

Diagrama de Casos de Uso

*Ricardo R. Gudwin
DCA-FEEC-UNICAMP
18/10/2010*

Casos de Uso

Casos de uso são abstrações de pequenas histórias narrativas envolvendo a interação entre um ou mais usuários (chamados de atores) e o sistema. A idéia é que estes casos de uso representem, por meio dessas pequenas histórias, as funcionalidades de um sistema. Imagina-se um conjunto de atores necessários a operar o sistema e passa-se a descrever o fluxo dos acontecimentos, onde o ator executa uma ação, e o sistema responde de alguma maneira a essa ação, até que alguma funcionalidade tenha sido contemplada. Especificar um caso de uso é como observar em nossa mente um conjunto de atores imaginários operando o sistema que pretendemos desenvolver, e descrever o que esses atores fazem e como o sistema responde. Normalmente utiliza-se somente um ator interagindo na maioria dos casos de uso, mas há casos em que mais de um ator podem ser necessários. Imagine um sistema de caixa de padaria, em que o caixa (o primeiro ator) executa a maioria das ações de acesso ao sistema, mas o cliente (o segundo ator) precisa em algum instante digitar sua senha de cartão de crédito, caso faça o pagamento com cartão. Para esse caso de uso, teríamos dois atores necessários para que a funcionalidade “Pagamento com Cartão de Crédito” pudesse ser implementada. Apesar da grande maioria de casos de uso demandar somente um único ator, pode ser que diferentes atores, representando diferentes papéis de usuários do sistema tenham diferentes níveis de permissão de acesso às funcionalidades. Dessa forma, um caso de uso como “Fazer o Login no Sistema” talvez possa ser realizado por um ator chamado “Usuário Não-Authenticado”, mas os demais casos de uso demandem um ator chamado “Usuário Autenticado”. Outros casos de uso, ainda, podem demandar um ator chamado “Administrador”.

Podemos entender um caso de uso, portanto, como uma sequência de ações (uma atividade, segundo a terminologia UML), executadas na forma de uma interação entre os atores e o sistema, onde ao final da atividade existe alguma vantagem ou ganho para o(s) usuário(s). Observe que não basta se descrever uma atividade para que esta seja um caso de uso. É necessário que haja um benefício direto ao usuário em função dessa atividade. Existem atividades que são casos de uso, e existem atividades que são somente sub-atividades de um caso de uso. O que determina que uma atividade seja um caso de uso é o benefício decorrente do resultado dessa atividade. Dessa forma, dizemos que um caso de uso é equivalente a uma funcionalidade do sistema. Essa equivalência entre “casos de uso” e “funcionalidades” é um dos princípios que orienta o “Processo Unificado” de desenvolvimento de software. Dessa forma, o Processo Unificado captura (ou elicita) os requisitos, por meio da descoberta e especificação de diferentes casos de uso. Para representar uma coleção de casos de uso, suas possíveis relações e quais os atores envolvidos com cada caso de uso, desenvolveu-se o diagrama de casos de uso UML.

Diagrama de Casos de Uso

Os diagramas de casos de uso mostram atores e casos de uso juntos com seus relacionamentos. Um exemplo de um diagrama de casos de uso pode ser visto na figura 1 a seguir:

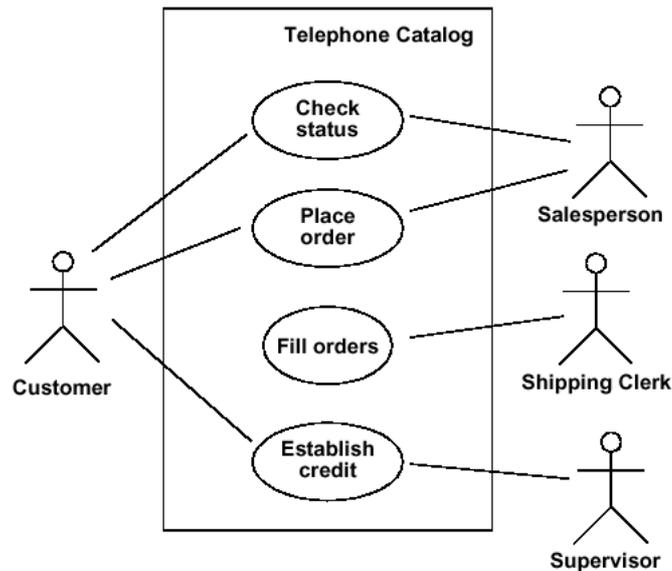


Figura 1: Exemplo de Diagrama de Casos de Uso

O motivo da existência dos diagramas de caso de uso são basicamente os seguintes. Em primeiro lugar, representar quais os atores envolvidos com cada caso de uso. Em segundo lugar, o diagrama de casos de uso permite que se represente possíveis relações entre os casos de uso. Mas talvez o mais importante motivo para termos os diagramas de casos de uso é o fato de que estes diagramas são os únicos diagramas, em toda a documentação desenvolvida no Processo Unificado que representam o sistema em sua totalidade. Os diagramas de caso de uso permitem uma “visão geral” do sistema, onde todas as suas funcionalidades estão concentradas. Esta visão geral permite aos desenvolvedores apreciar, durante a fase de especificação do sistema, se o sistema está suficientemente especificado, ou se faltam ainda funcionalidades necessárias. Dessa forma, torna-se um diagrama bastante importante e útil durante a fase de especificação do sistema.

Pontos de Extensão

Casos de uso podem ter pontos de extensão, ou seja, referências a uma localização dentro de fluxo de atividades onde sequências de ações de outros casos de uso podem ser inseridas. Cada ponto de extensão tem um único nome dentro de um caso de uso. Um exemplo de casos de uso com pontos de extensão é mostrado na figura 2 a seguir:

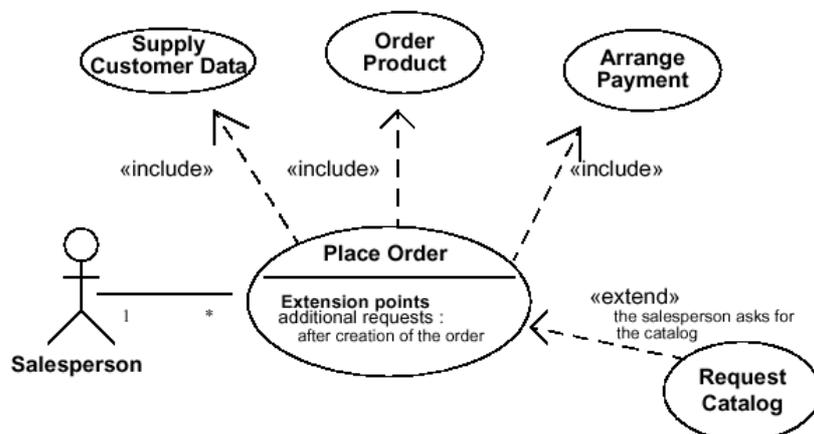


Figura 2: Caso de Uso com Ponto de Extensão

Relações entre Casos de Uso

Durante a fase de especificação dos requisitos, usualmente se levanta o maior número possível de casos de uso e os atores que dele participam. Estes casos de uso são considerados de maneira independente uns dos outros. O resultado é alguma coisa como o diagrama da figura 3 a seguir.

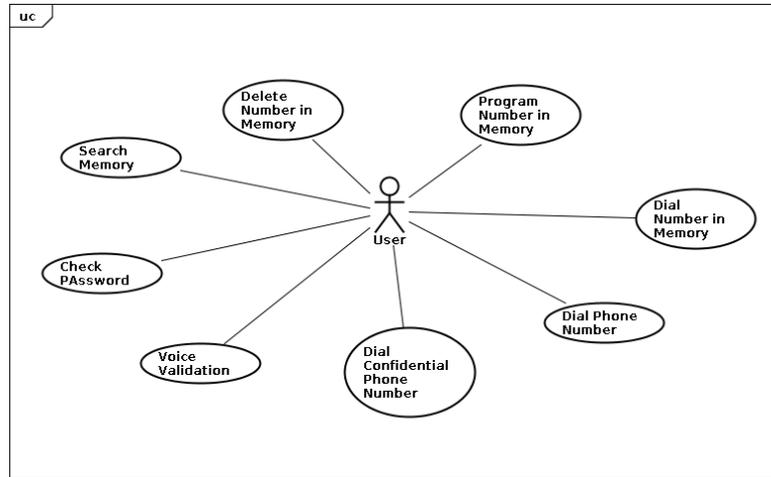


Figura 3: Diagrama de Casos de Uso Inicial

À medida que o conhecimento sobre o sistema vai crescendo, entretanto, e o número de casos de uso passa a crescer também, torna-se necessário estruturar o diagrama de casos de uso, e passar a considerar as relações que os casos de uso podem ter uns com os outros, bem como a relação que os atores podem ter uns com os outros. Durante a etapa do detalhamento dos casos de uso, pode ser que algum caso de uso em particular tenha se tornado demasiadamente complexo ou detalhado, quando então se torna útil desmembrá-lo em mais de um caso de uso. Da mesma forma, pode ser que se tenha percebido que muitos casos de uso possuem uma estrutura comum, que se repete diversas vezes. Dessa forma, torna-se necessário fazer um *refactoring* dos casos de uso, de tal forma que os mesmos sejam repensados e devidamente estruturados, de modo a gerar um modelo que seja homogêneo, consistente e simples de ser interpretado.

Basicamente, 4 tipos de relacionamentos podem ser indicados em diagramas de casos de uso:

- Associações
- *Extend*
- *Include*
- Generalização

As **associações** denotam a participação de atores em um caso de uso. É o único tipo de relacionamento entre um ator e um caso de uso. Um relacionamento do tipo **extend** é uma relação entre um caso de uso A para um caso de uso B que indica que uma instância do caso de uso B pode ser aumentada (sujeita a condições específicas de extensão) pelo comportamento especificado por A. Esse comportamento é inserido conforme especificado por um ponto de extensão em B. Um relacionamento do tipo **include** é uma relação entre um caso de uso A para um caso de uso B que indica que uma instância do caso de uso A contém o comportamento especificado por B. Esse comportamento é incluído na localização indicada em A por um ponto de extensão. Um relacionamento do tipo **generalização** entre um caso de uso A e um caso de uso B indica que A é uma especialização de B. Normalmente, essa generalização implica em que B é uma descrição mais abstrata de uma situação, e A é uma descrição mais detalhada. Além dos casos de uso, é possível

utilizarmos generalizações para indicarmos o relacionamento entre dois atores. Um exemplo deste tipo é apresentado na figura a seguir:

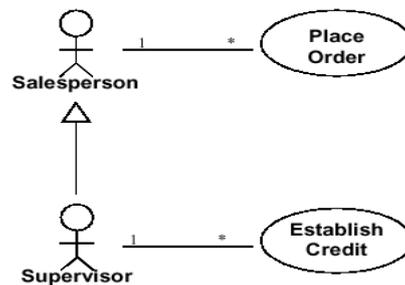


Figura 4: Generalização entre Atores

Observe que nesse caso, o ator Supervisor é um tipo de Salesperson, ou seja, uma especialização de um Salesperson. Observe ainda pelo exemplo que as associações entre atores e casos de uso podem também conter uma cardinalidade.

As associações entre os atores e os casos de uso podem ainda possuir uma navegabilidade (seta entrando ou saindo do caso de uso). Essa navegabilidade indica quem inicia o caso de uso. Caso a seta seja do ator para o caso de uso, é o ator quem inicia a interação. Caso seja do caso de uso para o ator, é o sistema quem inicia a interação. Veja o exemplo na figura 5 a seguir:



Figura 5: Exemplo de Navegabilidade em Casos de Uso

Neste exemplo, que mostra o caso de uso Participação em Conferência Eletrônica, é o professor quem inicia as atividades. O aluno passa a interagir posteriormente, a partir da iniciativa do sistema.

Durante o detalhamento dos casos de uso (que, assumiremos neste texto, seja feito por meio de um diagrama de atividades), podem haver atividades ou partes de atividades que são semelhantes em diversos casos de uso. Pode-se detectar uma situação como essa quando observamos que certas partes dos diagramas de atividades se repetem em mais de um caso de uso. De forma a reduzir a redundância, os casos de uso podem ser então reestruturados para tornar o modelo como um todo mais enxuto. Assim, esses trechos de casos de uso podem se tornar casos de uso independentes, que são reutilizados no modelo por meio da relação de generalização.

Além disso, outra possível análise é tentar identificar descrições adicionais ou opcionais de funcionalidade. Sabemos que o mecanismo de extensão permite a inserção de adições ao comportamento básico de casos de uso. Como vimos, esse relacionamento inclui as condições para a extensão e o ponto de extensão, onde o caso de uso deve ser inserido. Durante a etapa de estruturação de casos de uso, o que se faz é tentar identificar situações desse tipo, alterando o diagrama de casos de uso inicial de modo a incluir relacionamentos de extensão. Normalmente, os diagramas de atividades que detalham esses casos de uso demandam ser retrabalhados também, de forma a refletir as mudanças implementadas.

Por fim, uma última análise que se faz durante a estruturação dos casos de uso é tentar identificar descrições repetidas de funcionalidade. Sabemos que o relacionamento de inclusão permite a inserção incondicional e explícita do comportamento de um caso de uso em outros casos de uso. Nesses casos, tenta-se então verificar se situações desse tipo ocorrem, e nesse caso procede-se à

inserção de relacionamentos do tipo include no diagrama de casos de uso.

Uma dúvida muito frequente que ocorre é a dificuldade em diferenciar os relacionamentos de generalização e de inclusão. Temos que ter muito cuidado pois esses dois tipos de relacionamentos são muito semelhantes. A grande diferença que existe entre esses dois é que no relacionamento de generalização, o caso de uso que é generalizado é um caso de uso mais abstrato que, apesar de semelhante, será diferente em cada uma de suas especializações. Um bom exemplo é o exemplo mostrado na figura 6 abaixo, fruto da estruturação do modelo de casos de uso que apresentamos na figura 3.

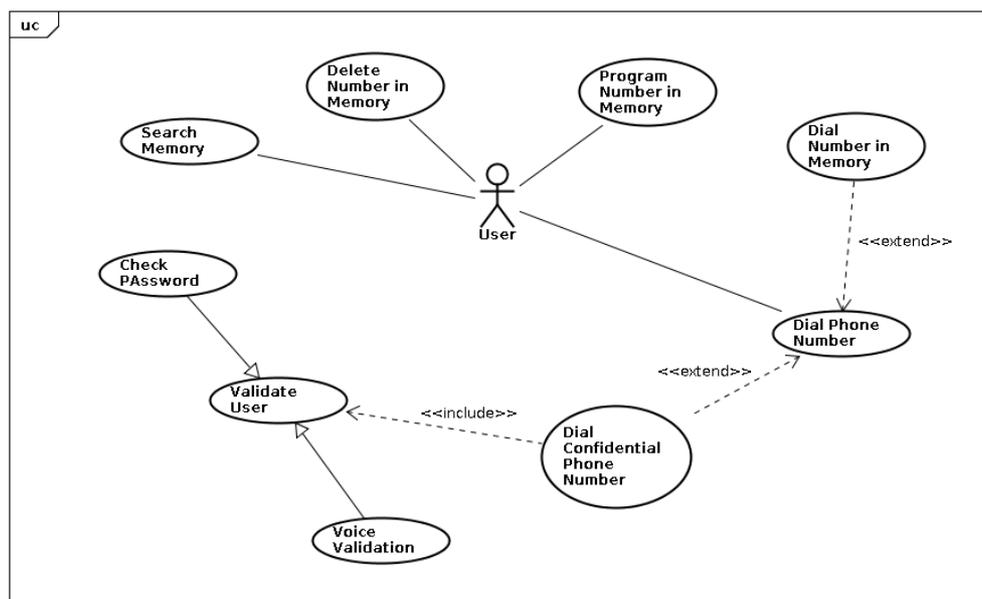


Figura 6: Exemplo de Diagrama de Casos de Uso Estruturado

Observe que anteriormente tínhamos dois casos de uso, o caso de uso *CheckPassword* e o *VoiceValidation*, que apresentavam um compartilhamento de funcionalidades. Ambos serviam para efetuar uma validação do usuário. Uma maneira de refinar nosso modelo foi criar um novo caso de uso abstrato chamado *ValidateUser*, que passou então a ser uma abstração, tanto de *CheckPassword* como de *VoiceValidation*. Para compreendermos a diferença entre o uso de uma generalização e de uma inclusão, observe que a descrição de uma checagem de password não tem nada a ver, em princípio com uma validação vocal. Em termos concretos, eles são casos de uso completamente diferentes. Entretanto, quando abstraímos o que ocorre em cada um desses casos de uso, vemos que ambos nada mais fazem do que validar o usuário. Por isso, usamos uma relação de generalização. O relacionamento do tipo inclusão, ao contrário, inclui totalmente o caso de uso como um sub-trecho de um caso de uso mais complexo. Assim, devemos ter cuidado durante esse refinamento. Em alguns casos, será necessário fazermos abstrações e incluirmos novos casos de uso mais abstratos. Em outros, vamos tentar verificar a mera repetição de situações e utilizaremos inclusões. Observe que no diagrama do exemplo, verificamos que a validação do usuário ocorre somente quando ocorre uma discagem de número confidencial. Por isso, colocamos o caso de uso *ValidateUser* como uma inclusão de *DialConfidentialPhoneNumber*. Por outro lado, tanto *DialConfidentialPhoneNumber* quanto *DialNumberInMemory* estendem a descrição de *DialPhoneNumber*. Por isso, colocamos ambos como extensões de *DialPhoneNumber*.

Repare que no processo de desenvolvimento, após chegarmos ao diagrama de casos de uso estruturado, pode ser necessário que modifiquemos o detalhamento dos casos de uso feito anteriormente, como por exemplo a inclusão de novos casos de uso (vide a inclusão de *ValidateUser* no exemplo).