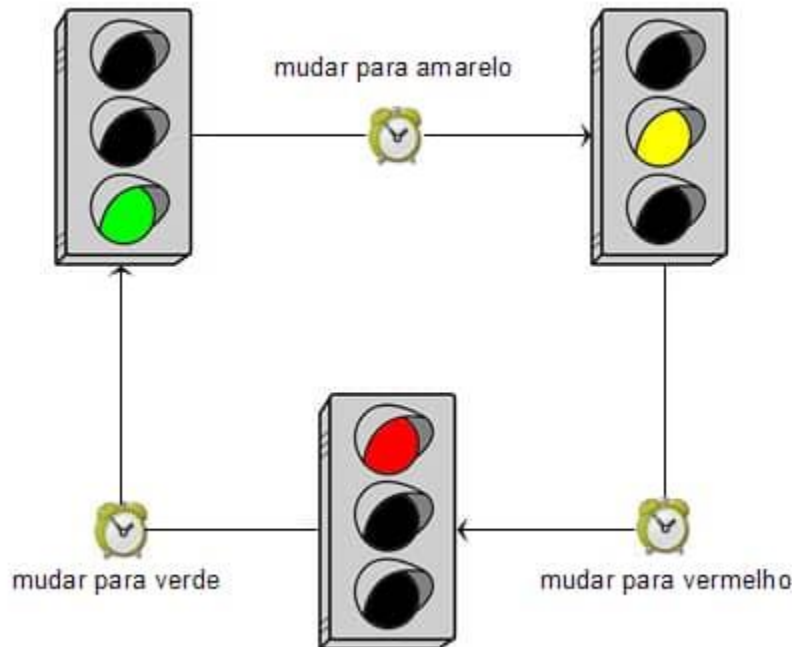


Empregar os diagramas de estado e de atividades no projeto de sistema

TRANSIÇÃO ENTRE ESTADOS



Objetos de um sistema de software orientado a objetos se encontram em um estado particular a cada momento. Um objeto muda de estado quando acontece algum evento interno ou externo ao sistema. Quando um objeto muda de um estado para outro, diz-se que ele realizou uma **transição entre estados**. Os estados e as transições de estado de um objeto constituem o seu ciclo de vida.

No momento de sua transição de um estado para outro, um objeto normalmente realiza determinadas ações dentro do sistema. Cada objeto pode passar por um número finito de estados durante a sua vida. Quando um objeto transita de um estado para outro, significa que o sistema no qual ele está inserido também está mudando de estado.

Pela análise das transições entre estados dos objetos de um sistema de software, é possível prever todas as possíveis operações realizadas, em função de eventos que podem ocorrer.

O diagrama da UML utilizado para realizar essa análise é o diagrama de estados, também conhecido como diagrama de máquina de estado ou de transição de estado (DTE).

Esse diagrama permite descrever o ciclo de vida de objetos de uma classe, os eventos que causam a transição de um estado para outro e a realização de operações resultantes.

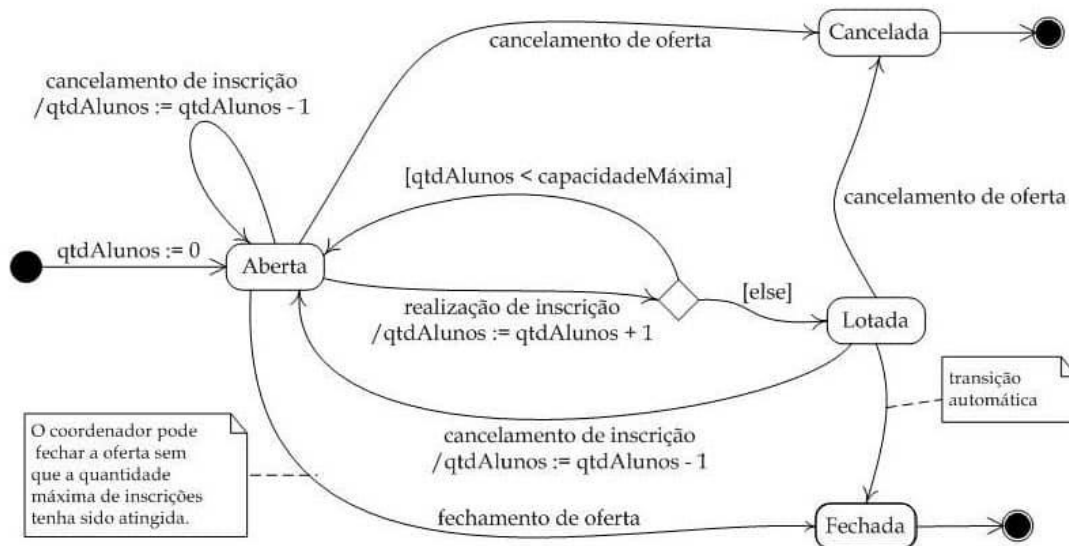
Atenção

Assim como diagramas de interação não são construídos para todos os casos de uso, os diagramas de transição de estado não devem ser construídos para todas as classes de um sistema, mas apenas para aquelas que possuem um número definido de estados conhecidos, e quando o comportamento das classes de objetos é afetado e modificado

pelos diferentes estados. Nessas classes, um DTE pode ser utilizado para enfatizar os eventos que resultam em mudanças de estado.

DIAGRAMA DE TRANSIÇÃO DE ESTADOS

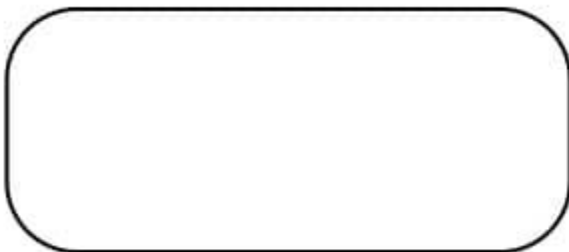
A UML tem um conjunto rico de notações para desenhar um DTE. Os elementos básicos de um diagrama de transição de estados são os estados e as transições. Associados a estas últimas estão os conceitos de **evento**, **ação** e **atividade**. Um DTE pode também conter elementos menos utilizados, mas às vezes úteis, como **transições internas**, **estados aninhados** e **estados concorrentes**. A figura a seguir mostra um diagrama de estados para o objeto oferta de disciplina, que pode ter quatro estados: **aberta**, **cancelada**, **lotada** e **fechada**.



Exemplo de diagrama de estados.

Um estado é uma situação na vida de um objeto durante a qual ele satisfaz alguma condição ou realiza alguma atividade. Cada estado de um objeto é normalmente determinado pelos valores dos seus atributos e (ou) pelas suas ligações com outros objetos.

A UML utiliza as seguintes notações para representar os estados:



Estado

O retângulo com as bordas arredondadas representa um estado. Em sua forma mais simples, o retângulo de um estado possui um único compartimento.



Estado inicial

O círculo preenchido indica o estado de um objeto quando ele é criado. Só pode haver um estado inicial em um DTE, restrição que serve para definir a partir de que ponto um DTE deve começar a ser lido.



Estado final

Representado por um círculo parcialmente preenchido. Diferentemente do estado inicial, pode haver vários estados finais, dependendo dos estados terminais do objeto modelado.

Transições

Os estados estão associados a outros pelas transições. Uma transição é mostrada como uma linha conectando estados, com uma seta apontando do estado origem para o estado destino. Quando ocorre uma transição entre estados, diz-se que a transição foi disparada. Note que, em uma transição, o estado subsequente pode ser igual ao estado original. Uma transição pode ser rotulada com uma expressão:

evento (lista-parâmetros) [guarda] / [ação]

A seguir, vamos entender os elementos envolvidos nas transições entre estados:
Clique na barra para ver as informações.

EVENTO

Um evento é algo que acontece em algum ponto no tempo e que pode modificar o estado de um objeto, provocando a transição de estado. Um evento pode conter uma lista de parâmetros que são passados para fornecer informações úteis ao objeto receptor de evento. A lista de parâmetros é especificada entre parênteses. Na figura **exemplo de diagrama de estados**, *realização de inscrição / qtdAlunos := qtdAlunos + 1* é um exemplo de evento.

Os eventos relevantes a uma classe de objetos podem ser classificados em quatro tipos:

1. **Evento de chamada:** recebimento de uma mensagem de outro objeto. Pode-se pensar nesse tipo de evento como uma solicitação de serviço de um objeto a outro.
2. **Evento de sinal:** recebimento de um sinal. Este é um tipo especial do evento anterior. Nesse evento, o objeto recebe um sinal de outro objeto que pode fazê-lo mudar de estado. A diferença básica entre o evento de sinal e o evento de chamada é que neste último o objeto que envia a mensagem fica esperando a execução dela. No evento de sinal, o objeto remetente continua o seu processamento após ter enviado o sinal. O evento de sinal raramente é utilizado.
3. **Evento temporal:** passagem de um intervalo de tempo predefinido. Em vez de receber uma mensagem específica, um objeto pode interpretar a passagem de um certo intervalo de tempo como sendo um evento.
4. **Evento de mudança:** uma condição que se torna verdadeira. O fato de determinada condição se tornar verdadeira também pode ser visto como um evento.

CONDIÇÃO DE GUARDA

Uma transição pode ter também uma condição de guarda. Uma condição de guarda, ou simplesmente guarda, é uma expressão de valor lógico, da mesma forma que a utilizada para diagramas de interação. A condição de guarda de uma transição pode ser definida utilizando-se parâmetros passados no evento e atributos e referências a ligações da classe em questão. Além disso, uma condição também pode testar o valor de um estado. Uma transição na qual foi definida uma condição de guarda é disparada somente se o evento associado ocorre e a condição de guarda é verdadeira. Se uma transição não tiver uma condição de guarda, ela sempre será disparada quando o evento ocorrer. É importante não confundir a condição de guarda de uma transição com um evento de mudança (que também é definido com uma expressão de valor lógico). A expressão condicional de uma condição de guarda é sempre apresentada entre colchetes, como em *[qtdAlunos < capacidadeMáxima]* e *[else]*, visto no **exemplo de diagrama de estados**.

AÇÕES

Ao transitar de um estado para outro, um objeto pode realizar uma ou mais ações. Uma ação é uma expressão que pode ser definida em termo dos atributos, das operações ou das associações da classe. Os parâmetros do evento também podem ser utilizados. Uma ação pode corresponder igualmente à execução de uma operação. A ação é representada na linha da transição e deve ser precedida por uma barra inclinada para a direita (símbolo "/"). Note que a ação associada a uma transição é executada somente se a transição for disparada. No **exemplo de diagrama de estados**, são exemplos de ações: */ qtdAlunos := qtdAlunos + 1* e */ qtdAlunos := qtdAlunos - 1*.

ATIVIDADES

Semelhante a uma ação, uma atividade é algo que é executado pelo objeto. No entanto, uma atividade pode ser interrompida (considera-se que o tempo de execução de uma ação é tão insignificante que ela não pode ser interrompida). Por exemplo, enquanto a atividade estiver em execução, pode acontecer um evento que a interrompa e cause uma mudança de estado. Outra diferença entre ação e atividade é que uma atividade sempre está associada a um estado (ao contrário, uma ação está associada a uma transição).

PONTO DE JUNÇÃO

Em algumas situações, o próximo estado de um objeto varia de acordo com o valor da condição de guarda. Se o valor da condição de guarda for verdadeiro, o objeto vai para

um estado E1; se for falso, o objeto vai para outro estado E2. É como se a transição tivesse bifurcações e cada transição de saída da bifurcação tivesse uma condição de guarda. Essa situação pode ser representada em um DTE por um ponto de junção. Um ponto de junção é desenhado como um losango em que chegam uma ou mais transições (provenientes de estados diferentes) e de onde partem uma ou mais transições. A cada transição de saída do ponto de junção está associada uma condição de guarda. A transição que o objeto segue é aquela para a qual a condição de guarda é verdadeira. Note que há um ponto de junção no **exemplo de diagrama de estados**.

ESTADOS ANINHADOS

Podem existir estados aninhados dentro de outro estado. Um estado que contém diversos outros é dito composto. Todos os estados dentro de um estado composto herdam qualquer transição deste último. Isso significa que o uso de estados compostos geralmente torna um DTE mais legível.

ESTADOS CONCORRENTES

Um estado concorrente é um tipo especial de estado composto. Um objeto em um estado concorrente pode, na verdade, se encontrar em dois ou mais estados independentes.

Identificação dos elementos de um diagrama de estados

Os estados podem ser vistos como uma abstração dos atributos e das associações de um objeto. A seguir temos alguns exemplos (os nomes em **negrito** são possíveis estados).

Um professor está **licenciado** quando não está ministrando curso algum durante o semestre.

Uma oferta de disciplina está **aberta** quando a capacidade máxima não for alcançada; caso contrário, a oferta estará **lotada**.

Um professor está **licenciado** quando não está ministrando curso algum durante o semestre.

Um pedido está **atendido** quando todos os seus itens estão entregues.

Um bom ponto de partida para identificar estados de um objeto é analisar os possíveis valores de seus atributos e as ligações que ele pode realizar com outros objetos. No entanto, a existência de atributos ou ligações não é suficiente para justificar a criação de um DTE para uma classe. O comportamento de objetos dessa classe deve depender de tais atributos ou ligações. Em relação a transições, devemos identificar os eventos que podem gerá-las. Além disso, é preciso examinar também se há algum fator que condicione o disparo da transição. Se existir, esse fator deve ser modelado como uma condição de guarda da transição. Já que as transições dependem de eventos para ocorrer, devem-se identificar esses eventos primeiramente.

Outro bom ponto de partida é a descrição dos casos de uso. Os eventos encontrados na descrição dos casos de uso são externos ao sistema. Contudo, uma transição pode também ser disparada por um evento interno ao sistema. Para identificar os eventos internos relevantes a um objeto, analise os **diagramas de interação**. Esses diagramas contêm mensagens sendo trocadas entre objetos e mensagens, que nada mais são do que solicitações de um objeto a outro. Essas solicitações podem ser vistas como eventos. De uma forma geral, cada operação com visibilidade pública de uma classe pode ser vista como um evento em potencial.

Dica

Outra fonte para identificação de eventos relevantes é analisar as regras de negócio definidas para o sistema. Normalmente, essas regras possuem informações sobre condições limites que permitem identificar estados e transições.

CONSTRUÇÃO DE DIAGRAMAS DE TRANSIÇÕES DE ESTADOS

Para sistemas bastante simples, a definição dos estados de todos os objetos não é tão trabalhosa. No entanto, a quantidade de estados possíveis de todos os objetos de um sistema complexo é grande. Isso torna a construção de um DTE para o sistema inteiro uma tarefa bastante complexa.

Para resolver o problema da explosão exponencial de estados, os diagramas de estados são desenhados por classe.

Geralmente, cada classe é simples o suficiente para que o diagrama de estados correspondente seja compreensível. Note, contudo, que essa solução de dividir a modelagem de estados por classes do sistema tem a desvantagem de dificultar a visualização do estado do sistema como um todo. Essa desvantagem é parcialmente compensada pela construção de diagramas de interação.

Nem todas as classes de um sistema precisam de um DTE. Um diagrama de estados é desenhado somente para classes que exibem um comportamento dinâmico relevante. A relevância do comportamento de uma classe depende muito de cada situação em particular. No entanto, objetos cujo histórico precisa ser rastreado pelo sistema são típicos para se construir um diagrama de estados. Uma vez selecionadas as classes para cada uma das quais se deve construir um diagrama de estados, acompanhe essa sequência de passos:

1. Identifique os estados relevantes para a classe.
2. Identifique os eventos relevantes aos objetos de uma classe. Para cada evento, identifique qual transição que ele ocasiona.
3. Para cada estado, identifique as transições possíveis quando um evento relevante ocorre.
4. Para cada estado, identifique os eventos internos e as ações correspondentes relevantes.
5. Para cada transição, verifique se há fatores que influenciam o seu disparo. Se esse for o caso, uma condição de guarda deve ser definida.
6. Verifique também se alguma ação deve ser executada quando uma transição é disparada (ações).
7. Para cada condição de guarda e para cada ação, identifique os atributos e as ligações que estão envolvidos. Pode ser que esses atributos ou ligações ainda não existam. Nesse caso, eles devem ser adicionados ao modelo de classes.
8. Defina o estado inicial e os eventuais estados finais.
9. Desenhe o diagrama de estados. Procure posicionar os estados de tal forma que o ciclo de vida do objeto possa ser visualizado de cima para baixo e da esquerda para a direita.

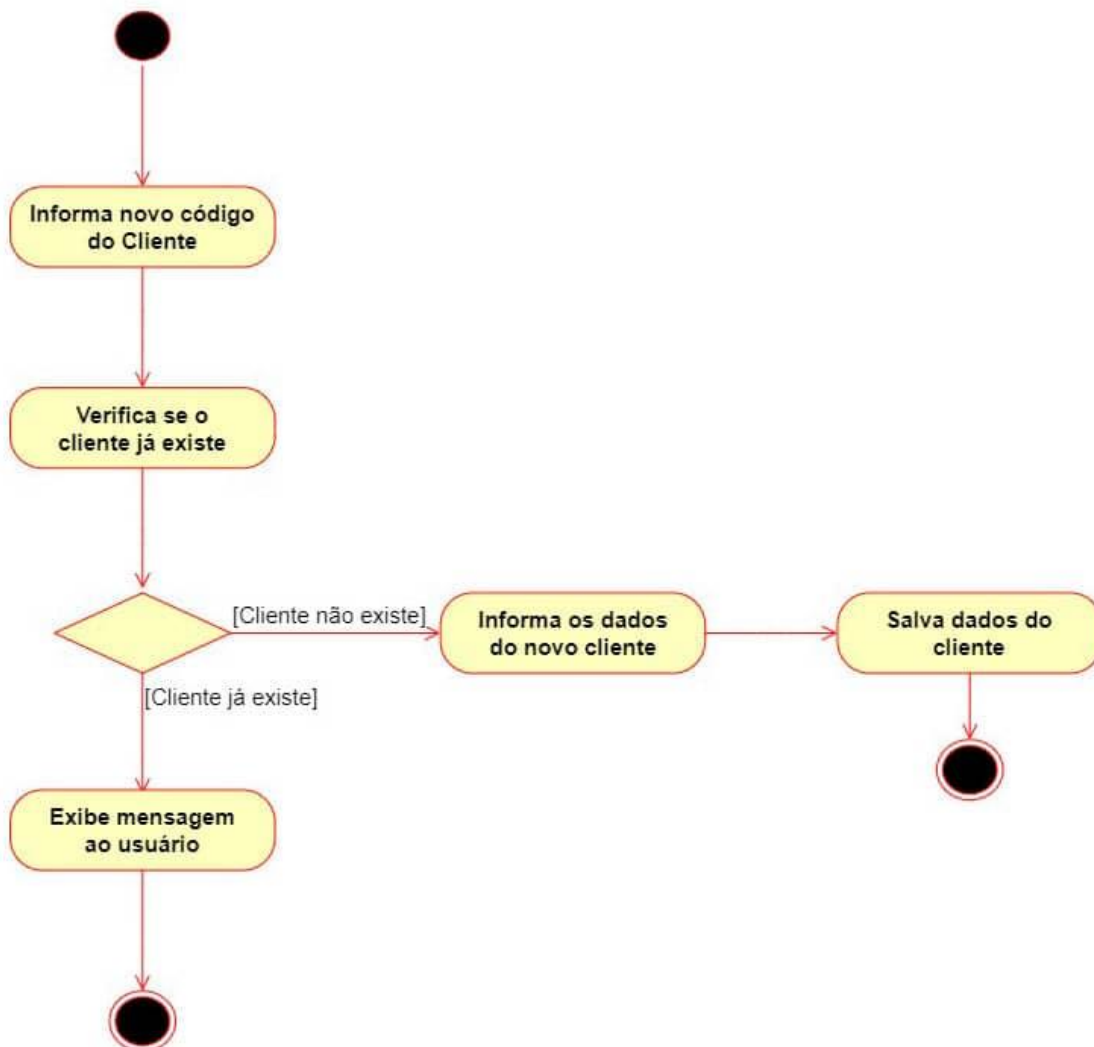
Atenção

A construção de diagramas de estados de um sistema frequentemente leva à descoberta de novos atributos para uma classe, principalmente atributos que servem de abstrações para estados. Além disso, esse processo de construção permite identificar novas operações na classe, pois os objetos precisam reagir aos eventos que eles recebem, e essas reações são realizadas mediante operações.

DIAGRAMA DE ATIVIDADES

Um diagrama de atividade pode ser considerado um tipo especial de diagrama de estados, em que são representados os estados de uma atividade em vez dos estados de um objeto. Ao contrário dos diagramas de estados, que são orientados a eventos, diagramas de atividade são orientados a fluxos de controle.

Muitos autores dizem que o diagrama de atividades é um fluxograma. Na verdade, o diagrama de atividade pode ser visto como uma extensão dos fluxogramas. Além de possuir toda a semântica existente em um fluxograma (com notação ligeiramente diferente), o diagrama de atividade possui notação para representar ações concorrentes (paralelas), juntamente com a sua sincronização.



Exemplo de diagrama de atividades. Elaborado na ferramenta Astah UML.

Os elementos de um diagrama de atividade podem ser divididos em dois grupos:

Elementos utilizados para representar fluxos de controle sequenciais.

Elementos utilizados para representar fluxos de controle paralelos.

Veremos mais detalhes de cada tipo a seguir.

Fluxo de controle sequencial

Os elementos de um diagrama de atividade usados no controle sequencial são:
Clique na barra para ver as informações.

ESTADO AÇÃO E ESTADO ATIVIDADE

Um diagrama de atividades exhibe os passos de uma computação. Cada estado corresponde a um dos passos da computação, em que o sistema está realizando algo. Um estado em um diagrama de atividade pode ser um estado atividade ou um estado ação. O primeiro leva um certo tempo para ser finalizado. Já o segundo é realizado instantaneamente.

ESTADOS INICIAL E FINAL E CONDIÇÃO DE GUARDA

Assim como no diagrama de estados, um diagrama de atividade deve ter um estado inicial; ele pode ter também vários estados finais e guardas associados a transições. Um diagrama de atividade pode não ter estado final, o que significa que o processo ou procedimento sendo modelado é cíclico.

TRANSIÇÃO DE TÉRMINO

Uma transição de término liga um estado a outro. Essa transição significa o término de um passo e o consequente início do outro. Observe que, em vez de ser disparada pela ocorrência de um evento (como nos diagramas de estados), essa transição é disparada pelo término de um estado de ação. Pode haver uma transição rotulada com a condição de guarda especial *[else]*, o que significa que, se todas as demais condições de guarda foram avaliadas como falsas, a transição associada a essa guarda especial é disparada.

PONTOS DE RAMIFICAÇÃO

Um ponto de ramificação possui uma única transição de entrada e várias de saída. Para cada transição de saída, há uma condição de guarda associada. Quando o fluxo de controle chega a um ponto de ramificação, uma e somente uma das condições de guarda deve ser verdadeira.

PONTO DE UNIÃO

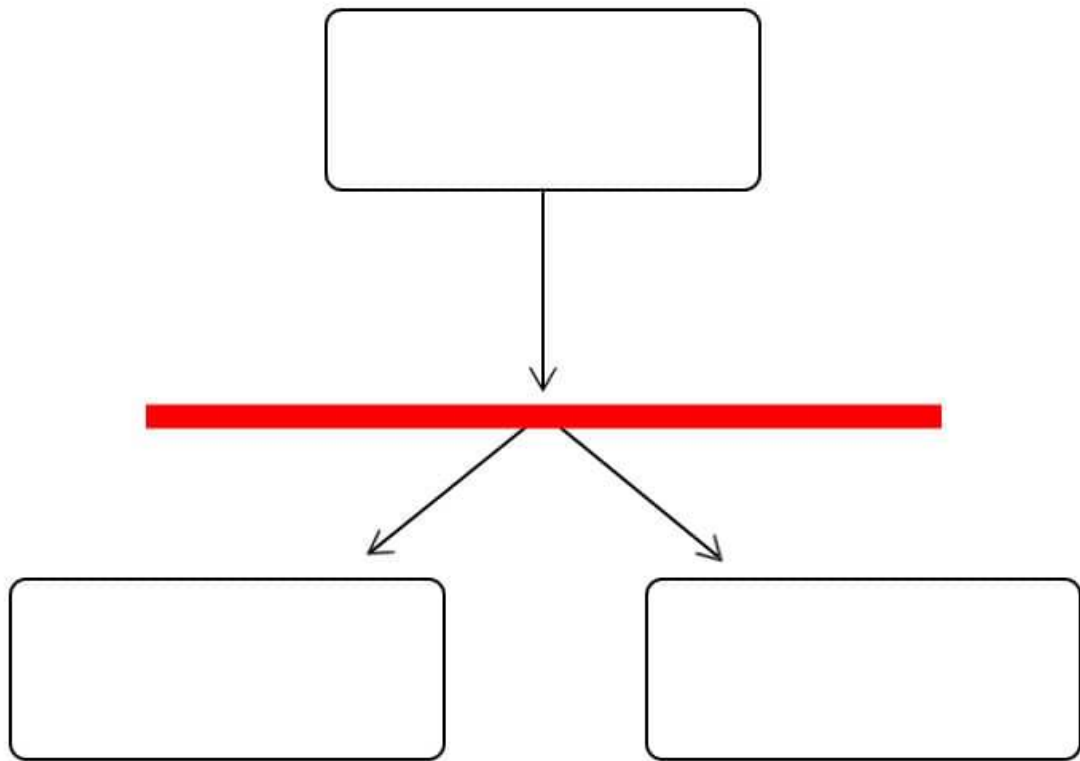
Um ponto de união reúne diversas transições que, direta ou indiretamente, têm um ponto de ramificação em comum.

Fluxo de controle paralelo

Um diagrama de atividade pode conter fluxos de controle paralelos. Isso significa que é possível haver dois ou mais fluxos de controle sendo executados simultaneamente em um diagrama de atividades. Para sincronizar dois ou mais fluxos paralelos, as barras de sincronização são utilizadas. Há dois tipos de barra de sincronização:

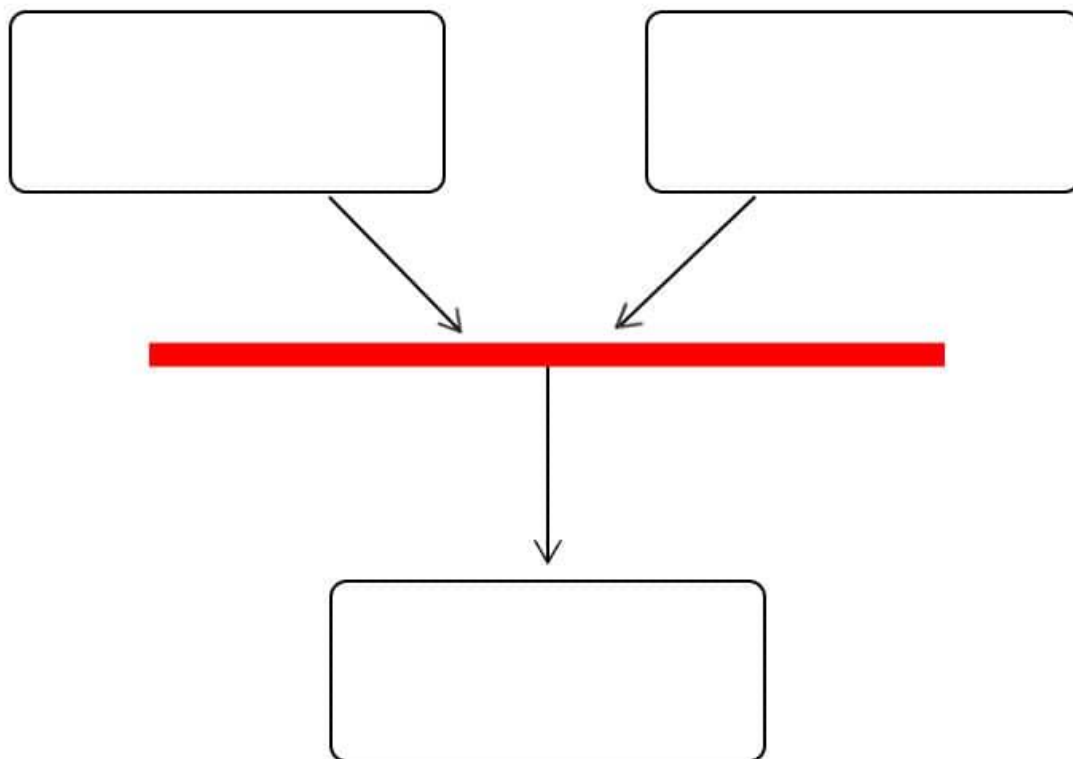
Barra de bifurcação (*fork*)

Uma barra de bifurcação recebe uma transição de entrada e cria dois ou mais fluxos de controle paralelos. A partir desse momento, cada um dos fluxos criados é executado independentemente e em paralelo com os demais.



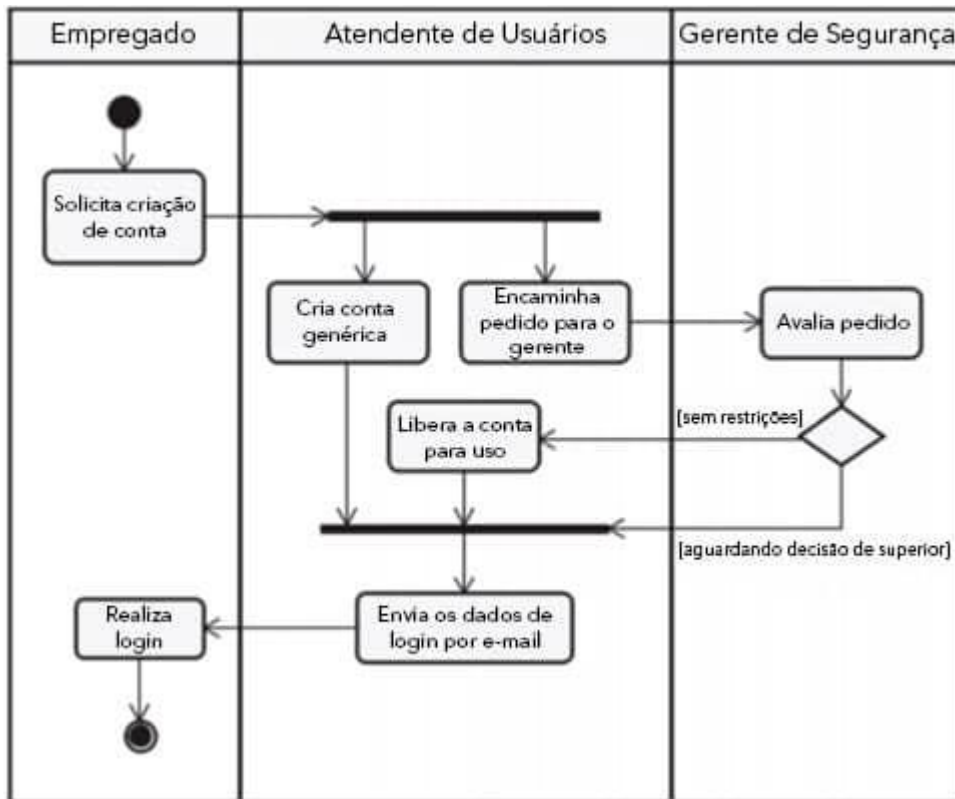
Barra de junção (**join**)

Uma barra de junção recebe duas ou mais transições de entrada e une os fluxos de controle em um único fluxo. Essa barra tem o objetivo de sincronizar fluxos de controle paralelos criados anteriormente no diagrama. As transições de saída da barra de junção somente são disparadas quando todas as transições de entrada tiverem sido disparadas.



Raias

Algumas vezes, as atividades de um processo podem ser distribuídas por vários agentes que o executarão. Isso normalmente ocorre em processos de negócio de uma organização, onde as tarefas são executadas por pessoas ou departamentos diversos. Nesses casos, o processo pode ser representado em um diagrama de atividades com o uso de raias de natação. As raias de natação dividem o diagrama de atividades em compartimentos. Cada compartimento contém atividades que são realizadas por um agente específico. Além disso, as entidades de cada compartimento podem estar realizando atividades em paralelo.



Exemplo de diagrama de atividades com raias e *fork*. Elaborado na ferramenta Astah UML.

MODELAGEM DA LÓGICA DE UMA OPERAÇÃO COMPLEXA

Quando um sistema de software é adequadamente decomposto em seus objetos constituintes e as responsabilidades de cada objeto estão bem definidas, a maioria das operações é bastante simples. Essas operações não necessitam de modelagem gráfica para serem entendidas. No entanto, em alguns casos, notadamente quando uma operação de uma classe de controle implementa uma regra de negócio, pode haver a necessidade de se descrever a lógica dessa operação ou da própria regra de negócio. Diagramas de atividade também podem ser utilizados com esse objetivo, nisso se assemelhando a um fluxograma.