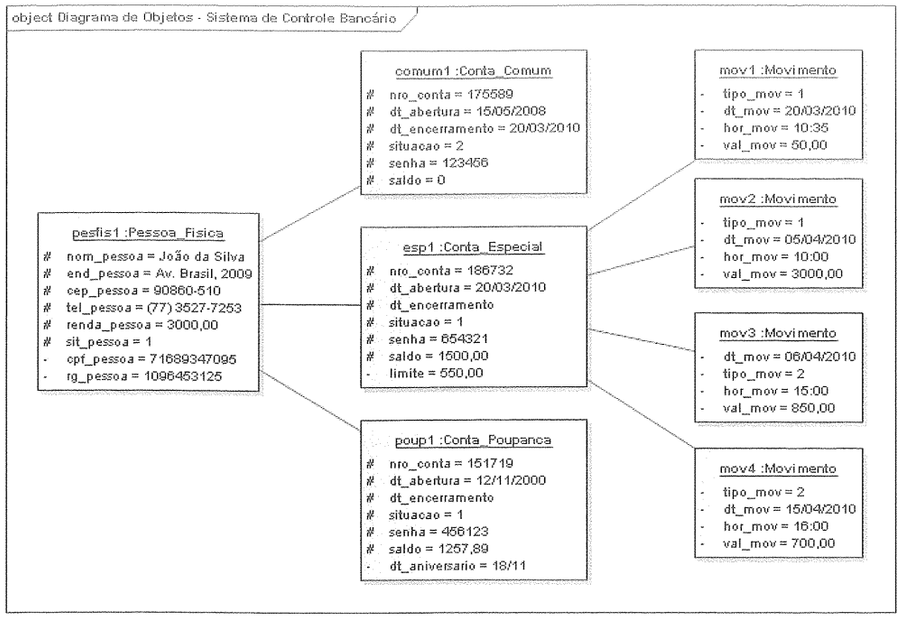

Figura 5.4 – Diagrama de Objetos.

Nesse exemplo percebe-se que o objeto `pesfis1`, da classe `Pessoa_Fisica`, está vinculado a três objetos: o primeiro da classe `Conta_Comum`, o segundo da classe `Conta_Especial` e o terceiro da classe `Conta_Poupanca`, chamados respectivamente de `comum1`, `esp1` e `poup1`. Dessa forma, podemos concluir que a pessoa física representada pelo objeto `pesfis1` possui ou possuiu três contas na instituição bancária. Ao examinarmos o objeto `comum1`, percebemos que essa conta já se encontra encerrada, uma vez que o atributo `situacao` armazena o valor 2, que, por convenção, significa que a conta está inativa, e porque o atributo `dt_encerramento` tem uma data definida. Já os outros dois objetos ainda estão ativos.

Se continuarmos examinando a figura perceberemos que o objeto `esp1` está vinculado a quatro objetos da classe `Movimento`, `mov1`, `mov2`, `mov3` e `mov4`. Os dois primeiros objetos referem-se a um depósito de valores, enquanto os dois últimos a saques.

Como podemos observar, nem todas as classes estão representadas por seus objetos nesse exemplo, o que é, aliás, recomendável. Um diagrama de objetos deve enfocar o menor conjunto possível de classes, porque as classes podem ter um número muito grande de objetos, e representar objetos de todas as classes em um diagrama tende a deixá-lo muito extenso e poluído. Assim, o melhor é criar vários diagramas de objetos enfocando pequenas partes do diagrama de classes.

CAPÍTULO 6

Diagrama de Pacotes

O diagrama de pacotes descreve como os elementos do modelo estão organizados em pacotes e demonstra as dependências entre eles. Esse diagrama é muito útil para a modelagem de subsistemas e para a modelagem de subdivisões da arquitetura de uma linguagem. Pode ser utilizado também para representar um conjunto de sistemas integrados, representados por pacotes, ou ainda os submódulos englobados por um sistema. O diagrama pode ser utilizado ainda para demonstrar a arquitetura de uma linguagem, como ocorre com a própria UML, ou a arquitetura de um processo de desenvolvimento.

6.1 Pacotes

Pacotes são utilizados para agrupar elementos e fornecer denominações para esses grupos. Um pacote pode representar um sistema, um subsistema, uma biblioteca ou uma etapa de um processo de desenvolvimento, entre outras alternativas. Um pacote pode inclusive conter outros pacotes. A figura 6.1 apresenta um exemplo de pacote.

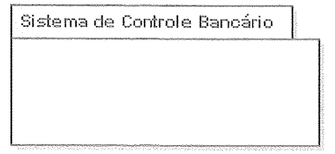

Figura 6.1 – Exemplo de Pacote.

O exemplo apresentado nessa figura contém um pacote intitulado Sistema de Controle Bancário, o que significa que ele engloba os elementos contidos nesse sistema. Esse exemplo apresenta somente o pacote, sem revelar seu conteúdo. No entanto, é possível encontrar pacotes com seu conteúdo ou parte dele explicitamente declarado, como demonstra a figura 6.2.

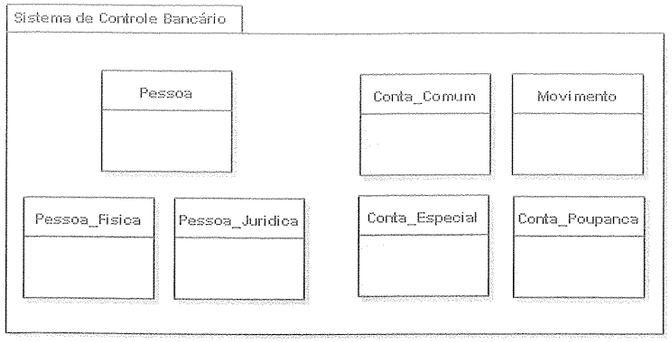

Figura 6.2 – Pacote com Detalhe de seus Membros.

Nesse exemplo, identificamos os elementos contidos pelo pacote, em termos de suas classes. Como o leitor pode notar, não definimos os atributos, métodos ou associações entre as classes. No entanto, isso é perfeitamente possível, podendo um pacote conter um diagrama de outro tipo completo dentro dele. O problema dessa abordagem é o grande espaço ocupado pelo pacote, sendo mais comum encontrar pacotes sem o detalhamento de seu conteúdo. Existe uma notação alternativa para identificar os membros do pacote por meio do conector de aninhamento, representado por um círculo contendo uma cruz, conforme demonstra a figura 6.3.

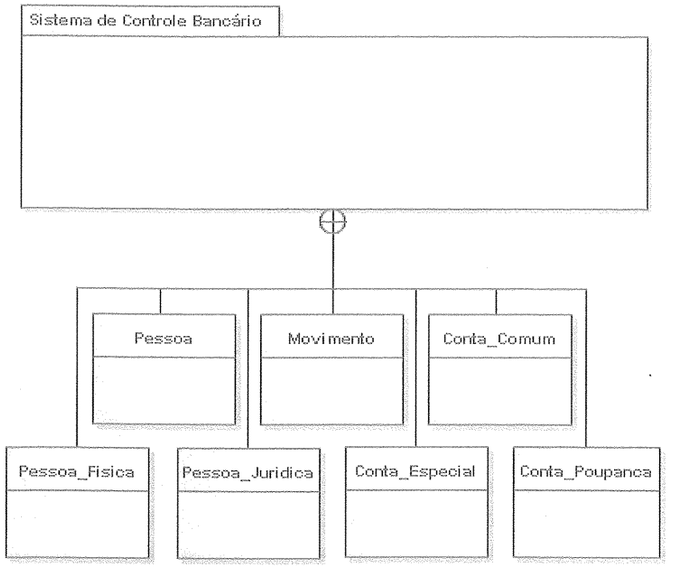

Figura 6.3 – Pacote com Detalhe de seus Membros – Notação Alternativa com Conector de Aninhamento.

6.2 Dependência

Pacotes normalmente contêm dependências entre si. Um relacionamento de dependência informa que o elemento dependente necessita de alguma forma do elemento do qual depende. A figura 6.4 apresenta um exemplo de relacionamento de dependência entre pacotes.

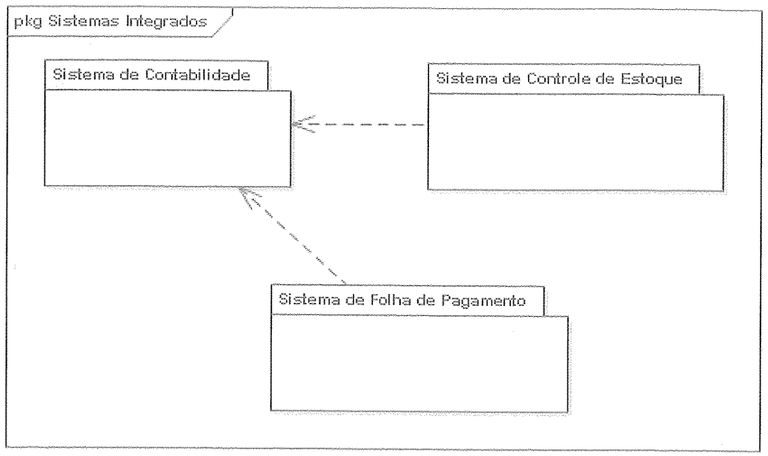

Figura 6.4 – Dependência entre Pacotes.

Nesse exemplo existem três sistemas integrados: o sistema de contabilidade, de estoque e de folha de pagamento. Os pacotes estão associados por meio de dependências, indicando que os sistemas de estoque e de folha de pagamento necessitam do sistema de contabilidade para lançar suas operações financeiras. Outro exemplo de dependência é apresentado na figura 6.5.

Nesse exemplo, os pacotes representam as etapas de um processo de desenvolvimento de software, onde primeiramente se define a interface, em seguida a aplicação propriamente dita e finalmente a forma como os dados serão persistidos fisicamente em disco.

O relacionamento de dependência no diagrama de pacotes pode ter dois estereótipos: <<merge>>, significando que os elementos do pacote que utiliza essa dependência serão unidos aos elementos do outro pacote, e <<import>>, significando que o pacote que utiliza essa dependência está importando alguma característica ou elemento do outro pacote.

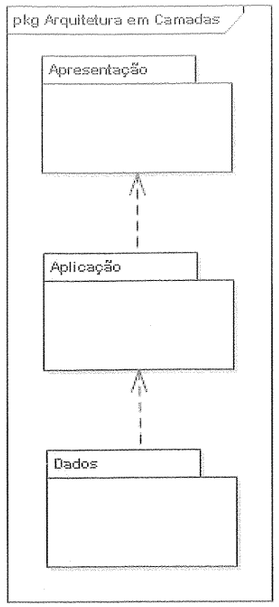

Figura 6.5 – Dependência entre Pacotes.

6.3 Pacotes Contendo Pacotes

É bastante comum, principalmente na definição de arquiteturas de linguagens de modelagem, que um pacote se subdivida em diversos outros pacotes. Isso ocorre, por exemplo, com a Biblioteca de Infraestrutura da UML, conforme a figura 6.6.

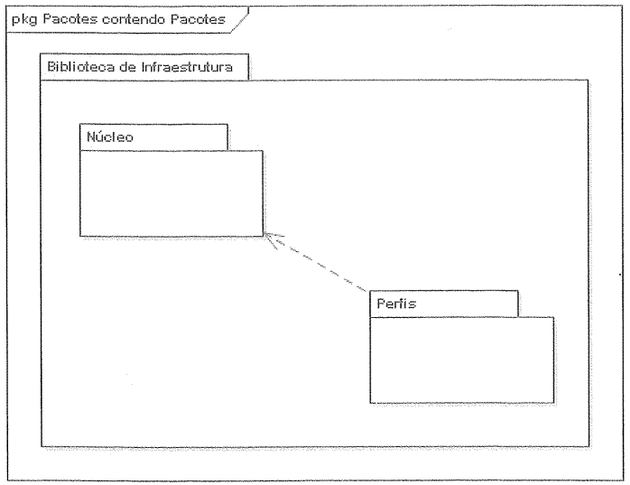

Figura 6.6 – Pacotes Contendo Pacotes.

Nesse exemplo percebemos que o pacote Biblioteca de Infraestrutura contém os pacotes Núcleo e Perfis e que o pacote Perfis tem uma relação de dependência com o pacote Núcleo.

6.4 Estereótipos Aplicados a Pacotes

É possível aplicar estereótipos aos pacotes, deixando claro que estes representam sistemas, subsistemas, frameworks ou modelos por exemplo, conforme apresentado na figura 6.7.

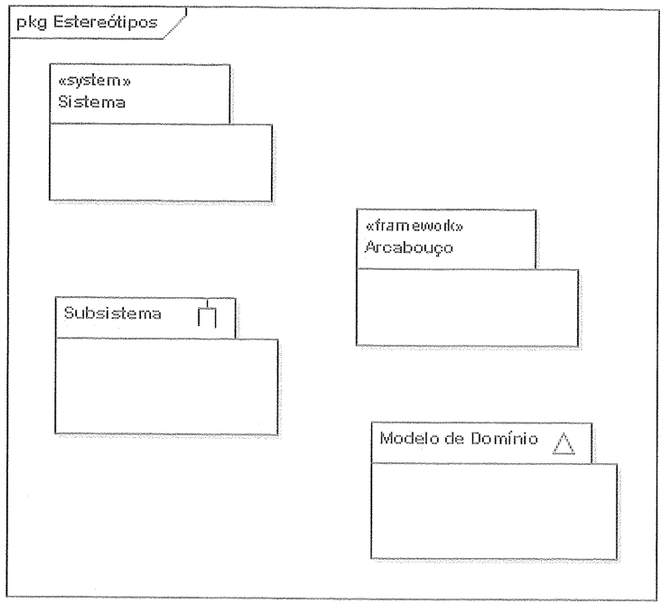

Figura 6.7 – Pacotes com Estereótipos.

Nessa figura apresentamos pacotes com estereótipos dos quatro tipos enunciados. Observe que enquanto os estereótipos <<system>> e <<framework>> são estereótipos de texto, <<subsystem>> e <<model>> são estereótipos gráficos que modificam um pouco o desenho-padrão do pacote.