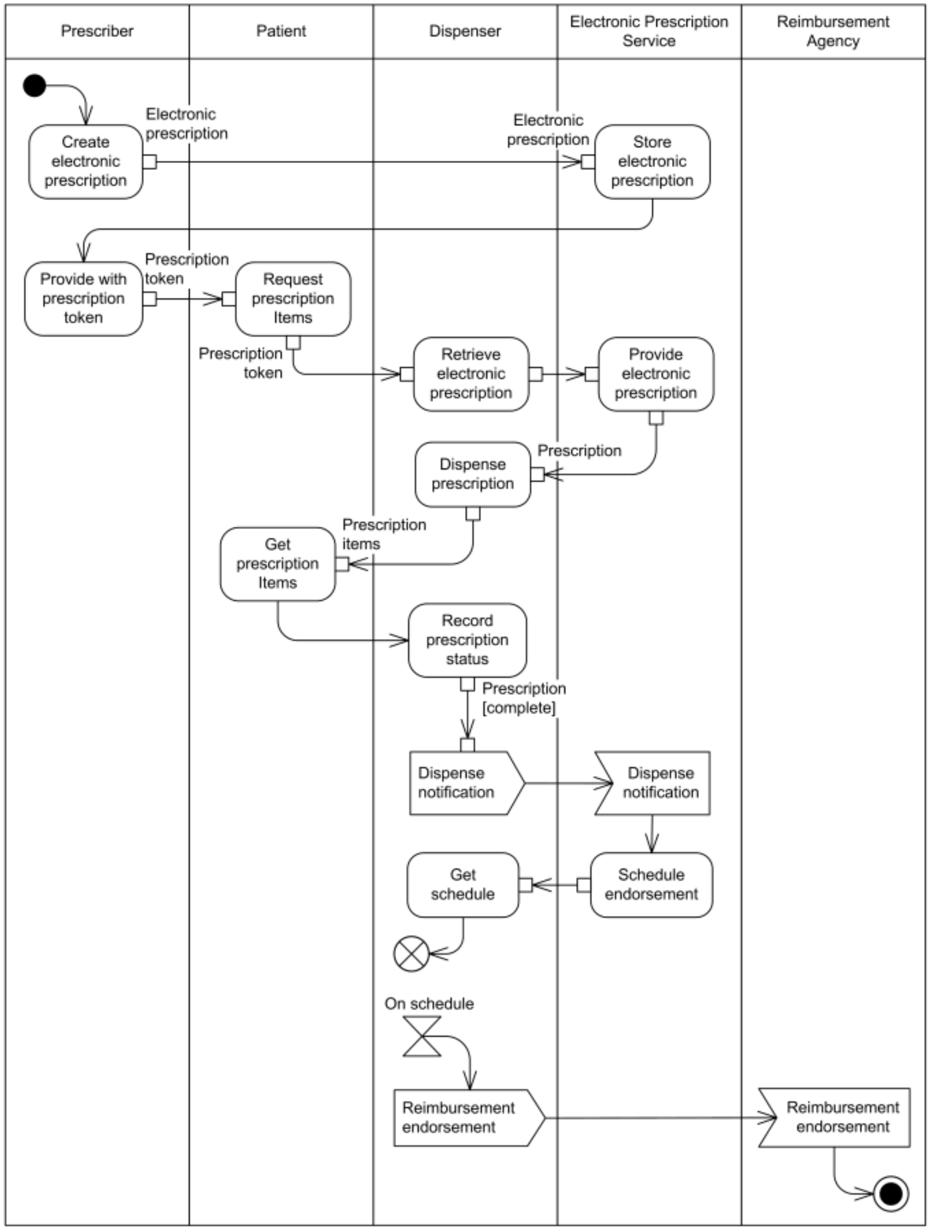
An object flow arrow linking two object node pins.



[OMG, 2020]



Exemplo


Prescriber
Patient
Dispenser
Electronic Prescription Service
Reimbursement Agency
Create electronic prescription
Electronic prescription
Electronic prescription
Store electronic prescription
Provide with prescription token
Prescription token
Request prescription Items
Prescription token
Retrieve electronic prescription
Provide electronic prescription
Dispense prescription
Prescription
Get prescription Items
Prescription items
Record prescription status
Prescription [complete]
Dispense notification
Dispense notification
Get schedule
Schedule endorsement
On schedule
Reimbursement endorsement
Reimbursement endorsement

Bibliografia

[Booch et al., 1998]

G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1998.

[Eriksson et al., 2004]

H. Eriksson, M. Penker, B. Lyons, D. Fado, *UML 2 Toolkit*, Wiley, 2004.

[OMG, 2020]
Unified Modeling Language (Specification), OMG, 2020.