

Reuso e versionamento de conteúdos didáticos digitais em repositório interinstitucional compartilhado: Modelagem e validação por Redes de Petri Coloridas.

Corneli Gomes F. Júnior^{1,2}, Danielo Gonçalves Gomes¹,
José Marques Soares¹, Giovanni Cordeiro Barroso¹, Regia Talina Silva Araujo²

¹Departamento de Teleinformática – Universidade Federal do Ceará (UFC)
Campus do PICI S/Nº, Bloco 725 – Fortaleza – CE – Brasil

²Instituto Federal de Educação Ciência e Tecnologia do Ceará (IFCE)

{cjunior, regia}@ifce.edu.br, {danielo, marques}@ufc.br, gcb@fisica.ufc.br

Abstract. *The free availability of e-learning objects (ELOs) is clearly growing on the Web. A good ethical conduct requires that modified versions be made available along with the original, keeping the proper credit of authorship. It is not clear, however, repositories that provide appropriate tools for version control. In this work we present using Colored Petri Nets a model of content management used by EPCT Virtual based on integration of Database Management Systems (DBMS) and Version Control Systems (SVN).*

Resumo. *A livre disponibilização de conteúdos didáticos digitais (CDDs) vem crescendo de maneira expressiva na Web. A boa conduta ética requer que versões modificadas sejam disponibilizadas juntamente com a original, guardando os créditos de autoria. Não se percebe, entretanto, repositórios que forneçam instrumentos apropriados para o controle de versões. Neste trabalho apresentamos através de Redes de Petri Coloridas o modelo de gestão e versionamento de conteúdos proposto para o Portal EPCT Virtual. O modelo proposto se baseia na integração entre Sistemas Gerenciadores de Bancos de Dados (SGBD) e Sistemas de Controle de Versão (SVN).*

1. Introdução

A idéia da criação e disponibilização de conteúdos de apoio ao processo de ensino e aprendizagem a partir da combinação de componentes reutilizáveis armazenados em bases de dados vem sendo discutida há tempo [Muzio et al. 2002]. Nos últimos anos, o Ministério da Educação (MEC) têm financiado tanto o desenvolvimento de ferramentas como o trabalho de catalogação e disponibilização de conteúdos através de editais específicos. Dentre os projetos financiados pelo MEC podemos destacar: o Rived [RIVED 2010]; Portal do Professor [SEED-MEC 2010]; Fabrica Virtual [SEED-MEC 2010] e o Portal EPCT Virtual [EPCT Virtual 2010].

Com o objetivo de evitar trabalhos redundantes, muitas vezes desenvolvidos dentro das mesmas instituições de ensino, conteúdos didáticos disponibilizados podem ser reaproveitados e associados a outros elementos para compor um novo conteúdo, permitindo aos autores maior produtividade e economia de tempo [Júnior et al. 2009].

O versionamento de conteúdo é uma atividade de grande utilidade em diversas áreas do conhecimento. É útil, por exemplo, para acompanhar a evolução de uma pesquisa, o desenvolvimento de um código fonte ou as modificações de um conteúdo didático digital (CDD). Versionamento, no âmbito dos CDDs, deve ser entendido como a operação de criar novas versões de um material didático a partir de um conteúdo pré-existente. Ferramentas que provêem essas funcionalidades são conhecidas por sistemas de controle de versão (SCV). Dentre os SCV mais populares podemos citar o Subversion [SUBVERSION 2011] e o [CVS 2011].

O EPCT Virtual é um espaço digital destinado a contribuir com a formação, informação, instrumentalização e comunicação de professores, pesquisadores, alunos e interessados em tecnologia educacional aplicada à Educação Profissional Científica e Tecnológica (EPCT) [Júnior et al. 2009].

Um dos requisitos do EPCT Virtual é que os conteúdos disponibilizados, quando modificados, sejam versionados e retornados ao repositório, guardando os devidos créditos de autoria. Deve ser possível a recuperação eficiente de qualquer versão de um conteúdo postado, de acordo com a necessidade do usuário.

As redes de Petri [Murata 1989] e suas extensões constituem uma classe de modelos conceituais utilizada na modelagem de diversos tipos de sistemas computacionais paralelos. Por exemplo, têm-se usado redes de Petri para modelar protocolos de comunicação, processos de manufatura, arquiteturas de computadores etc. Tais modelos possibilitam a representação de sistemas paralelos, os quais podem então ser simulados, por exemplo, nos níveis funcional e lógico, incluindo ou não aspectos de temporização [BATISTA et al. 2010].

Tendo em vista todas as vantagens das abordagens baseadas em Redes de Petri, principalmente relacionadas ao poder de análise, e a especificação de um repositório que permite o reuso e o acesso eficiente a CDDs [Júnior et al. 2011], neste artigo é proposto e desenvolvido um modelo de armazenamento eficiente de conteúdos e metadados versionados, baseados na integração de Sistemas Gerenciadores de Base de Dados (SGBD) e de Sistemas de Controle de Versão (SCV), que deverá ser utilizado na nova versão do EPCT Virtual.

Este documento está organizado da seguinte maneira: após a introdução realizada nesta seção, os trabalhos relacionados ao reuso e versionamento de conteúdos são apresentados na seção 2. Na seção 3 são apresentados os resultados de desempenho dos sistemas de armazenamento MySQL, PostgreSQL e Subversion, utilizados na composição do modelo. Na seção 4 são apresentadas algumas questões relacionadas a Redes de Petri Coloridas (RPC); O modelo de armazenamento versionado de conteúdos e metadados será apresentado na seção 5. Os resultados e conclusões serão expostos na seção 6.

2. Trabalhos Relacionados

A produção de conteúdos de qualidade, em especial a produção de recursos didáticos para apoio ao processo de ensino e aprendizagem, é uma atividade de elevado custo. Dessa forma, recomenda-se que estes valorizem o reaproveitamento e a interoperabilidade entre diferentes plataformas [GIRARDI 2004].

Allen e Mugisa [ALLEN and MUGISA 2010] defendem a programação

orientação a objetos (POO) como meio para prover recursos digitais reutilizáveis e interoperáveis. Com esse paradigma de programação, é possível obter o benefício da reutilização de códigos, agregação de módulos prontos e o conseqüentemente o incremento na produtividade.

Algumas equipes do Fábrica Virtual utilizam um arquivo externo ao CDD para representar sua arquitetura de informação, que é constituída de metadados. Os metadados, de acordo com Girardi (2004), são informações que definem características como: conteúdo, objetivos, autor(es), idioma, data, versão, etc. Os metadados podem ser descritos em XML (Extensible Markup Language), que é uma linguagem de marcação extensível, permitindo definir e compor elementos específicos de um domínio. Por ser extensível, todo documento XML deve obedecer a regras pré-estabelecidas.

A importância de metadados para a descrição de CDDs está na possibilidade de localização, armazenamento e (re-)utilização destes materiais. Os metadados também facilitam o compartilhamento e o intercâmbio dos mesmos, para isso, existem padrões que facilitam a troca de dados entre os conteúdos que utilizam o mesmo padrão, permitindo o compartilhamento de dados.

Muzio [Muzio et al. 2002] descreve suas experiências práticas na criação, armazenamento, uso e reuso de conteúdos digitais no CEDAR¹. Cada conteúdo produzido recebe um conjunto de metadados, definidos através de palavras-chave, e que não seguem um padrão único. Essa abordagem permite o desenvolvimento de conteúdos reutilizáveis, que darão origem a novos conteúdos. CEDAR reconhece a necessidade da criação de um padrão para descrever o material produzido, estando consciente da necessidade de uma nova catalogação dos objetos posteriormente.

O LOM (Learning Object Metadata) é um padrão que segue os propósitos genéricos de metadados, possibilitando que conteúdos desenvolvidos, organizados e armazenados nesse padrão possam ser recuperados [SCHEER and GAMA 2004]. Cabe salientar que o LOM é utilizado em padrões mais modernos e abrangentes, como SCORM [BOHL et al. 2002], além de recomendado pelo RIVED. Apesar disso, nem todas as equipes seguem ou adotam um padrão de metadados.

Os conteúdos postados no EPCT Virtual podem ser reaproveitados, modificados ou adaptados a propósitos particulares, sem necessariamente precisar passar por um novo processo de desenvolvimento integral. Nesse sentido, o versionamento configura-se como uma técnica adequada ao desenvolvimento de revisões diversas de um CDD. Cada revisão pode ser uma cópia da anterior, adaptada a públicos específicos ou desenvolvida com tecnologia diferente, permitindo a sistematização do mecanismo de compartilhamento e devolução ao repositório das modificações propostas, contribuindo com a democratização do acesso às informações [Júnior et al. 2011].

Nos trabalhos relacionados precedentemente, nota-se que o reuso é mais fortemente centrado na representação do CDD do que em aspectos relacionados à sua composição em termos de estrutura de dados. Neste trabalho, privilegia-se o tratamento dado ao conteúdo digital (vídeos, apresentações em forma de slides eletrônicos ou quaisquer outros tipos de documentos digitais) a fim de evitar a redundância no armazenamento de novas versões, buscando, entretanto, soluções que não comprometam o desempenho

¹Centre for Economic Development and Applied Research.

de acesso aos conteúdos disponibilizados.

O modelo de armazenamento versionado de CDDs proposto neste trabalho é baseado na integração entre os tradicionais SGBDs e SVCs, podendo ser estendido a qualquer sistema cujo objetivo final seja o armazenamento versionado de conteúdos binários. A análise de desempenho realizada por Júnior [Júnior et al. 2011] é base para a construção do nosso modelo, e será apresentada na seção seguinte.

3. Análise de desempenho dos sistemas de armazenamento envolvidos na modelagem

Visando especificar um repositório que permita não só o reuso, mas também o acesso eficiente a CDDs Júnior (2011) avalia os aspectos positivos e negativos de técnicas distintas para versionamento na Web, apresentando uma análise comparativa de desempenho entre os repositórios PostgreSQL, MySQL e SVN.

O SVN oferece dois subsistemas *backend* para armazenamento de deltas². O primeiro utiliza um ambiente de banco de dados chamado Berkeley DB (BDB). Repositórios baseados nesse ambiente também são conhecidos como *BDB-backed*. O segundo usa o próprio sistema de arquivos do sistema operacional, em um mecanismo conhecido como FSFS. O *backend* de armazenamento utilizado pelo repositório SVN influencia diretamente em seu desempenho [COLLINS-SUSSMAN et al. 2007] e, devido a isso, foram realizados testes com as duas configurações.

A análise de desempenho foi realizada com o uso de ferramentas de testes desenvolvidas especificamente para reproduzir as funções de inserção (*upload*) e resgate (*download*) de CDDs via HTTP. Os requisitos de construção do repositório considerados são: o acesso eficiente para consulta via Web incluindo situações de concorrência, o reuso dos arquivos disponibilizados e, conseqüentemente, a geração de novas versões e, por último, a disponibilidade permanente de acesso às versões anteriores.

Após as experimentações e testes realizados, foi constatado que os melhores tempos para inserção e resgate de conteúdos, bem como o menor consumo de memória e de CPU foram obtidos com o uso do MySQL, seguido pelos sistemas PostgreSQL, SVN *FSFS-backed* e SVN *BDB-backed*, nesta ordem. Observando-se a métrica ocupação de espaço em disco, devido ao modelo de armazenamento que considera apenas as diferenças entre os conteúdos em diferentes versões (deltas), os repositórios SVN *FSFS-backed* e SVN *BDB-backed* apresentaram os melhores resultados. Nota-se, entretanto, que o SVN *BDB-backed* não apresenta desempenho que justifique sua substituição pelo SVN *FSFS-backed*.

Tendo em vista a construção de um repositório Web que ofereça um acesso eficiente aos conteúdos nele armazenados, os resultados apresentados, por um lado, remetem a uma solução cujos objetos mais frequentemente utilizados sejam mantidos em SGBDs tradicionais. Nota-se, para este fim, vantagem em relação ao tempo de resposta no uso do MySQL. Todavia, considera-se que o PostgreSQL, embora tenha apresentado desvantagem relativa aos tempos de resposta do MySQL, pode ser utilizado em aplicações com fortes requisitos de segurança e robustez, visto que ainda mantém melhores índices do que o SVN.

²Diferença entre versões sucessivas de um determinado arquivo.

Supondo, por outro lado, que versões mais antigas dos conteúdos tendam a ser requeridas com menor frequência, ou apenas em situações particulares, a análise realizada sugere ser mais eficiente, do ponto de vista do espaço físico para armazenamento, o uso de ferramentas especificamente concebidas para realizar a deltificação.

A decisão sobre quais versões devem ser mantidas em SGBD pode se apoiar em dados estatísticos ou empíricos. Entretanto, em qualquer situação, tal decisão dependerá do tipo de aplicação. Por exemplo, uma aplicação cujas diferenças de uma versão correspondam sempre a um conjunto de melhorias sobre a versão anterior indica que as mais recentes sejam aquelas com maior volume de acesso e, portanto, devam ser mantidas em SGBDs.

Os resultados apontam para a concepção de um ambiente de armazenamento híbrido, cujas versões mais utilizadas sejam mantidas por SGBD e as menos acessadas em sistema SVN. A justificativa para a integração de um sistema de versionamento a um SGBD se deve ao fato destes últimos não terem sido concebidos com funcionalidades de deltificação, o que exigiria um investimento considerável em desenvolvimento desse tipo de funcionalidade em sistemas exclusivamente baseados em Bancos de Dados.

Maiores detalhes sobre a análise de desempenho das ferramentas MySQL, PostgreSQL e SVN é fornecido por Júnior (2011). Na seção posterior serão apresentados alguns conceitos relacionados às RPC.

4. Redes de Petri Coloridas

As Redes de Petri (RP) são um instrumento de modelagem que permitem representar aspectos importantes de sistemas computacionais, tais quais: concorrência, controle, conflito, sincronização e compartilhamento de recursos. Além da capacidade de modelar aspectos estruturais de sistemas reais, com as RP é possível executar simulações e usar ferramentas de análise que permitem a validação do sistema antes mesmo de sua implementação [CARDOSO and VALETTE 1997].

Uma RP é um grafo bipartido cujos nós são divididos em duas categorias: lugares e transições. Lugares são utilizados para especificar um estado ou condição de um componente do sistema. As transições representam eventos ou operações. Para interligar os componentes de uma RP, existem dois tipos de arcos direcionados. Arcos de entrada que ligam uma transição a um lugar, e os arcos de saída que ligam um lugar a uma transição.

A dinâmica da rede se apoia no uso de fichas que são armazenadas nos lugares. As fichas podem representar recursos ou condições, descrevendo o estado da rede em um instante no tempo. As fichas são inseridas ou retiradas dos lugares através do disparo de uma transição. Para uma transição ser disparada, é necessário que haja uma quantidade de fichas nos lugares de entrada igual ou superior aos pesos dos seus respectivos arcos de entrada. Após o disparo, fichas são inseridas nos lugares de saída de acordo com o peso dos arcos de saída. É através dos disparos das transições que se pode visualizar o fluxo de controle e de informação no sistema. Um estudo mais completo e aprofundado de RP é fornecido por [CARDOSO and VALETTE 1997].

As Redes de Petri Coloridas (RPC) são uma extensão das RP em que as fichas, diferentemente das RP ordinárias, podem ser representar tipos e estruturas de dados mais complexos e possuem uma linguagem de programação associada. O CPN Tools é um

software que implementa as RPC e permite a visualização de simulações e a geração de relatórios para análise. Maiores detalhes podem ser vistos por Jensen e Kristensen [Jensen and Kristensen 2009]. Devido aos recursos para representação, simulação e análise com as RPC, este trabalho apresenta a estrutura e o comportamento do sistema de armazenamento versionado de CDDs modelados com o uso do CPN Tools.

5. Repositório versionado com acesso eficiente: modelagem e análise

A seguir será apresentado o modelo em RPC de armazenamento proposto para o EPCT Virtual. A rede foi executada de forma exaustiva, a fim de avaliar o comportamento dinâmico do sistema. Inicialmente o repositório encontra-se vazio aguardando a adição de conteúdos.

Para uma melhor organização e manipulação da RPC em estudo, a mesma está organizada em uma árvore de páginas. Uma página pode ser considerada como uma RPC simples, cuja semântica depende de outras páginas e/ou superpáginas [BATISTA et al. 2010].

A Figura 1 apresenta o modelo de armazenamento em mais alto nível. Os eventos disponíveis para esta RPC estão melhor detalhados em outras páginas do modelo.

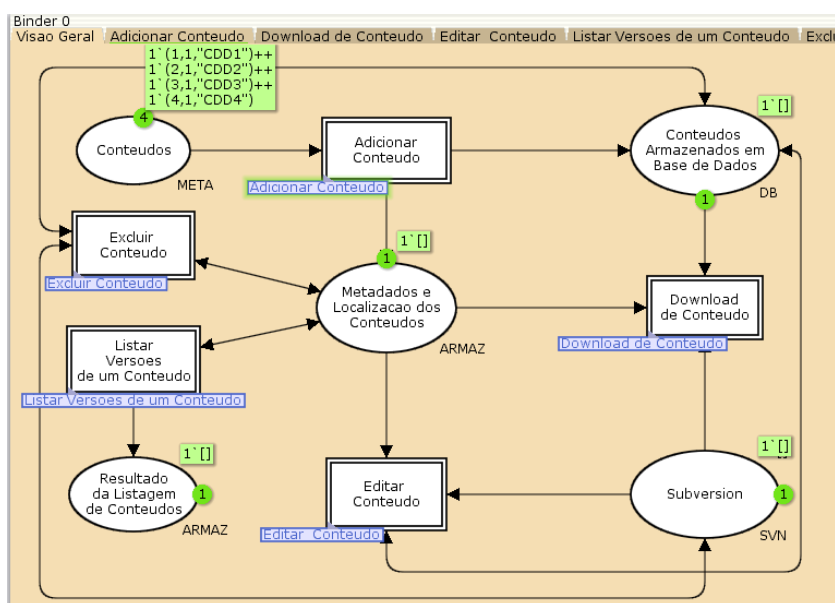


Figura 1. Visão geral do modelo de armazenamento versionado de conteúdos.

Os estados e eventos desta rede são apresentados a seguir:

- Lugar “Conteúdos” modela o estado inicial do sistema, onde o repositório está vazio e aguardando a adição de arquivos. Cada registro é representado por uma ficha de cor “META”, que é o produto entre um identificador único (ID) do conteúdo, um ID da versão do conteúdo e uma descrição para o mesmo.
- Transição “Adicionar Conteúdo” (circundada por verde e única habilitada para o sistema em estado inicial) modela a adição de conteúdos no repositório.
- Lugares “Base de Dados” e “Controle de Versão” modelam os conteúdos armazenados em um SGBD e SCV, respectivamente.

- Transição “Editar Conteúdo” modela o processo de criação de novas versões (versionamento) de um arquivo. Este evento exige maior atenção, sendo posteriormente detalhado.
- Transição “Listar Versões de um Conteúdo” descreve a ação de busca e exibição de versões de um mesmo conteúdo para o usuário.
- Lugar “Resultado da Listagem de Conteúdos” recebe fichas após o disparo da transição “Listar Versões de um Conteúdo”. Este lugar modela o resultado de busca de conteúdos e suas revisões.
- Transição “Download de Conteúdo” modela a cópia de conteúdos do repositório para o cliente. Este evento é capaz de fornecer informações estatísticas de demanda por conteúdos.
- Transição “Excluir Conteúdo” descreve o processo de exclusão de arquivos e seus metadados do repositório.

A seguir serão detalhados os eventos “Adicionar Conteúdo”, “Editar Conteúdo” e “Excluir Conteúdo”. As demais, mas não menos importantes, tarefas serão detalhadas com a continuação deste trabalho.

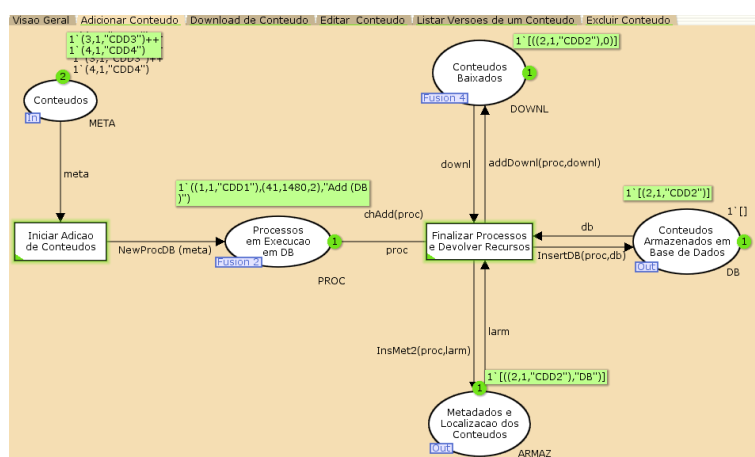


Figura 2. Detalhes do evento “Adicionar Conteúdo”.

A Figura 2 apresenta em detalhes o evento “Adicionar Conteúdo”. O disparo da transição “Iniciar Adição de Conteúdos” adiciona fichas do tipo PROC no lugar “Processo em execução em DB”. Fichas deste tipo armazenam informações sobre o conteúdo, a operação em andamento e o consumo de recursos para a realização da tarefa, e representam os processos em execução na base de dados. O disparo da transição “Finalizar Processos e Devolver Recursos” conclui o ciclo de adição de um conteúdo, composto pela adição do conteúdo e seus metadados na base de dados, e libera os recursos de *hardware* utilizados na operação. A demanda por recursos em operação de adição de arquivos está baseada no trabalho de Júnior [Júnior et al. 2011].

A Figura 3 descreve o evento “Editar Conteúdo”. Por se tratar de um ambiente de armazenamento distribuído, a RPC guarda a localização de onde estão salvos os CDDs previamente adicionados ao sistema. Esta funcionalidade é modelado pelo lugar “Metadados e Localização dos Conteúdos”.

O disparo da transição “Determinar conteúdo para Edição” seleciona do lugar “Metadados e Localização dos Conteúdos” um conteúdo para edição. O conteúdo selecio-

nado está representado por uma ficha no lugar “Conteúdo para Edição”. O próximo passo no armazenamento de nova versão de um arquivo acontece com o disparo da transição “Salvar Conteúdo em TMP” que salva o novo conteúdo em um diretório temporário do sistema, modelado pelo lugar “Armazenamento Temporário de Novas Versões”, e adiciona fichas, do tipo PROC, em “Processos em Execução em SCV”.

Novamente, fichas deste tipo armazenam informações sobre o conteúdo, a operação em andamento e o consumo de recursos para a realização da tarefa, e representam os processos em execução no SCV. Dando seqüência ao versionamento, o disparo da transição “Remover Conteúdo anterior de DB e Adicioná-lo em SVN” descreve a ação de resgatar do SGBD a versão do conteúdo em edição e adicioná-lo ao SVC. Isso se deve ao fato de versões posteriores que um conteúdo serem, constantemente, mais utilizadas que versões anteriores do mesmo arquivo [Júnior et al. 2011].

O passo seguinte, realizado pelo disparo da transição “Salvar Novo Conteúdo em DB” recupera a nova versão do conteúdo salva no diretório temporário e o adiciona na base de dados. Concluído o ciclo de edição de conteúdos, o lugar “Metadados e Localização dos Conteúdos” armazena a localização atualizada de todos os arquivos e suas revisões mantidos no sistema.

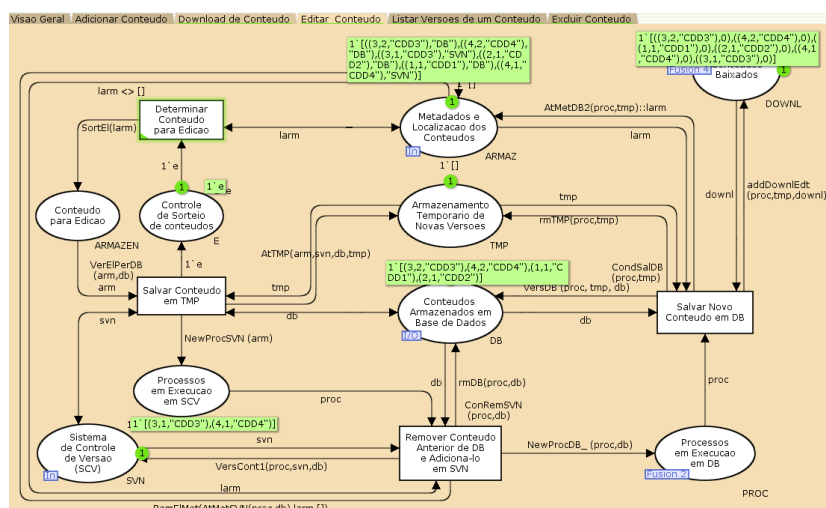


Figura 3. Detalhes do evento “Editar Conteúdo”.

A Figura 4 apresenta dois passos de simulação do evento “Excluir Conteúdo”. O disparo da transição “Determinar conteúdo para Exclusão” seleciona do lugar “Metadados e Localização dos Conteúdos” um conteúdo para exclusão. Segundo a Figura 4 (a) o próximo arquivo a passar pelo processo de exclusão é representado pela ficha ((2,1,“CDD2”),SVN), ou seja, primeira versão do conteúdo com ID 1, cuja descrição é CDD2. É possível observar, também pela ficha, que esse conteúdo está armazenado em um SCV.

A tarefa “Excluir Conteúdo” possui peculiaridades podendo apresentar dois comportamentos distintos. O autor pode excluir a última versão de um arquivo que é, portanto, armazenada em SGBD, ou uma revisão deste conteúdo, armazenada em SCV. A exclusão de um conteúdo de SGBD requer a substituição deste pela sua última revisão armazenada em SCV, enquanto que a exclusão de um conteúdo de SVC deve ser feita em cadeia, ou

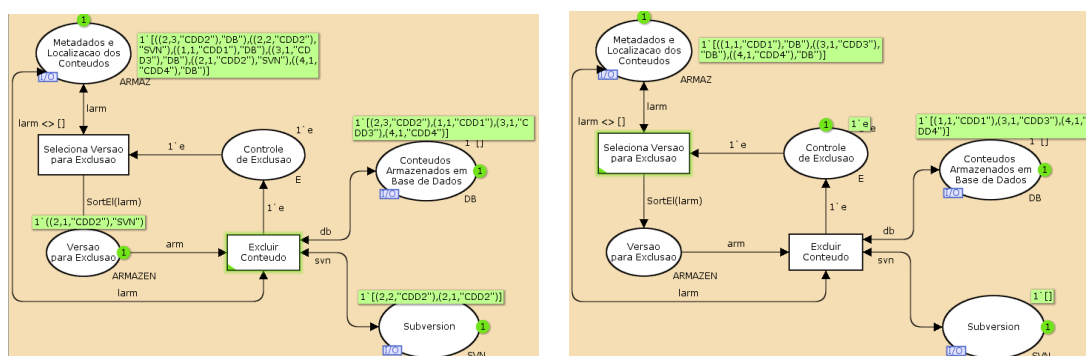


Figura 4. Detalhes do evento “Excluir Conteúdo” (a) antes e (b) depois da exclusão de um arquivo.

seja, todas as versões originadas a partir da versão escolhida para exclusão devem ser eliminadas do sistema.

Após o disparo da transição “Excluir Conteúdo”, segundo a Figura 4 (b) todos CDDS iguais, e originados a partir de (2,1,“CDD2”) serão excluídos mantendo-se, assim, o desejo do autor em retirar seu material do repositório.

A seguir serão apresentados os resultados obtidos e realizadas as considerações finais.

6. Resultados Obtidos e Conclusões

Ao longo de toda a simulação da rede, diversos dados considerados importantes para a análise do desempenho do sistema foram coletadas de Júnior [Júnior et al. 2011]. Os resultados apontam que o uso do formalismo proposto é capaz de determinar políticas de controle, de análise e de identificação de conflitos com facilidade antes da implementação do sistema.

Neste trabalho apresentamos um modelo de armazenamento versionado, que permite o reuso e acesso eficiente, de conteúdos. Foram modeladas, levando em conta critérios de desempenho, os principais eventos relacionados ao armazenamento versionado de conteúdos digitais: Adição, Exclusão, Edição, Listagem e Download de conteúdos.

Foi observado que o modelo se comportou segundo a especificação, ou seja, ocorreu a adição de conteúdos, observada pela adição de fichas no lugar “Conteúdos Armazenados em Base de Dados” ; ocorreu edição de conteúdos, observada pela substituição de fichas do lugar “Conteúdos Armazenados em Base de Dados” representando as novas versões dos conteúdos, estando o conteúdo anterior depositado em “Sistema de Controle de Versão”; O evento “Excluir Conteúdo” também está de acordo com as especificações do sistema e pôde ser observado pela remoção de fichas dos lugares “Sistema de Controle de Versão” e/ou “Conteúdos Armazenados em Base de Dados”.

Assim, o modelo RPC se comportou de forma eficiente e várias situações práticas no funcionamento do sistema de armazenamento versionado de conteúdos foram simuladas, utilizando-se o modelo proposto. Portanto, se a especificação e validação do sistema foram obtidas e preservados em um modelo RPC, espera-se que a implementação do mo-

delo apresente bons resultados.

O próximo passo deste trabalho concentra-se na implementação do sistema de armazenamento versionado de conteúdos no Portal EPCT Virtual.

Referências

- ALLEN, C. and MUGISA, E. (2010). Achieving learning object interoperability e reusability through object orientation. *Education Technology e Computer (ICETC), 2010 2nd International Conference on*, 2:22–24. Junho, 2010.
- BATISTA, Í., BARROSO, G. C., ALMEIDA, O. D. M., VARELA, A. T., and SOUSA, J. R. B. (2010). Navegação de robôs móveis com ênfase em planejamento e supervisão de trajetórias. *XVIII Congresso Brasileiro de Automática*.
- BOHL, O., SCHEUHASE, J., SENGLER, R., and WINAND, U. (2002). The sharable content object reference model (scorm) - a critical review. *Computers in Education: Proceedings. International Conference on*, pages 950–951. Dec. 2002.
- CARDOSO, J. and VALETTE, R. (1997). *Redes de Petri*. Ed. Da UFSC.
- COLLINS-SUSSMAN, B., FITZPATRICK, B. W., and PILATO, C. M. (2007). *Controle de Versão com Subversion*. O'Reilly.
- CVS (2011). Online: disponível em <http://www.nongnu.org/cvs/>. Acesso em Janeiro de 2011.
- EPCT Virtual (2010). Online: disponível em <http://www.redenet.edu.br/geral/siep1.php>. Acesso em Novembro de 2010.
- GIRARDI, R. A. D. (2004). Framework para coordenação e mediação de web services modelados como learning objects para ambientes de aprendizado na web. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio. páginas 14-26.
- Jensen, K. and Kristensen, L. M. (2009). *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer.
- Júnior, C., BONETTI, A., HONÓRIO Filho, P., CUNHA, R., SIQUEIRA, R., SOARES, J., BEZERRA, F., and Bezerra, E. (2009). Integração entre ferramentas e instituições: um modelo de colaboração para o desenvolvimento da educação profissional e tecnológica. *XX Simpósio Brasileiro de Informática na Educação*. Florianópolis.
- Júnior, C., Gomes, D., Soares, J., and Barroso, G. (2011). Análise de desempenho para o armazenamento versionado de conteúdos didáticos digitais binários e acesso eficiente via web. In *CPNTE2011-1*.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.
- Muzio, J. A., Heins, T., and Mundell, R. (2002). Experiences with reusable e-learning objects: From theory to practice. *The Internet e Higher Education*, Vol. 5:21–34.
- RIVED (2010). Rede internacional virtual de educação. Online: disponível em <http://www.rived.mec.gov.br>. Acesso em Janeiro de 2010.

SCHEER, S. and GAMA, C. L. G. (2004). Construção de um repositório para objetos educacionais hiperídia. *Congresso Nacional de Ambientes Hiperídia para Aprendizagem – CONAHPA*, page 8 p. Florianópolis - SC.

SEED-MEC (2010). Secretaria de educação a distância - ministério da educação (mec). Online: disponível em <http://portal.mec.gov.br>. Acesso em Dezembro de 2010.

SUBVERSION (2011). Open source software engineering tools. Online: disponível em <http://subversion.tigris.org>. Acesso em Janeiro de 2011.