

LockED: Uma Ferramenta para o Controle de Alterações no Desenvolvimento Distribuído de Artefatos de Software

Hugo Vidal Teixeira

Leonardo Gresta Paulino Murta

Cláudia Maria Lima Werner

{hvidal, murta, werner}@cos.ufrj.br

COPPE/UFRJ – Programa de Engenharia de Sistemas e Computação

Universidade Federal do Rio de Janeiro

Caixa Postal 68511 – CEP. 21945-970

Rio de Janeiro – Brasil

Abstract

This work describes an approach to support change control in distributed development of software projects. This approach differs from existing ones because it does not only handle source code, but also other artifacts produced within other stages of the software development process. The proposed approach uses Internet resources for communication and exchange of artifacts among the developers and the server.

1 Introdução

Os projetos de desenvolvimento de software contemporâneos têm progressivamente aumentado de tamanho e complexidade, sendo cada vez mais comum sua realização por equipes de médio porte (entre dez e vinte desenvolvedores) e grande porte (acima de vinte desenvolvedores). Com as facilidades de comunicação proporcionadas pela Internet, a necessidade de experiência em diversas áreas de conhecimento e a pressão por cronogramas mais restritos, alguns projetos são realizados por diversas equipes trabalhando concorrentemente. Ainda mais, as dificuldades de reunir os especialistas necessários em um mesmo local físico e a delegação do desenvolvimento de determinados componentes para outras empresas, são exemplos de fatores que podem exigir que as equipes participantes de um projeto estejam geograficamente distribuídas (MAIR, 1997).

A existência de mecanismos eficientes de comunicação, como a Internet, não é suficiente para solucionar os problemas de desenvolvimento concorrente de projetos de software. Ao contrário, o novo cenário traz novos problemas para o processo de desenvolvimento. Por exemplo, o trabalho concorrente de equipes geograficamente distribuídas dificulta o controle de alterações nos componentes de um projeto em desenvolvimento (PRESSMAN, 1997). Mesmo quando precedida de uma precisa definição das interfaces entre os componentes, a realização de um projeto pode exigir que diversos desenvolvedores alterem simultaneamente os mesmos componentes. Esta situação exige a adoção de políticas para manter a consistência entre os componentes da versão atual do projeto ou permitir que essa consistência seja posteriormente restituída.

Atualmente, a grande maioria das abordagens para solucionar esse tipo de problema é focada em código-fonte. Contudo, os sistemas mais complexos demandam um esforço adicional para a sua compreensão, tornando necessário o uso de artefatos que descrevem aspectos não representados pelo código-fonte, agregando informação em um nível de abstração mais elevado. Consideramos como artefatos de software (ou simplesmente artefato) qualquer produto gerado em qualquer etapa do processo de desenvolvimento de software. Desta forma, uma abordagem para solucionar este problema deve permitir o controle de alterações de todos os artefatos de software, como por exemplo: classes (com o seu código-fonte), diagramas de classe, diagramas de seqüência, diagramas de caso de uso, etc.

O objetivo deste artigo é descrever uma abordagem que visa solucionar o problema de controle de alterações de artefatos de software no desenvolvimento distribuído de projetos de software. Inicialmente, desenvolvemos uma abordagem focada no controle de alterações em código-fonte (MURTA et al., 2000), porém esta não se mostrou suficiente para o entendimento do projeto como um todo. Essa abordagem foi aprimorada para permitir o

controle de alterações de todos os artefatos de software produzidos em um projeto, sendo apoiada pela ferramenta LockED, discutida nesse artigo.

Este artigo está organizado em três seções, além desta introdução. A Seção 2 trata das abordagens existentes para solucionar o problema de controle de alteração de artefatos de software em ambientes distribuídos. A Seção 3 apresenta a abordagem proposta e, finalmente, a seção 4 apresenta as conclusões.

2 Ferramentas de Apoio ao Desenvolvimento Distribuído de Software

As ferramentas para controle de alteração de software disponíveis atualmente são focadas em artefatos de software armazenados em arquivos do sistema operacional (SO). Dentre elas, podemos citar:

Code Co-op (RELIABLE SOFTWARE, 2001): Permite o controle de versões de artefatos em desenvolvimento de forma transparente em relação a localização geográfica dos desenvolvedores, sem a necessidade de um servidor central, armazenando os artefatos de forma distribuída entre os desenvolvedores. Baseia o seu sincronismo em serviços de e-mail, rede local ou disquete. Sua licença de uso custa proporcionalmente ao número de desenvolvedores.

PVCS Version Manager (MERANT, 2001): Ajuda a organizar, proteger e gerenciar artefatos de software, utilizando interface gráfica baseada no Windows Explorer. Contém mecanismos para recuperação de erros e alterações perdidas. Sua versão completa permite o controle de versões em ambiente web.

Visual SourceSafe 6.0 (MICROSOFT CORPORATION, 2001): É integrado com todos os ambientes de programação da Microsoft e permite o gerenciamento da equipe juntamente com os artefatos desenvolvidos por ela. Esta ferramenta é altamente escalável e pode ser usada facilmente por equipes com pouca experiência técnica. Existem extensões que permitem o seu uso através da Internet, como por exemplo o SourceOffSite [4].

Tlib (BURTON SYSTEM SOFTWARE, 2001): Uma ferramenta rápida, que tem a estrutura de armazenamento aberta e permite a importação de projetos que já utilizam outras ferramentas de controle de alterações. Suporta a junção (merge) simultânea de diferentes versões do mesmo artefato de código.

GP-Version (QUALITY SOFTWARE COMPONENTS, 2001): Idealizado inicialmente para suportar o controle de alterações de código escrito em Delphi e C++ Builder, foi posteriormente estendido para suportar a linguagem Visual Basic. É baseado em extensões add-ins e foi escrito em Delphi.

Token (MURTA et al., 2000): Ferramenta precursora deste trabalho. As suas principais funcionalidades são: o cadastramento dos desenvolvedores participantes do projeto, a troca de informações entre estes desenvolvedores e o controle de alterações nos componentes do projeto. Token foi desenvolvida em ambiente Linux, utilizando a linguagem de script PHP3, o banco de dados MySQL e o servidor web Apache. A ferramenta é acessível via um navegador Internet, sendo portanto independente tanto da plataforma cliente quanto da plataforma servidora.

Essas ferramentas fazem o controle de modificações em textos, páginas html, código-fonte ou qualquer outro artefato que possa ser relacionado um para um com arquivos do SO. Entretanto, para que modelos possam ser armazenados e alterados sob o controle de uma ferramenta, esta não pode ser baseada em arquivos do SO, pois deve conseguir lidar internamente com o arquivo que contém os modelos a serem controlados.

3 A ferramenta LockED

Segundo os mais conhecidos princípios da engenharia de software, o desenvolvimento de sistemas é uma atividade muito mais complexa do que parece. A lógica envolvida por trás das funcionalidades da aplicação, juntamente com todas as suas classes, funções e interações torna o entendimento do problema uma tarefa muito complexa e trabalhosa. Da mesma forma, um dos erros mais cometidos pelos desenvolvedores de sistemas é o fato deles considerarem o

código do sistema o único produto do processo que merece atenção especial. Ao negligenciarem os modelos que compõem a arquitetura da aplicação, e que foram a base da análise e do projeto, eles estão perdendo valiosas informações sobre a complexa lógica que explica todo o comportamento do software. De fato, estão criando um problema que os futuros mantenedores do produto precisam enfrentar, que é a dificuldade do rastreamento do código e o seu correto entendimento.

Como vimos anteriormente, a ferramenta Token somente se preocupa com o código-fonte do sistema em desenvolvimento, não abrindo espaço para os demais produtos do processo de desenvolvimento no qual a equipe está inserida. Isto exige que o gerente da equipe procure por outras ferramentas, fazendo com que o sucesso do projeto se torne um fator de organização e comprometimento dos seus integrantes.

Analisando por outra perspectiva, a cooperação existente em todo o trabalho de equipe muitas vezes presencia grandes contratempos. A maior parte das dificuldades concentra-se na dispersão da informação e na maneira com a qual esta é compartilhada pelo grupo. Muitas pesquisas já contribuíram para o assunto, trazendo métodos, ferramentas e práticas de trabalho que facilitam a coordenação dos membros da equipe (CRNKOVIC, 1997; SCHOOLER, 1996). Entretanto, a complexidade envolvida neste trabalho existe e jamais deixará de ser considerada um desafio.

Observando através de um cenário mais específico, o desenvolvimento de sistemas dentro de um determinado domínio de aplicação apresenta particularidades sobre o trabalho em equipe. No contexto da engenharia de domínio (BRAGA et al., 1999), a modelagem do domínio e sua posterior evolução exigem um grande esforço tanto da equipe de desenvolvimento quanto dos especialistas na área. Os produtos resultantes desta análise (artefatos do domínio) precisam estar disponíveis para todos os desenvolvedores, apoiando as suas discussões e fundamentando as decisões sobre o desenvolvimento de um sistema em particular. De fato, quando a equipe está toda concentrada em um único local – cenário típico de pequenos projetos – as interações entre estes membros são razoavelmente fáceis de serem controladas. Por outro lado, quando a equipe se apresenta geograficamente distribuída, a concorrência sobre as alterações dos artefatos de software requer mais cautela.

Dentro deste contexto, foi desenvolvida a ferramenta LockED, a qual visa controlar e impor ordem sobre a criação e as alterações de artefatos do domínio, disponibilizando-os para alocação e desalocação. Seguindo o mesmo princípio da ferramenta Token, um usuário somente pode alterar um artefato quando este já estiver previamente alocado ao mesmo através da ferramenta LockED. A proposta é que haja uma base oficial de informações sobre o domínio em um servidor conectado a Internet / Intranet, a qual será acessada por toda a equipe. Cada integrante trabalhará em sua própria estação de trabalho utilizando a infraestrutura de reutilização provida pelo seu ambiente de modelagem, realizando atividades de modelagem, consultas e instanciando aplicações a partir destes artefatos.

Neste trabalho, integramos a ferramenta LockED a infra-estrutura de reutilização de software fornecida pelo projeto Odyssey (PROJETO ODYSSEY, 2001).

A infra-estrutura Odyssey (WERNER et al., 2000) visa apoiar uma reutilização efetiva durante todo o desenvolvimento de software, provendo métodos, ferramentas e procedimentos adequados para a especificação de modelos e aplicações do domínio. A infra-estrutura Odyssey também pode ser vista como um arcabouço onde modelos conceituais do domínio, modelos de projeto (arquiteturas de software específicas do domínio) e modelos implementacionais (frameworks) são especificados para um determinado domínio, disponibilizando componentes reutilizáveis neste domínio. Durante o processo de desenvolvimento de aplicações, estes componentes são refinados e adaptados ao contexto de um projeto específico.

3.1 Detalhamento da Arquitetura

Pelo fato da ferramenta LockED representar uma extensão da ferramenta Token, suas arquiteturas apresentam muitas características em comum, como a presença do servidor e dos clientes se comunicando via rede Internet ou intranet. A comunicação das estações de trabalho

Odyssey com a ferramenta LockED é feita utilizando-se arquivos exportados e importados por ambos. A Figura 1 ilustra a interação entre as ferramentas.

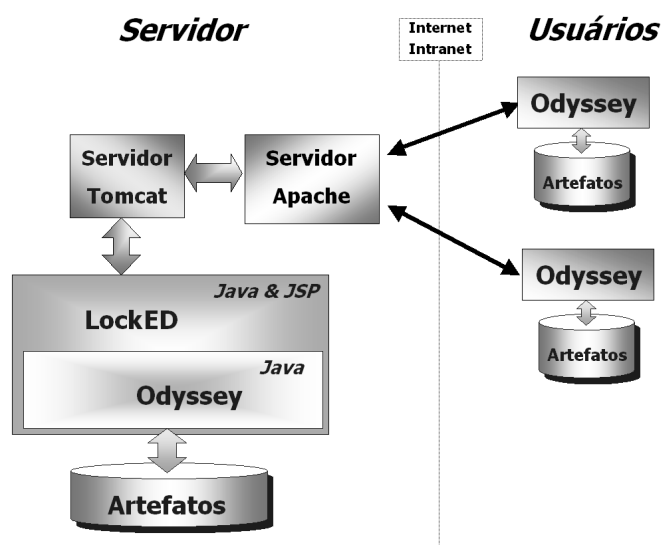


Figura 1: Arquitetura do LockED.

A realização do processo está baseada na alocação e desalocação dos artefatos do domínio. Para entendermos melhor como funciona este mecanismo, vamos acompanhar o processo desde o início. Suponha que uma equipe precise trabalhar com o domínio de telecomunicações e nada se encontra modelado. Primeiramente, alguém deverá criar o domínio no ambiente Odyssey, modelar o que desejar, e exportá-lo através do sistema LockED. A partir deste ponto, todos os outros membros da equipe poderão adquiri-lo no servidor (alocando o que desejarem) e importá-lo em suas estações de trabalho (Figura 2). Deste ponto em diante, cada usuário trabalhará em sua própria estação de trabalho, modificando o que estiver alocado em seu nome, além de poder criar novos artefatos. Quando o usuário desejar, ele poderá exportar o domínio (para um arquivo) fornecendo-o posteriormente ao sistema LockED, de forma que este atualize a sua base de dados.

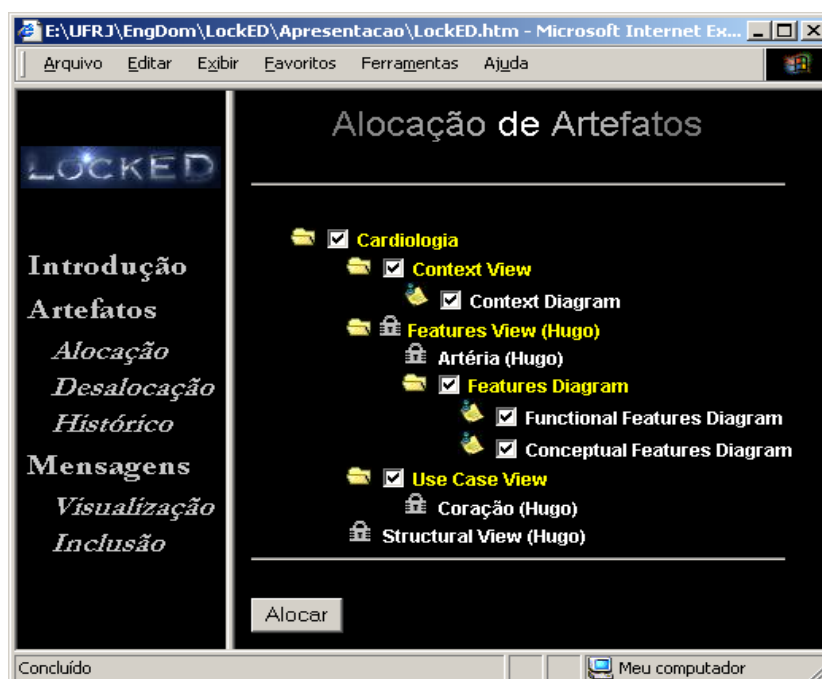


Figura 2: Alocação de artefatos.

Para que fosse possível a interação entre o Odyssey e o LockED, foi necessária a definição de algoritmos de alocação e desalocação de artefatos. A parte mais complexa desses algoritmos está no momento do *merge* das informações atuais com as novas, pois trabalhos já existentes não podem ser destruídos. A Figura 3 apresenta o algoritmo de importação de artefatos para a base do LockED (servidor), e a Figura 4 apresenta o algoritmo de importação de artefatos para a base do Odyssey (cliente).

```
(1) Faça o Upload da árvore
(2) Verifique se o domínio existe. Caso não exista, crie-o.
(3) Para cada nó X da árvore cujo dono é o usuário, faça:
  (3.1) busque o mesmo nó (antigo) na base do LockED
  (3.2) Se encontrar o nó X:
    (3.2.1) coloque o novo nó no lugar do antigo, preservando o seu pai
    e os seus descendentes
  (3.3) Senão (o nó foi criado)
    (3.3.1) busque o pai do nó X criado na árvore do LockED
    (3.3.2) insira o nó X criado no pai encontrado
    (3.2.2) Retire o nome do usuário do nó X (variável dono)
(4) Para cada nó Y presente na árvore da base LockED cujo dono ainda é o
usuário, faça:
  (4.1) Apague o nó Y
```

Figura 3: Algoritmo de importação de artefatos para a base do LockED.

```
(1) Faça o import da árvore pelo arquivo serializado
(2) Para cada nó X da base do Odyssey cujo dono é o usuário, faça:
  (2.1) Se o nó X foi criado (X.foiCriado == true)
    (2.1.1) Busque o pai do nó criado na base do LockED
    (2.1.2) Insira o nó criado no pai encontrado.
  (2.2) Senão, faça:
    (2.2.1) Busque o nó X na base do LockED
    (2.2.2) Copie X por cima do nó encontrado, preservando o seu pai e
    seus descendentes
(3) Substitua a base do Odyssey pela base serializada.
```

Figura 4: Algoritmo de importação de artefatos para a base do Odyssey.

4 Conclusão

Apresentamos neste artigo a ferramenta LockED, que foi implementada pelo grupo de reutilização de software da COPPE/UFRJ, com o objetivo de controlar as modificações realizadas em artefatos de software em um processo onde a equipe encontra-se geograficamente dispersa. A ferramenta foi desenvolvida utilizando-se a linguagem Java, combinada com a tecnologia JSP, sendo disponibilizada em um servidor web conectado à Internet/Intranet. Esta abordagem contribuiu com inúmeras vantagens tanto para a equipe como para a consistência das informações manipuladas, apresentando como benefícios diretos:

1. A consistência e a integridade da informação;
2. A unificação de todos os artefatos na base de dados oficial no servidor, o qual representa o ponto de contato entre todos os participantes da equipe;
3. A visibilidade sobre a alocação dos artefatos;
4. A facilidade de acesso à ferramenta pela Internet/Intranet;
5. A flexibilidade sobre a ordem de alocação e desalocação dos artefatos.

Este sistema foi integrado à infra-estrutura de reutilização de software Odyssey, a qual apresenta suporte para a modelagem de domínios e aplicações voltados para o desenvolvimento de software. A utilização do sistema LockED combinado com o Odyssey pode melhorar muito a maneira com a qual estamos acostumados a desenvolver software. Isto

porque não será mais necessário administrar o código-fonte dos sistemas em desenvolvimento com se fosse um produto do processo, visto que podemos gerá-lo a partir dos artefatos que descrevem a aplicação, os quais são modelados e armazenados em bases de dados do Odyssey.

Neste momento, o sistema LockED está em fase de experimentação pela equipe de desenvolvimento do Projeto Odyssey (cerca de 10 pessoas envolvidas, entre docentes, alunos de doutorado, mestrado e iniciação científica), não existindo ainda resultados estatísticos positivos ou negativos sobre sua utilização. Independente desses resultados, vislumbramos como uma possível evolução da abordagem aqui proposta, a inclusão de um contexto de trabalho cooperativo, onde um ambiente multi-usuário voltado para a modelagem dos domínios e aplicações de software fosse criado.

Referências Bibliográficas

- BRAGA, R. M. M., WERNER, C. M. L., 1999, "Odyssey-DE: Um Processo para Desenvolvimento de Componentes Reutilizáveis", *X CITS*.
- BURTON SYSTEM SOFTWARE, 2001, site na Internet em <http://www.burtonsys.com/>, acessado em 11/05/2001.
- CRNKOVIC, I., 1997, "Distributed Development Project using WWW", *19th International Multimedia Conference*, Opatija, Croácia, maio.
- MAIR, Q., 1997, "Technical Issues in the Design of a Virtual Software Corporation", *ECSCW'97 OOGP workshop*.
- MERANT, 2001, site na Internet em <http://www.merant.com/products/pvcs/vm/index.asp>, acessado em 11/05/2001.
- MICROSOFT CORPORATION, 2001, site na Internet em <http://msdn.microsoft.com/ssafe>, acessado em 11/05/2001.
- MURTA, L. G. P., BARROS, M., WERNER, C. M. L., 2000, "Token: Uma Ferramenta para o Controle de Alterações em Projetos de Software em Desenvolvimento", *XIV Simpósio Brasileiro de Engenharia de Software, Sessão de Ferramentas*, João Pessoa, outubro.
- PRESSMAN, R. S., 1997, "Software Engineering: A Practitioner's Approach", *McGraw-Hill*.
- PROJETO ODYSSEY, 2001, site na Internet em <http://www.cos.ufrj.br/~odyssey>, acessado em 11/05/2001.
- QUALITY SOFTWARE COMPONENTS, 2001, site na Internet em <http://www.qsc.co.uk/gpv/gpversion.htm>, acessado em 11/05/2001.
- RELIABLE SOFTWARE, 2001, site na Internet em http://www.relisoft.com/co_op/, acessado em 11/05/2001.
- SCHOOLER, E. M., 1996, "Conferencing and collaborative computing", *Multimedia Systems*, pp. 110-125.
- WERNER, C. M. L., BRAGA, R. M. M., MATTOSO, M., et al., 2000, "Infra-estrutura Odyssey: estágio atual", *XIV Simpósio Brasileiro de Engenharia de Software Sessão de Ferramentas*, João Pessoa, outubro.