MAC5715 - TÓPICOS EM POO
PADRÃO GESTOR DE DOCUMENTOS*

Germano Capistrano Bezerra & Silvio do Lago Pereira

25 de outubro de 2002

1 Objetivo

Gerenciar um repositório de documentos, controlando os acessos de leitura e de edição de conteúdo.

2 Motivação

Com o grande volume de documentos que circulam em uma corporação, pública ou privada, faz-se necessário estabelecer metodologias para o gerenciamento da elaboração, aprovação e disponibilização desse material para os integrantes da instituição, seja de forma física ou através do uso de um sistema de software. Como exemplo, podemos citar a necessidade de gerenciar documentos tais como:

- código-fonte de um sistema;
- páginas publicadas em *web sites*;
- peças jurídicas de um escritório de advocacia;
- procedimentos e normas de qualidade de uma corporação (*e.g.* ISO-9000).

Tomando o exemplo da documentação da ISO, há um conjunto de normas que são elaboradas por alguns colaboradores. Após elaboradas, essas normas são apreciadas pela diretoria e aprovadas, ou não, para publicação. Os documentos publicados são acessíveis aos funcionários da corporação. Em qualquer momento, um determinado colaborador pode tomar um documento e fazer alterações, reiniciando mais um ciclo de vida do documento. A abordagem mais simples e imediata para essa situação é disponibilizar o material em um diretório do sistema de arquivos da rede da corporação.

*O formato adotado na apresentação desse padrão segue recomendações encontradas em [1].

3 Contexto

Os documentos ficam disponíveis num repositório de acesso público. Os usuários podem, a qualquer momento, retirar cópias de um dado documento desse repositório para simples leitura ou, então, editá-lo e devolvê-lo ao repositório.

4 Forças

- Como não há controle de acesso, qualquer usuário pode editar um documento.
- Mais de um usuário pode editar um mesmo documento, ao mesmo tempo.
- Leitores não sabem se a versão do documento que estão lendo é a mais recente, nem se ela foi elaborada por uma pessoa devidamente autorizada.
- Revisores não sabem se estão editando a versão corrente do documento.
- Versões mais antigas de um documento podem ser sobrepostas por outras versões mais recentes e, portanto, não há como obter um histórico das modificações realizadas nos documentos.

5 Problema

No contexto apresentado, não há como:

- garantir que um documento seja alterado apenas por pessoas autorizadas, de maneira disciplinada;
- impedir que dois ou mais usuários editem o mesmo documento, simultaneamente;
- identificar a versão corrente de um determinado documento no repositório;
- garantir que todas as versões anteriores de um documento sejam mantidas no repositório.

6 Solução

O próprio documento deve gerenciar as suas versões, permitindo que qualquer uma delas possa ser lida, simultaneamente, por qualquer número de usuários e garantindo que, em cada instante, apenas uma cópia de sua versão corrente, denominada *rascunho*, possa ser editada, de modo exclusivo, por um único usuário.

7 Estrutura

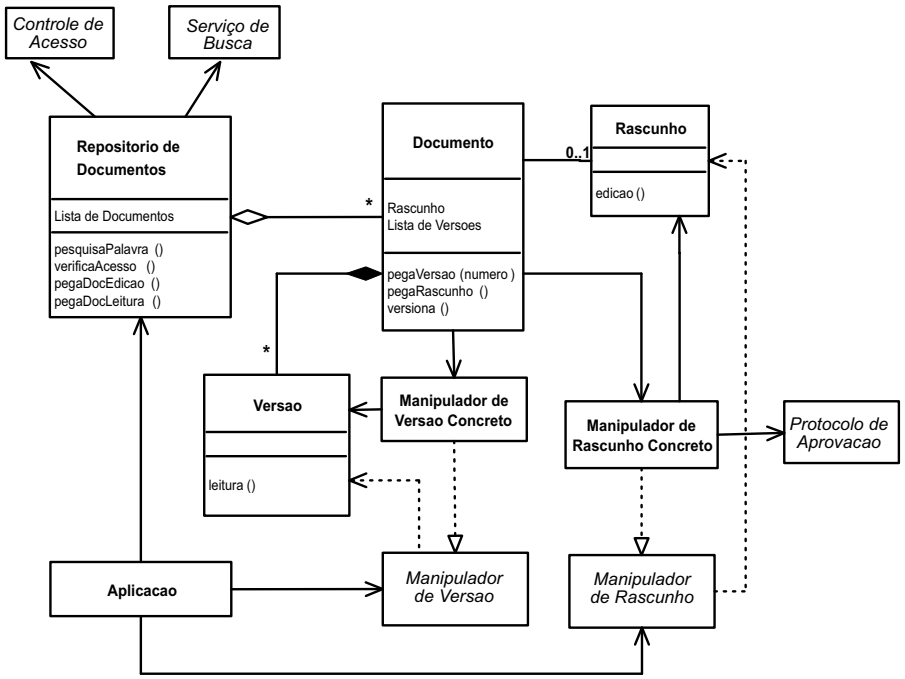

Figura 1: *Diagrama de classes para o padrão Gestor de Documentos*

7.1 Participantes

- Repositorio de Documentos - O repositório é responsável por guardar uma lista com todos os documentos disponíveis para as aplicações clientes. Todas as solicitações de documentos, seja para leitura ou para escrita, são feitas a ele. Assim, o repositório pode fazer uso de serviços estendidos, oferecendo mais funcionalidades às aplicações. Por exemplo, pode-se implementar um mecanismo de controle de acesso, possibilitando que apenas usuários autorizados acessem determinados documentos protegidos. Em outro cenário, pode-se acrescentar um serviço de busca, possibilitando a pesquisa por documentos através de classificadores ou por texto integral, dependendo da implementação fornecida do serviço.

- Documento - O documento é o elemento central do padrão. O conteúdo propriamente dito não é armazenado no documento, apenas alguns metadados de classificação. Em vez disso, o conteúdo é mantido numa coleção de versões que compõe o documento. O papel do documento é manter uma listagem das versões existentes e controlar o uso do rascunho, através de acesso exclusivo dado a um único usuário

por vez. O documento é responsável por instanciar os manipuladores de documentos. Há dois tipos de manipuladores: um para escrita, que manipula objetos do tipo Rascunho, e outro para leitura, que manipula objetos do tipo Versao. Esses manipuladores são os elementos de interação dos clientes. Os clientes devem conhecer uma interface abstrata de manipulação, sendo os manipuladores concretos instanciados por cada documento. Assim, dependendo do tipo ou formato do documento, pode ser instanciado um determinado manipulador, com suas devidas especializações.

- Versao - Objetos do tipo Versao são responsáveis por armazenar o conteúdo do documento para leitura. Todas as versões publicadas são mantidas pelo Documento. Assim, o usuário pode simplesmente requisitar a última versão ou solicitar um determinado histórico de versionamento do documento.
- Rascunho - Objetos do tipo Rascunho diferem dos de Versao por oferecerem uma interface para edição do conteúdo. Ademais, Documento só permite que um usuário por vez faça sua instanciação. Vários rascunhos podem existir, se associados a diferentes documentos.
- Manipulador de Versao e Manipulador de Rascunho - Os manipuladores, como mencionado acima, apenas fornecem um meio de acesso aos objetos Versao e Rascunho. Em particular, manipuladores concretos podem definir comportamentos específicos, como o uso de um determinado editor de textos ou chamadas a um protocolo de aprovação, no caso do Manipulador de Rascunho.

7.2 Colaborações

Nos procedimentos de leitura, segue-se esses quatro passos básicos:

1. Inicialmente, a aplicação cliente solicita um determinado documento ao Repositorio de Documentos. Neste momento, o repositório pode fazer uso de serviços estendidos, conforme mencionado. Como parâmetro de chamada, a aplicação pode informar o número da versão desejada. Por padrão, pode-se considerar a última versão publicada.
2. O repositório envia, então, uma chamada para o Documento, indicando qual a versão desejada. O Documento cria um manipulador de versão, passando como parâmetro o objeto Versao correspondente. A criação de manipuladores de versão é indiscriminada, uma vez que diversos usuários podem desejar ler um determinado documento simultaneamente.
3. A aplicação cliente recebe uma referência ao manipulador da versão solicitada. O Manipulador de Versao possui interface para acesso de leitura do conteúdo das versões do documento. A aplicação não tem acesso direto à instância de Versao.

4. A aplicação envia mensagens para o manipulador e, quando desejado, encerra a atividade de leitura, liberando o uso do manipulador.

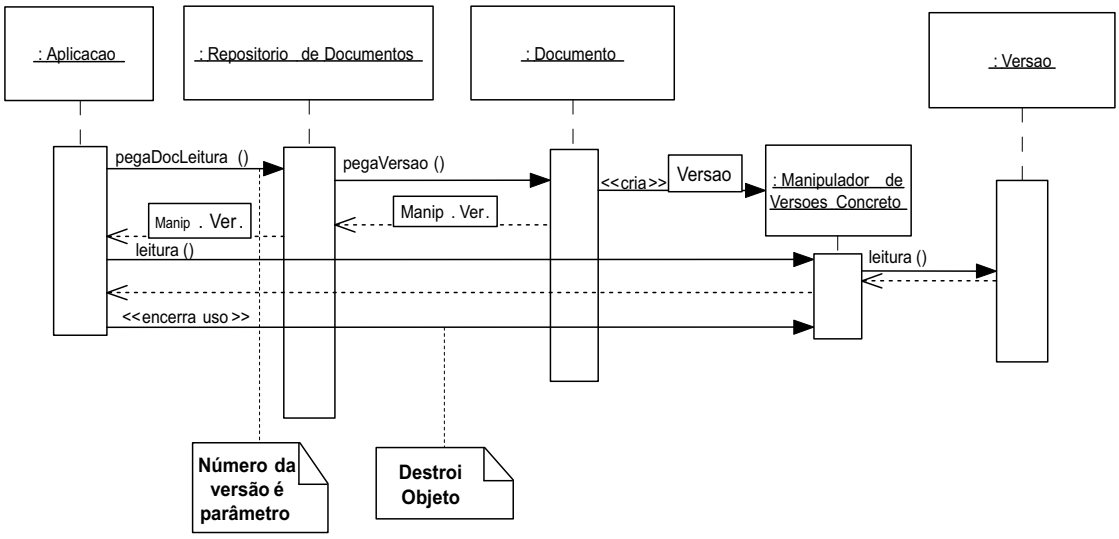

Figura 2: *Diagrama de seqüência para leitura de documento.*

Quando se deseja editar um determinado documento, o seguinte procedimento deve ser adotado:

1. De modo semelhante à leitura, a aplicação cliente faz solicitação de um determinado documento ao Respositorio de Documentos. Do mesmo modo, o repositório pode lançar mão de serviços estendidos.

2. O repositório envia a chamada ao Documento, que deve então proceder ao controle de edição. Se não houver nenhuma instância de Rascunho, então um novo Rasunho é criado. Se já houver instância no sistema, é enviada resposta negativa à solicitação. O mecanismo de controle de existência de rascunho deve ser executado de forma bloqueada, por tratar-se de região crítica para múltiplos fluxos de execução.

3. A aplicação cliente recebe uma referência para o manipulador de rascunho. Desse modo, também não recebe referência direta à instância de Rascunho.

4. A aplicação envia mensagens de modificação para o manipulador e, quando desejado, encerra a atividade de edição. O manipulador concreto instanciado pode estar associado a um protocolo de aprovação, liberando o rascunho editado para publicação apenas após a devida aprovação dos responsáveis.

5. O manipulador envia uma mensagem de versionamento para o Documento. O documento cria uma nova versão, com o conteúdo do rascunho. Em seguida, destrói

a instância de Rascunho, liberando o documento para edição por parte de outros colaboradores. Em alguns variações, pode-se manter uma lista de espera de editores, onde solicitações de edição negadas são armazenadas em uma fila e notificadas quando liberadas para uso.

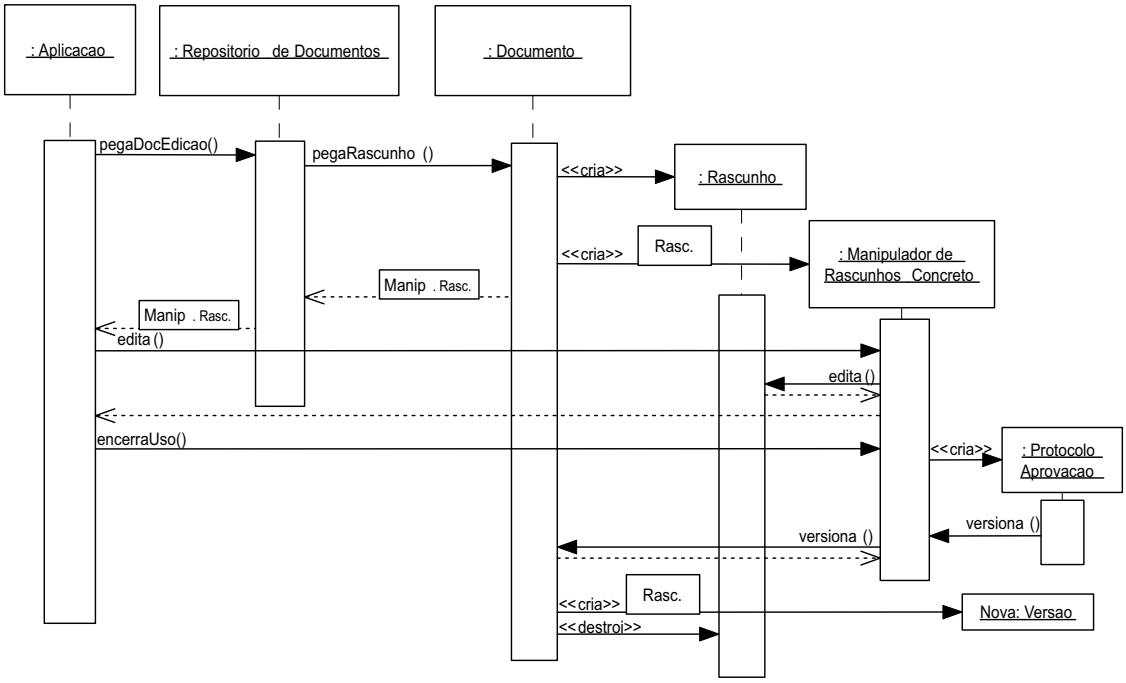

Figura 3: *Diagrama de seqüência para edição de documento.*

É importante lembrar que a elaboração de novos documentos segue os mesmos passos da edição, considerando apenas que um novo Documento é criado com a lista de versões vazia.

8 Contexto resultante

Após a implementação desse padrão, os usuários terão certeza de estarem lendo versões aprovadas dos documentos e de que apenas um revisor estará editando um determinado documento, num determinado instante. Além disso, as versões antigas dos documentos são mantidas inalteradas, garantindo assim um histórico confiável dos documentos.

Alguns cenários, como fábricas de software, têm como requisito a edição simultânea de documentos. O padrão *Gestor de Documentos* não é recomendado para esses casos, que devem utilizar mecanismos específicos de versionamento simultâneo.

9 Padrões relacionados

Possivelmente, há padrões que complementam o padrão *Gestor de Documentos*, tais como aqueles que implementam o *Controle de Acesso* e o *Protocolo de Aprovação*. Por exemplo, diversas implementações de controle de acesso se utilizam de perfis de usuários e de permissão hierárquica (semelhante ao que é definido nos sistemas de arquivos). O protocolo de aprovação define que etapas um documento deve percorrer antes de ser aprovado para publicação; sendo que, dependendo do documento em questão, diversos membros da diretoria ou presidência da corporação podem ser envolvidos na sua aprovação. Três tipos de protocolos de aprovação possíveis são: em *série*, no qual um aprovador só emite seu parecer após a análise de um determinado aprovador anterior; em *paralelo*, no qual o documento é submetido simultaneamente para análise de todos os aprovadores; e em *série/parelelo*, no qual os dois protocolos anteriores são combinados. Juntamente com o padrão *Gestor de Documentos*, os padrões de *Controle de Acesso* e de *Protocolo de Aprovação* definem uma linguagem de padrões para o domínio do conhecimento de gerenciamento eletrônico de documentos, conhecido sob a sigla GED.

Outro padrão relacionado é o *Version-Controlled Environment* [2], que provê mecanismos necessários para a reconstrução de versões antigas de software. Esse padrão, entretanto, tem objetivo mais restrito que o padrão *Gestor de Documentos*.

10 Usos conhecidos

Honeywell Commercial Avionics Systems [2] - implementa um mecanismo que garante um histórico confiável de todos os sistemas de controle dos aviões já produzidos pela empresa, mesmo para aqueles muito antigos (por serem caros, aviões têm vida útil muito longa). Se algum problema com um modelo é descoberto, ou ocorre algum acidente onde o software é apontado como possível causa, para investigar o problema, a Honeywell precisa reproduzir exatamente a versão do software que controlava o avião. Nesse caso, poder reconstruir qualquer versão de um sistema de controle é imprescindível.

Hummingbird DM [3] - em detalhe, este software implementa mecanismos bem mais complexos que os descritos neste documento. Entretanto, os conceitos de cópia de documento somente para leitura, cópia de documento para edição (aqui denominado rascunho), versionamento, controle de acesso e serviços de busca são compatíveis com o Gestor de Documentos.

NextPage NXT 3 [4] - esta plataforma de software comercial integra fontes de informações heterogêneas distribuídas, fornecendo ponto de acesso único e poderosas ferramentas de recuperação da informação. Os clientes deste sistema são navegadores de internet e podem solicitar diversos documentos para visualização simultânea. Entretanto, no módulo de gerenciamento do conteúdo nativo (o conteúdo distribuído em bancos de dados ou sistemas de arquivos não está sujeito a este controle), é possível estabelecer critérios de acesso e um mecanismo de cópia para edição. Isso faz com que apenas o

usuário que retirou o documento tenha direitos de inserir uma nova versão, garantindo que outros rascunhos para o mesmo documento não estejam ativos simultaneamente. Este produto não ofecere suporte ao versionamento, mas permite integração de módulos personalizados, desenvolvidos por terceiros, para este fim.

Referências

[1] Appleton, B. *Patterns and Software: Essential Concepts and Terminology*, In http://www.enteract.com/~bradapp, 2000.

[2] Cabrera, R., Appleton, B. and Berczuk, S. P. *Software Reconstruction: Patterns for Reproducing Software Builds*, In PLoP'99 Conference, 1999.

[3] Hummingbird Ltd. *Sítio*, In http://www.hummingbird.com.

[4] NextPage, Inc. *Sítio*, In http://www.nextpage.com.