

DiSEN-SCV: Uma estratégia para replicação de repositórios e alocação de artefatos

César Alberto da Silva, Elisa Hatsue Moriya Huzita

Departamento de Informática – Universidade Estadual de Maringá (UEM)
CEP 87020-900 – Maringá – PR – Brasil

{cesar_as,emhuzita}@din.uem.br

Abstract. *Several artifacts, such as: documents, models, source code, are generated through the software development and they are stored in repositories. Sometimes several versions of them are generated. So the artifacts version control in a Distributed Software Development Environment (DSDE) brings challenges concerned with artifact allocation. This paper presents a Version Manager for DSDE, considering a strategy for artifacts allocation as well as the repository replication. The difficulties found to maintain artifacts and their corresponding metadata synchronized are also discussed.*

Resumo. *No desenvolvimento de software vários são os artefatos gerados, podendo-se citar: documentos, modelos, código fonte; que podem, muitas vezes acabar sendo gerados em várias versões. O controle de versões de artefatos em um Ambiente de Desenvolvimento distribuído de software (ADDS) traz desafios no tocante à alocação de artefatos. Este artigo tem por objetivo apresentar um gerenciador de versões para um ADDS que trata de uma estratégia para alocação de artefatos e replicação de repositórios. São discutidas também as dificuldades encontradas para manter sincronizados os artefatos e seus metadados.*

1. Introdução

As organizações têm, cada vez mais, utilizado o desenvolvimento de software remoto como uma facilidade adicional, levando ao que é conhecido como Desenvolvimento Distribuído de Software (DDS) [Kiel, 2003] [Herbsleb e Moitra, 2001].

Desenvolvedores pertencentes a uma equipe de projeto podem estar dispersos geograficamente, apoiados por decisões estratégicas, motivados por redução de custos e aumento de produtividade. Assim, é desejável que as atividades sejam apoiadas por algum software (*groupware*), a fim de manter a consistência e uniformidade dos artefatos produzidos e desta forma minimizar o retrabalho.

A existência de mecanismos eficientes de comunicação, como a *Internet*, não é suficiente para solucionar os problemas de desenvolvimento cooperativo de projetos de software. Ao contrário, o novo cenário traz novos problemas para o processo de desenvolvimento. Por exemplo, o trabalho de equipes geograficamente distribuídas dificulta o controle de versões dos artefatos de um projeto em desenvolvimento [Pressman, 2001]. Mesmo quando precedida de uma definição precisa das interfaces entre os componentes, a realização de um projeto pode exigir que diversos desenvolvedores alterem, simultaneamente, os mesmos artefatos. Essa situação exige a

adoção de políticas adequadas para manter a consistência entre as versões do artefato do projeto ou permitir que essa consistência seja posteriormente restituída.

Um artefato é resultado de uma atividade e pode ser utilizado posteriormente como matéria-prima para aquela ou para outra atividade a fim de gerar novos artefatos. Dessa forma, uma atividade pode consumir artefatos (de entrada) e gerar novos artefatos (de saída). Artefatos são freqüentemente persistentes, armazenados em repositórios e possuem versões [Pascutti, 2002]. Como exemplo de artefato pode-se citar documentos, manuais, código fonte, modelos, etc..

O controle de versões de artefatos em um Ambiente de Desenvolvimento Distribuído de Software (ADDS) traz novos desafios, como por exemplo, a alocação de artefatos. Quando um desenvolvedor aloca um artefato, nenhum outro pode submeter uma nova versão deste e a sua desalocação é realizada quando o desenvolvedor, que está com o artefato alocado, submete uma nova versão ou desaloca-o por vontade própria. Dessa maneira, os desenvolvedores que trabalham na mesma atividade e não possuem a alocação do artefato, deverão aguardar, até que o artefato seja desalocado, para submeter uma nova versão. Isto leva a redução do paralelismo na execução da atividade.

Um outro problema desse cenário é em relação aos repositórios de armazenamento de artefatos. Com base nas características de um ADDS, podem existir diversos serviços replicados para garantir disponibilidade ao ambiente. Um desses serviços é o repositório de artefatos, portanto, ele deve ser gerenciado para que não haja inconsistência entre os repositórios, caso um repositório fique indisponível por um período de tempo.

Este artigo apresenta um gerenciador de versões de artefatos para um ADDS e descreve a solução implementada para melhorar o tempo de alocação dos artefatos entre os membros da equipe, bem como a estrutura utilizada para gerenciar os repositórios de artefatos distribuídos.

Dessa forma, o artigo está organizado em quatro seções, além desta introdução. A Seção 2 fornece uma visão geral do contexto onde o DiSEN-SCV (Sistema de Controle de Versão para ambiente DiSEN) está inserido. A Seção 3 apresenta o DiSEN-SCV e as características implementadas em relação à estratégia utilizada para alocação de artefatos e a forma utilizada para distribuição dos repositórios de artefatos. Por fim, a Seção 4 apresenta as conclusões.

2. Contexto

Este artigo está inserido no contexto do Projeto DiSEN (*Distributed Software Engineering Environment*), que visa prover um ambiente distribuído de desenvolvimento de software que considera as características de sistemas distribuídos e fornece suporte à cooperação entre espaços de trabalho distintos, além de apoiar um conjunto de ferramentas que auxiliam na tarefa do desenvolvimento de software.

O DiSEN se encontra em desenvolvimento pelo Grupo de Estudos de Engenharia de Software Distribuído, da Universidade Estadual de Maringá. Ele também incorpora a tecnologia de agentes segundo o padrão da FIPA¹. Segundo Pascutti (2002), a arquitetura do DiSEN foi projetada para utilizar, dentre outras, a MDSODI [Huzita, 2007], uma metodologia para desenvolvimento de software que leva em consideração

1 *Foundation for Intelligent Physical Agents*. <http://www.fipa.org>

algumas características identificadas em sistemas distribuídos, tais como concorrência, paralelismo, comunicação, sincronização e distribuição. A MDSODI também mantém as principais características do *Unified Process* [Jacobson, Booch, Rumbaugh, 1999], dirigida a *use-case*, centrada na arquitetura e desenvolvimento iterativo e incremental.

O objetivo do DiSEN é fornecer o suporte necessário para o desenvolvimento distribuído de software; a equipe poderá estar distribuída em locais geográficos distintos e trabalhar de forma cooperativa com uma metodologia para desenvolvimento distribuído de software.

A arquitetura do ambiente DiSEN foi proposta, inicialmente, por Pascutti (2002) e, posteriormente, Schiavoni (2007) definiu um *framework*, denominado FRADE (*Framework to Infrastructure of Distributed Software Development Environment*), para especificar a infra-estrutura que capacita o ambiente a gerenciar a comunicação entre os diversos participantes.

A arquitetura definida por Pascutti é apresentada na Figura 1.

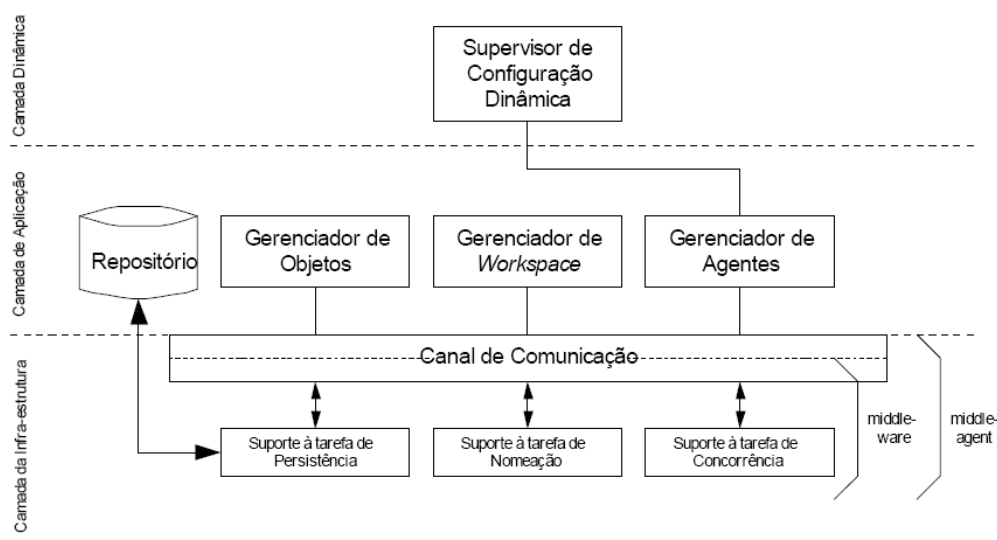


Figura 1: Arquitetura DiSEN [Pascutti, 2002]

A arquitetura do DiSEN é constituída por gerenciadores (objetos, *workspace* e agentes), suportes (persistência, nomeação e concorrência) e um supervisor de configuração dinâmica para prover infra-estrutura necessária na realização do desenvolvimento distribuído. O gerenciador de objetos engloba os gerenciadores: projeto, processo, artefato, entre outros.

3. Gerenciador de Artefatos - DiSEN-SCV

O Gerenciador de Artefatos é um componente do Ambiente Distribuído de Desenvolvimento de Software – DiSEN, com a finalidade de gerenciar as versões dos artefatos que são gerados durante a execução das atividades do projeto.

Além de gerenciar as versões dos artefatos, o Gerenciador de Artefatos - DiSEN-SCV é responsável por controlar o acesso aos artefatos e, também, resolver os conflitos, dentre as versões, que possam ocorrer durante o desenvolvimento do projeto.

Esse artigo explora a estratégia de alocação de artefatos e o método utilizado para replicação dos repositórios de artefatos.

Estratégia de alocação de artefatos

A estratégia de alocação de artefatos evita a ocorrência de conflitos entre as versões de artefatos. Borghoff e Schlichter (2000) apresentam duas formas para resolução de conflitos: pessimista e a otimista. Na forma pessimista, assume-se que as ações serão incoerentes e atingirão um estado inválido e indesejado. Neste caso, a ação de um membro deve bloquear as ações dos demais sobre o mesmo artefato. Na forma otimista, assume-se que as ações serão coerentes, mesmo ocorrendo de forma isolada, e o resultado final será um estado válido. Neste caso, mais de um membro realiza ações sobre um mesmo artefato.

O membro da equipe, que desejar trabalhar isoladamente, deverá solicitar o bloqueio sobre um determinado artefato. No entanto, o uso de bloqueios impossibilita o paralelismo de atividades que concorrem pelo artefato bloqueado. Em geral, os membros adotam a estratégia de geração de cópias do artefato, que são trabalhadas em paralelo e, posteriormente, consolidadas em uma única versão. Assim, por um lado o bloqueio evita a ocorrência de conflitos, mas reduz o paralelismo no trabalho, enquanto que o uso de cópias apenas posterga a detecção dos conflitos, mas possibilita um maior paralelismo de atividades.

Portanto, apenas adoção da técnica pessimista ou otimista na forma originalmente propostas agravam os problemas relacionados, respectivamente, ao paralelismo na execução das atividades e à coerência de alterações realizadas sobre os artefatos. Tais problemas se agravam ainda mais quando se considera uma equipe dispersa geograficamente em um ADDS. Assim, a estratégia adotada no DiSEN-SCV é a alocação de artefato por um período de tempo determinado. Desta forma, cada desenvolvedor autorizado pode alocar um artefato para realizar as alterações necessárias, iniciando a contagem do tempo correspondente. Tão logo este tempo expire ou quando a submissão, no repositório, de uma nova versão do artefato é realizada, ocorre a desalocação do artefato para este desenvolvedor. A partir deste momento, este artefato está disponível para outro desenvolvedor que esteja interessado em realizar alterações.

Caso um desenvolvedor solicite a alocação de um artefato, que já esteja alocado para um outro desenvolvedor, a solicitação de alocação será armazenada em uma fila de espera. Quando o tempo de alocação expirar ou o desenvolvedor, que está com o artefato alocado, enviar uma nova versão do artefato para o repositório, o artefato será alocado para o primeiro da fila de espera, caso não existir nenhuma solicitação na fila, o artefato será liberado (desalocado).

A Tabela 1 apresenta as características que foram consideradas na avaliação e quanto à utilização do Subversion e o SGBD como repositório de artefatos. Apesar do SGBD, de um modo geral, não disponibilizar todas as características, foram analisadas as possibilidades e as dificuldades associadas à sua implementação. Com base nas dificuldades encontradas para utilizar o Subversion, dentro do ambiente DiSEN, decidiu-se por utilizar um SGBD como repositório de artefatos.

Tabela 1: Comparação do uso do Subversion e SGBD como repositório de artefatos

	SGBD	SubVersion
Versionamento de diretórios	Sim	Sim
Alocação de artefatos por um período de tempo	Sim	Não
<i>Branching</i>	Sim	Sim
Integração ao ambiente DiSEN	Fácil	Difícil
Adição de novos estados para os artefatos (ex. Beta)	Sim	Não
Controle de concorrência	Sim	Sim
Sincronização com os metadados (atividades e artefatos)	Fácil	Difícil

A utilização de um SGBD como repositório de artefatos, possibilitou algumas vantagens:

- o ambiente DiSEN possui um mecanismo de persistência que está consolidado e integrado ao *middleware* do ambiente. Portanto, facilitou o uso do SGBD como repositório de artefatos.
- o uso do SGBD garante o controle de concorrência das transações.
- o armazenamento dos artefatos no mesmo repositório de seus metadados, facilitou o sincronismo entre os mesmos.
- com a utilização do *middleware* Sequoia, permitiu a existência de repositórios distribuídos.

Como desvantagens da utilização do SGBD, em relação a utilização do Subversion, pode-se citar:

- necessidade de implementar o controle de versões de artefatos, mantendo o histórico de atualizações e versionamento de diretórios, ao invés de utilizar o controle do Subversion.
- necessidade de implementar o algoritmo de *merge* para artefatos no formato texto, ao invés de utilizar o algoritmo do Subversion.

Replicação dos repositórios de artefatos

Na primeira implementação do DiSEN-SCV foi utilizado o Subversion como repositório de artefatos e o servidor de aplicações Apache² como meio de comunicação. O Subversion é um Sistema de Controle de Versões (SCV) *open-source* e o armazenamento dos artefatos é centralizado.

2 Servidor de aplicações Apache. <http://httpd.apache.org>

O Subversion foi escolhido, inicialmente, por fornecer: versionamento de diretórios, histórico das versões, atomicidade no processo de *commit*, permissão para realizar *branching e merge*.

A arquitetura do Subversion pode ser visualizada na Figura 2.

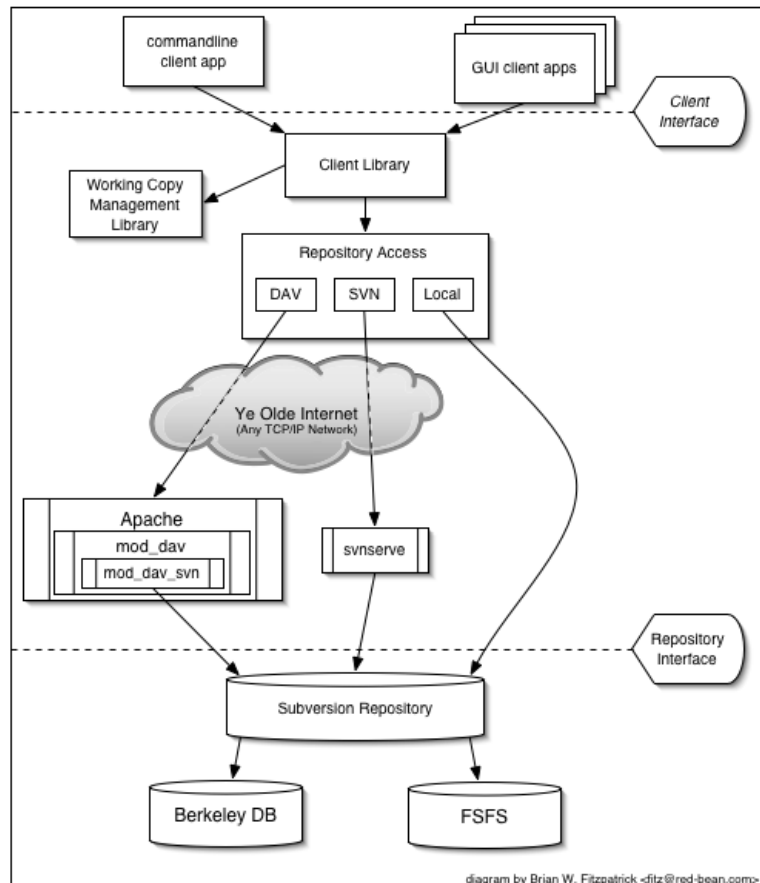


Figura 2: Arquitetura do Subversion [Collins-Sussman, Fitzpatrick e Pilato, 2004]

A arquitetura do Subversion disponibiliza uma interface para acesso ao repositório e uma biblioteca para as aplicações “cliente” acessarem o repositório. A biblioteca fornece três formas para acesso ao repositório: utilizando o servidor de aplicações Apache, o servidor *svnserver* e acesso local.

Além da utilização do Subversion como repositório de artefatos, foi utilizado um Sistema de Gerenciamento de Banco de Dados (SGBD) para armazenar os metadados dos artefatos.

O DiSEN possui um módulo para gerenciamento de projetos, que planeja e controla a execução dos projetos. Esse módulo controla: as atividades que deverão ser realizadas, o prazo, as dependências, qual o participante do projeto que irá executar uma atividade, quais os artefatos gerados, entre outras. Mais detalhes podem ser encontrados em [Enami, 2006] [Schiavoni, 2007].

Além de manter os artefatos associados a uma atividade de projeto, o DiSEN-SCV deve armazenar os metadados dos artefatos. Os metadados são armazenados em um SGBD e, devido à utilização do Subversion, os artefatos são armazenados no repositório do Subversion.

Diante do contexto de utilização do DiSEN-SCV, em um ADDS, essa situação pode gerar problemas de sincronização entre os repositórios, de metadados e de artefatos. Em um ADDS, os repositórios de metadados e de artefatos podem estar em servidores diferentes, portanto, deve existir um mecanismo para garantir o sincronismo desses repositórios (Figura 3).

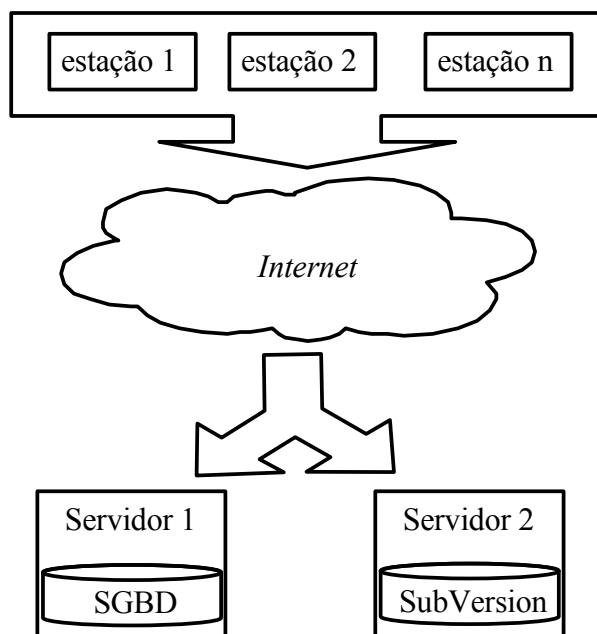


Figura 3: Um exemplo de localização de repositórios em ADDS

Devido ao problema de sincronismo entre os repositórios, a sua utilização ficou inviabilizada. Assim, foi implementada uma segunda versão para o DiSEN-SCV, utilizando outro repositório de artefatos.

Com a necessidade de manter sincronizados os artefatos com seus metadados, foi decidido utilizar um único repositório para o armazenamento de ambos. Como os metadados do projeto são armazenados em um SGBD e é necessário, também, armazenar os metadados dos artefatos, foi utilizado, também, o SGBD como repositório de artefatos.

Além do SGBD armazenar os artefatos e seus metadados, ele garante que as transações serão executadas de forma segura, seguindo as propriedades ACID (Atomicidade, Consistência, Isolamento, Durabilidade).

No ambiente DiSEN é utilizado o Sequoia para gerenciamento dos SGBDs. Sequoia é um middleware que oferece *clustering*, balanceamento de carga e serviços de gerenciamento de falhas. Os *databases* são distribuídos e replicados entre vários *host*, também fornece suporte para manutenção e recuperação on-line [Continuent, 2007].

Um exemplo de aplicação do *middleware* Sequoia é apresentada na Figura 4.

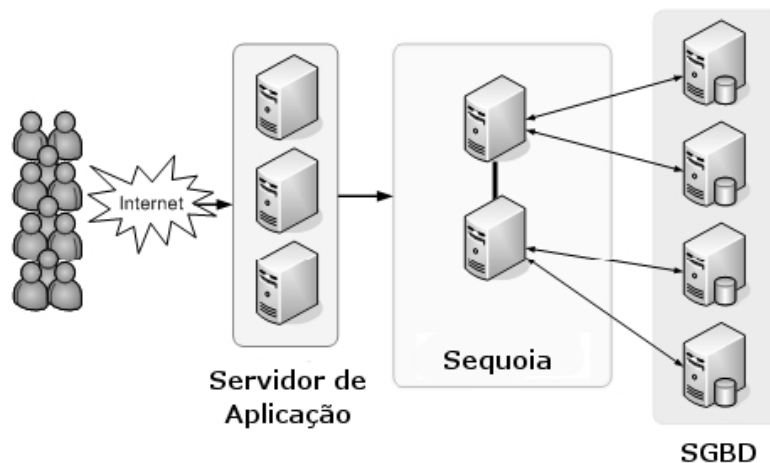


Figura 4: Sequoia [Continuent, 2007]

Portanto, a utilização do SGBD como repositório de artefatos dentro do ambiente DiSEN e a utilização do *middleware* Sequoia, permitiriam o armazenamento dos artefatos em repositórios distribuídos e, ainda, os problemas que ocorriam durante a sincronização dos artefatos com seus metadados deixaram de existir.

Para avaliação e comprovação da replicação dos repositórios de artefatos, foram instalados em dois servidores o SGBD PostgreSQL e em um outro servidor foi instalado o DiSEN-SERVER e o *middleware* Sequoia. A configuração dos SGBDs no Sequoia é ilustrada pela Figura 5.

```

<DatabaseBackend name="disen1" driver="org.postgresql.Driver"
url="jdbc:postgresql://200.201.30.204:5432/disen"
connectionTestStatement="select 1">
  <ConnectionManager vLogin="user" rLogin="disen" rPassword="disen">
    <VariablePoolConnectionManager initPoolSize="40"/>
  </ConnectionManager>
</DatabaseBackend>

<DatabaseBackend name="disen2" driver="org.postgresql.Driver"
url="jdbc:postgresql://200.201.30.193:5432/disen"
connectionTestStatement="select 1">
  <ConnectionManager vLogin="user" rLogin="disen" rPassword="disen">
    <VariablePoolConnectionManager initPoolSize="40"/>
  </ConnectionManager>
</DatabaseBackend>
    
```

Figura 5: Configuração do *middleware* Sequoia

Após a instalação dos servidores e configuração do Sequoia, foram criados e enviados quatro artefatos para o repositório. Para verificar se os artefatos foram armazenados em ambos os repositórios, foi realizada, individualmente, uma consulta em cada repositório (Figuras 6a e 6b) utilizando a ferramenta Squirrel SQL³.

3 Universal SQL Client. <http://www.squirrelsql.org>

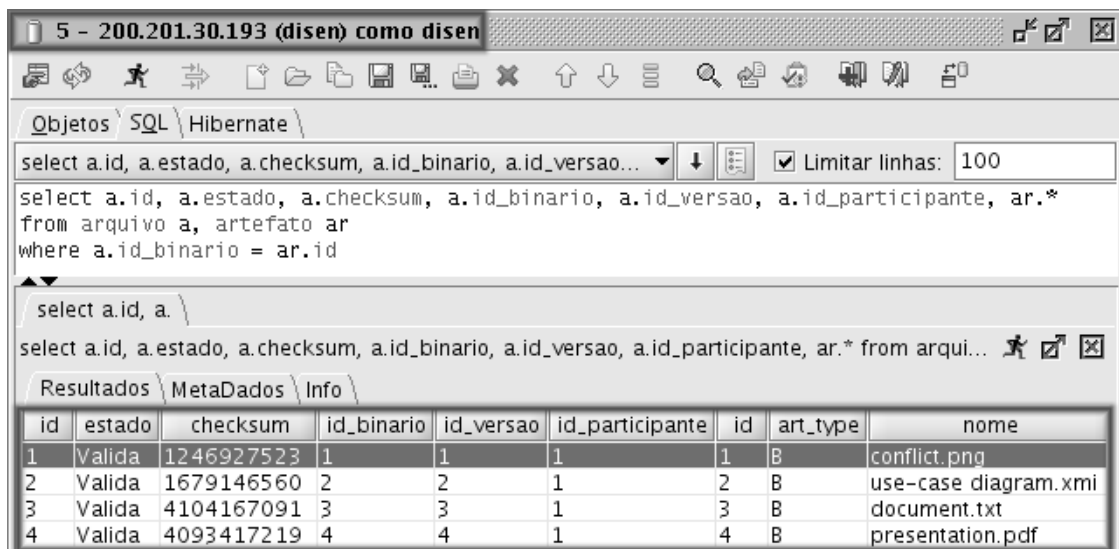


Figura 6a: Consulta aos artefatos do servidor 1

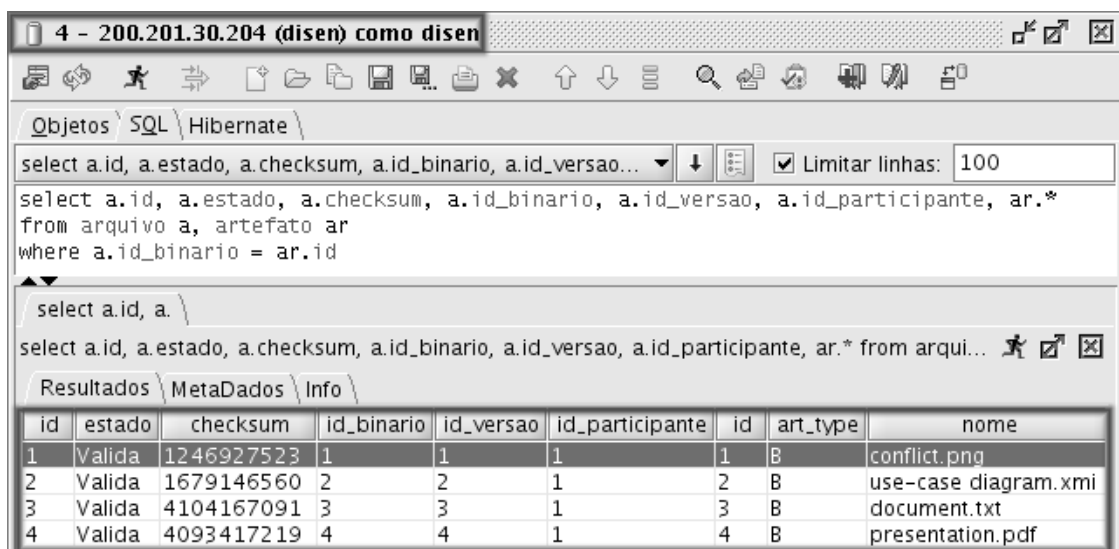


Figura 6b: Consulta aos artefatos do servidor 2

As Figuras 6a e 6b ilustram a consulta realizada, respectivamente, nos servidores 200.201.30.193 e 200.201.30.204. Também pode ser observado nessas figuras que os artefatos armazenados, em ambos os servidores, são idênticos.

4. Conclusões

Este artigo apresentou o DiSEN-SCV e explorou as características implementadas em relação à estratégia de alocação de artefatos e a forma utilizada para armazenar os artefatos em repositórios distribuídos.

A alocação de artefatos utilizada visa evitar a ocorrência de conflitos entre as versões dos artefatos. Entretanto, a alocação de artefatos na forma pessimista, proposta por Borghoff e Schlichter (2000), diminui o paralelismo na execução das atividades. Portanto, o DiSEN-SCV utiliza a alocação de artefatos por um período de tempo determinado. O tempo de alocação é determinado pelo gerente de projeto, de acordo

com o grau de dificuldade para a execução da atividade e o conhecimento do desenvolvedor.

O controle de alocação de artefatos por um período de tempo, evita que a desalocação do artefato seja realizada somente quando o desenvolvedor desejar, permitindo melhor equilíbrio no tempo de alocação do artefato entre os membros de uma equipe.

Os repositórios de artefatos distribuídos se concretizou com a utilização de um SGBD para armazená-los e o uso do *middleware* Sequoia para o gerenciamento, balanceamento de carga, replicação e distribuição entre os repositórios de artefatos. Dessa maneira, pode-se garantir a sincronização entre os artefatos e seus metadados, em um ADDS.

Referências Bibliográficas

- Borghoff, U. M., Schlichter, J. H. (2000), Computer-Supported Cooperative Work: Introduction to Distributed Applications. *Springer*, USA.
- Collins-Sussman, B., Fitzpatrick, B. W., Pilato, C. M. (2004), Version Control with Subversion. Editora O'REILLY. 1ª edição. 319 pág., ISBN 10: 0-596-00448-6 | ISBN 13:9780596004484. Junho.
- Continuent (2007), Middleware Sequoia. Disponível em: <http://sequoia.continuent.org>. Acesso em 17/05/2008.
- Enami, L. N. M. (2006) “Um Modelo de Gerenciamento de Projetos para um Ambiente de Desenvolvimento Distribuído de Software”. Dissertação de Mestrado – Programa de Pós-Graduação em Ciência da Computação, Universidade Estadual de Maringá, Maringá.
- Herbsleb, J. D., Moitra, D. (2001) “Guest Editors' Introduction: Global Software Development”, IEEE Software, vol. 18, no. 2, pp. 16-20, March/April.
- Huzita, E. H. M. et. al. (2007) “Um Ambiente de Desenvolvimento Distribuído de Software – DiSEN”. In: SBES - I Workshop de Desenvolvimento Distribuído de Software, João Pessoa.
- Jacobson, I., Booch, G., Rumbaugh, J. (1999) “The Unified Software Development Process”. [S.l]: Addison-Wesley.
- Kiel, L. (2003) “Experiences in Distributed Development: A Case Study, In: Workshop on Global Software Development at ICSE 2003”, Oregon, EUA.
- Pascutti, M. C. D. (2002) “Uma proposta de arquitetura de um ambiente de desenvolvimento de software distribuído baseado em agentes”. Dissertação (Mestrado) - Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- Pressman, R. S. (2001), Software Engineering: A Practitioner's Approach, McGraw-Hill International Editions, 5th ed.
- Schiavoni, F. L. (2007) “FRADE – Framework para infra-estrutura de um Ambiente Distribuído de Desenvolvimento de Software”. Dissertação de Mestrado – Programa de Pós-Graduação em Ciência da Computação, Universidade Estadual de Maringá, Maringá.