

Um Ambiente de Suporte ao Trabalho Cooperativo

Carlos A. T. Aguiar Ana Carolina Salgado

Departamento de Informática
Universidade Federal de Pernambuco
Cx. Postal 7851, 50732-970 Recife - PE - Brasil
e-mails: cat@di.ufpe.br & acs@di.ufpe.br

Abstract

The use of computer to intermediate and facilitate human interaction has increased at present. Tools that support several people working together have been defined to assist this workgroups in coloboration, communication and coordination of their colective tasks. The Cooperative Environment proposed here, based in a simplified cooperation model, assist not only a specific activiçy, but a wide range of cooperative work applications.

Keywords

Computer-Supported Cooperative Work, Groupware, Cooperação, Colaboração, Comunicação, Coordenação, Modelo de Cooperação, Versões.

1 Introdução

Atividades coletivas são caracterizadas pela divisão de tarefas e troca de informações entre os participantes. Dentro deste escopo, a cooperação entre os integrantes do grupo e o compartilhamento de dados entre eles assumem um papel de extrema importância.

Computer-Supported Cooperative Work (CSCW) ou Computação Colaborativa, um dos mais novos tópicos em computação, focaliza o papel do computador na cooperação, no compartilhamento da informação, no trabalho em grupo [02]. CSCW fez surgir uma nova classe de aplicações: Groupware.

Um ambiente de suporte ao trabalho cooperativo reúne as ferramentas de software que auxiliarão os usuários (integrantes de um grupo) na realização das ações cooperativas (projeto), assistindo-os na comunicação, na colaboração (manipulação e compartilhamento da informação) e na coordenação de suas atividades (gerência do desenvolvimento do projeto).

Problemas surgem quando verificamos que os requisitos de suporte à cooperação (comunicação e colaboração entre os usuários, coordenação, controle de versões, etc) não são adequadamente atendidos até mesmo pelos ditos Sistemas de Banco de Dados não-convencionais [13]. Como consequência, *Modelos de Cooperação* têm sido definidos ([11, 13, 15, 05]) para atender às exigências do trabalho cooperativo em ambientes de desenvolvimento.

Propomos aqui um Ambiente de Suporte ao Trabalho Cooperativo que está fundamentado em um modelo de cooperação simplificado e que busca satisfazer não só às necessidades

de uma aplicação específica (CAD, por exemplo) mas de várias atividades de natureza coletiva.

O Processo de Cooperação, que inclui a definição do projeto, o mecanismo de versões e o modelo de cooperação é apresentado na próxima seção. Na sequência, descrevemos o Ambiente Cooperativo, identificando seus principais componentes. Por fim, concluímos com algumas considerações.

2 O Processo de Cooperação

Colaboração é um sinal de maturidade [10]. Uma idéia é primeiro desenvolvida para ser, então, forçada à comunidade. A existência de ferramentas de grupo não obscurece a necessidade de aplicações individuais. O desenvolvimento individual persiste, só que atingimos um nível onde a divisão de tarefas (ou objetivos) tornar-se-á essencial para a obtenção de um resultado satisfatório e em tempo hábil.

O processo de cooperação se inicia, exatamente, nesta divisão do projeto em subprojetos e na atribuição de papéis (ou funções) aos seus integrantes. Para suportar esta divisão de tarefas e a conseqüente interação entre vários usuários, temos que definir um *Modelo de Cooperação*. Este, por sua vez, está fundamentado em um outro modelo, o *Modelo de Versões*.

2.1 A Definição do Projeto

Modelamos a aplicação de projeto como uma hierarquia de (sub)projetos em forma de árvore ([15, 16, 17]). Os nodos da árvore representam as tarefas a serem realizadas no âmbito do projeto, enquanto os ramos especificam as relações de dependência do tipo tarefa/subtarefa (ou objeto/subobjeto). A divisão do projeto em um conjunto de subprojetos é repetida até que os subprojetos atinjam o nível de complexidade desejado.

Cada usuário pertencente ao projeto tem pelo menos um papel (ou função) associado [08]. Identificamos quatro possíveis funções dentro do projeto: **coordenador**, **projetista**, **subcoordenador** e **revisor**. Todo projeto tem um coordenador (normalmente aquele que sugere a definição de um projeto) e vários projetistas. A figura do subcoordenador só aparece quando um subprojeto é novamente dividido. A este subprojeto associamos um subcoordenador. Já os projetistas estão relacionados aos subprojetos terminais da árvore. O revisor tem a função específica de comentar documentos já liberados (versões não-privativas), sugerindo modificações aos seus autores.

À parte da estrutura hierárquica do projeto, definimos uma ordem parcial de desenvolvimento para os subprojetos, estabelecendo uma dependência temporal entre eles. Além disso, cada subprojeto identificado tem um *deadline* para conclusão e um conjunto de objetivos a serem cumpridos.

Podemos mostrar, através de um exemplo, como seria definido um projeto. Este exemplo não pretende ser completo, mas deve ser o suficiente para se ter noção do processo de definição de um projeto.

O projeto-exemplo envolve a aquisição e análise de sinais EEG (eletroencefalográficos) [01]. Em síntese o sistema TecnoLab compreende:

- Conversão analógico-digital, que transforma uma variação de tensão em um sinal digital;
- Aquisição e amostragem (visualização) do sinal digital, em tempo-real;
- Análise do sinal adquirido em um determinado intervalo de tempo.

O objetivo do projeto é, em linhas gerais, pegar o sinal analógico, convertê-lo para digital (de forma que o computador possa tratá-lo), fazendo sua aquisição e amostragem durante um certo intervalo de tempo para posterior análise.

Uma primeira divisão em subprojetos identifica três tarefas (ou objetivos), que correspondem aos itens anteriormente especificados, ou seja, conversão, aquisição/amostragem e análise.

Apesar da aquisição/amostragem estarem reunidas em um único subprojeto, pois estão intimamente relacionadas, podemos pensar em mais um nível de detalhamento, dividindo esta tarefa em duas subtarefas:

- aquisição do sinal digital;
- amostragem (visualização) deste sinal;

Deste modo, podemos pensar em um módulo de amostragem que recebe um valor qualquer e transforma-o em um ponto gráfico. Não sendo necessário qualquer outro desdobramento, chegamos a hierarquia de projetos mostrada na figura 1.

De acordo com a hierarquia de subprojetos, associamos e definimos os papéis dos integrantes do grupo (como pode ser visto na figura 1):

- Um **coordenador** tem a função de coordenar o projeto como um todo;
- Ao subprojeto *conversão A/D* podemos associar um ou mais **projetistas**;
- Como o subprojeto *aquisição/amostragem* foi subdividido, definimos a figura do **sub-coordenador** neste nível, e para cada subprojeto deste associamos um ou mais **projetistas**;
- À *análise* também podemos associar um ou mais **projetistas**.

Os objetivos são descritos informalmente, apesar de aplicações de teste e validação poderem ser utilizadas sobre as versões produzidas. Por exemplo, para este sistema ser viável, devemos verificar se o tempo gasto na amostragem de um ponto não é maior que o tempo entre duas aquisições sucessivas. Isto pode ser determinado verificando o tempo gasto pelo módulo de visualização para amostrar um ponto. Se esta restrição não estiver garantida, o objetivo ainda não terá sido atingido.

Definido o projeto, partimos para o seu desenvolvimento. Todo o trabalho envolvido é realizado sobre versões de objetos, as quais são resultado dos vários subprojetos. Para suportar o conceito de versões, definimos um *Modelo de Versões*.

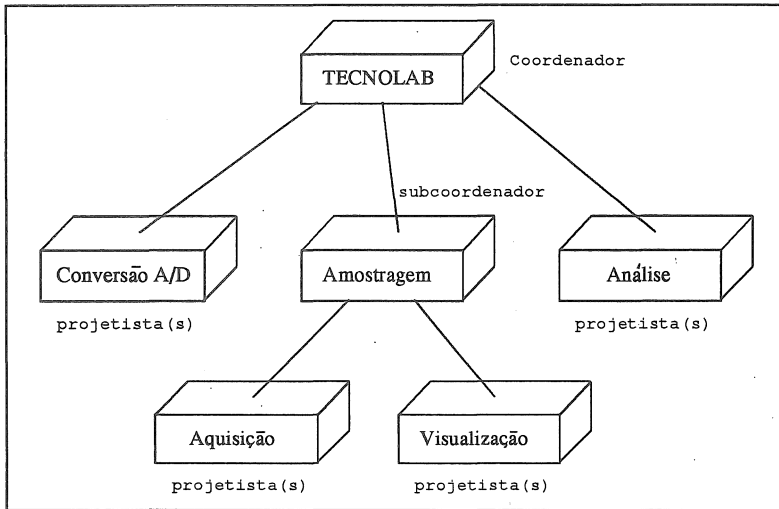


Figura 1: Hierarquia de projetos do TecnoLab

2.2 Modelo de Versões

Durante o desenvolvimento de um projeto, os subprojetos deste passam por vários estágios de implementação, gerando o que chamamos de versões de objeto. Depois da criação inicial do(s) objeto(s) do subprojeto, novas versões deste objeto podem ser derivadas, e outras podem ser derivadas destas, formando uma hierarquia de derivação de versões para o objeto. Uma hierarquia de derivação de versões captura a evolução do desenvolvimento e indica a ordem parcial das versões do objeto [04].

Em nosso modelo, identificamos três tipos de versões, em relação a sua disponibilidade e aos tipos de operação que podem ser realizadas sobre elas: *versões privadas*, *versões de (sub)projeto* e *versões públicas* [14].

Uma versão privada tem as seguintes propriedades:

- ela só pode ser consultada e atualizada (inclusive eliminada) pelo usuário que a criou; quando mais de um projetista está associado a um mesmo subprojeto haverá mais de um autor, e aí teremos o compartilhamento da versão privada;
- uma nova versão privada pode ser derivada de uma outra versão privada já existente. A versão existente pode, então, ser promovida a versão de (sub)projeto;
- ela é armazenada em uma área acessível apenas ao usuário (ou usuários) que a criou (banco de dados privado).

Versões de (sub)projeto têm as seguintes características:

- ela pode existir tanto a nível de projeto como a nível de subprojeto;
- ela não pode ser atualizada;
- ela pode ser removida pelo seu criador;
- uma versão privativa pode ser derivada de uma versão de (sub)projeto;
- uma versão de subprojeto é gerada a partir da promoção (explícita ou implícita) de uma versão privativa; uma versão de projeto pode ser criada a partir de uma versão privativa ou de subprojeto.

Versões acessíveis a todos os usuários, de qualquer projeto, são versões públicas. Como versões de (sub)projeto, elas não podem ser atualizadas, podem ser removidas pelos seus criadores e podem originar uma versão privativa.

Uma versão privativa pertence ao usuário que a criou, e a sua existência não será de conhecimento dos outros usuários do projeto até que ela seja promovida a versão de (sub)projeto, liberada pelo seu criador e avalizada pelo (sub)coordenador. Uma versão de subprojeto é definida quando um subprojeto não está diretamente ligado à raiz da hierarquia de (sub)projetos. O subprojeto que originou esta divisão agrupa tais versões. Tomando como exemplo o projeto definido anteriormente (figura 1), teremos versões de subprojeto associadas ao módulo *aquisição/amostragem*, originadas dos subprojetos *aquisição* e *amostragem*.

Tendo à disposição uma versão de (sub)projeto, todos os usuários associados a subprojetos derivados na hierarquia podem utilizá-la como tal, mas apenas com permissão de leitura, sem possibilidades de atualização. No entanto, como identificado anteriormente, versões privativas poderão ser geradas a partir destas e, portanto, atualizadas sem qualquer alteração nas versões de (sub)projeto que as originaram.

2.3 Modelo de Cooperação

Modelos de cooperação têm sido propostos ([11, 13, 15, 05]) não só para suprir a falta de controle automático de consistência no acesso concorrente ao banco de dados, gerada pela falta de isolamento e atonicidade das transações cooperativas, mas também com o objetivo de definir novos critérios tanto para o controle da cooperação entre projetistas (usuários de um grupo) como para a gerência do desenvolvimento do projeto como um todo [12].

Em nosso modelo, projetistas interagem através da utilização de versões de objetos disponíveis a todos usuários associados a subprojetos derivados na hierarquia (versões de (sub)projeto). Ao interagirem através de uma versão de (sub)projeto, os projetistas assumem ou o papel de criador ou o de usuário da versão. Ao usuário da versão cabe o direito de utilizá-la, mas sem possibilidades de alteração; ao criador, a condição de removê-la, mas só quando esta não estiver sendo usada por qualquer outro projetista.

Nosso modelo prevê o compartilhamento de áreas privativas, quando mais de um projetista esta engajado em um mesmo subprojeto. Neste caso, um dos autores bloqueia uma

versão de projeto quando passa a utilizá-la. Este bloqueio é mantido durante toda a atividade de atualização, podendo ser retirado após um período que indique a falta de uso (*idle period*). Os outros usuários que compartilham a área serão notificados do bloqueio desta versão.

Além disso, o modelo permite, através da estrutura hierárquica de derivação de versões, que o processo evolucionário da construção dos objetos de projeto esteja disponível aos projetistas. Assim, eles têm uma visão global do projeto, bem como de sua evolução. Estas e outras funções são atribuições do *Gerenciador de Cooperação* que mantém o estado do projeto através da manutenção e controle do estado dos seus subprojetos.

Com este enfoque, obtemos um modelo de cooperação genérico o bastante para atender às necessidades dos usuários no desenvolvimento de software, CAD, co-autoria, etc, atividades cooperativas por natureza.

3 O Ambiente de Cooperação

Ambientes voltados para o trabalho em grupo referem-se a sistemas de software que suportam dois ou mais usuários trabalhando, simultaneamente ou não, numa tarefa comum e que provê uma interface do ambiente compartilhado ([07, 06]). Neste ambiente compartilhado as ações de um usuário são visíveis aos outros integrantes da equipe de trabalho. Seu objetivo é assistir grupos de usuários na comunicação, colaboração e coordenação de suas atividades.

A existência de um espaço compartilhado não elimina a necessidade de um espaço privativo, espaço ao qual apenas o usuário tem acesso, e onde não haja restrições quanto ao uso de aplicações diversas. A liberdade de escolha do usuário deve ser mantida.

Um ambiente de suporte ao trabalho cooperativo deve fornecer subsídios para a manipulação e compartilhamento da informação (colaboração), coordenação das atividades cooperativas (gerência do desenvolvimento) e comunicação entre os participantes de um projeto. O ambiente proposto consegue suprir essas necessidades através da implementação de três módulos: o *Gerenciador da Informação*, o *Gerenciador de Cooperação* e o *Sistema de Comunicação*. Descrevemos a seguir, de forma sucinta, cada um deles.

3.1 Gerenciamento da Informação

Nosso ambiente está definido em termos de projetos e de usuários (projetistas, revisores, subcoordenadores e coordenadores) que cooperam dentro de cada projeto. Um projeto reúne, portanto, um grupo de usuários e um conjunto de objetos (que podem ser privativos, de projeto ou públicos). Os objetos (versões dele) no banco de dados de (sub)projeto estão disponíveis a todos aqueles usuários ligados ao (sub)projeto, enquanto os objetos no banco de dados privativo são de propriedade e estão acessíveis apenas a seu(s) autor(es). Um projetista pode compartilhar dados (versões de objetos) com outros projetistas do mesmo grupo através da área de (sub)projeto [04].

A hierarquia do banco de dados (BD) em nosso ambiente inclui os bancos de dados privativos, os bancos de dados de subprojeto e projeto, e o banco de dados público. Além

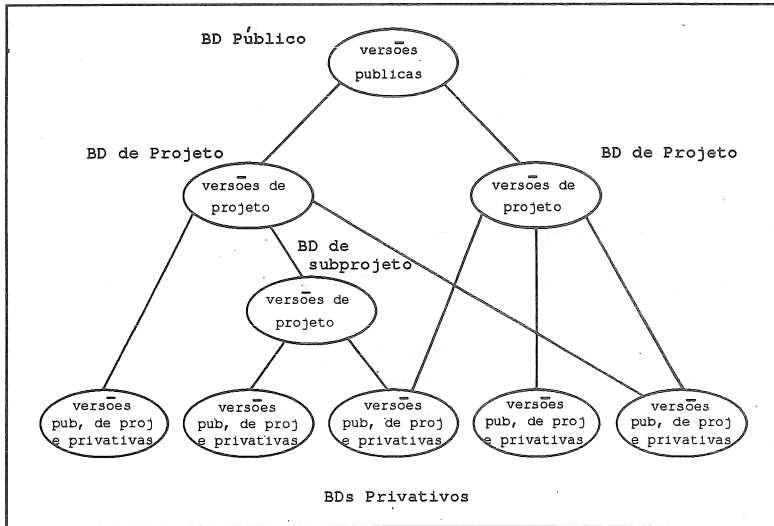


Figura 2: Hierarquia do Banco de Dados do Ambiente

destes, um banco de dados de controle mantém informações relevantes ao ambiente, mas não faz parte desta hierarquia. A figura 2 representa um exemplo de hierarquia.

Versões de (sub)projeto podem existir tanto no BD de (sub)projeto como no BD privativo; já versões privativas só estão definidas nos BDs privativos. Os BDs privativos são compartilhados por mais de um projeto quando o usuário integra mais de um grupo de projeto; são compartilhados por mais de um projetista (apenas a parte comum ao subprojeto) quando estes estão associados ao mesmo subprojeto.

Bancos de dados privativos estão armazenados na área do usuário, enquanto os de (sub)projeto estão distribuídos em uma rede local de forma a otimizar o acesso à informação por parte dos projetistas.

3.2 Gerenciamento da Cooperação - a interface do espaço compartilhado

O Gerenciador de Cooperação é o responsável pela manutenção do estado atual do projeto, permitindo que ações de um usuário sejam visíveis aos outros projetistas do grupo [12]. Entendemos como *ações visíveis* aquelas que afetam o (sub)projeto em questão, mas que não fazem parte do espaço privativo do usuário.

O usuário desenvolve sua tarefa dentro de uma *sessão de trabalho*. Nesta sessão o usuário passa a ter um papel (ou função) bem definido. A atribuição de papéis (ou funções) aos usuários integrantes de um projeto permite que o gerenciador de cooperação determine

o nível de acesso à informação, a partir do controle da interface do ambiente compartilhado.

Informar os usuários sobre o estado de suas ações, das dependências e objetivos atingidos, bem como notificá-los de acordo com a necessidade, são atribuições do gerenciador de cooperação.

A interface compartilhada dá a exata noção do estado atual do (sub)projeto e fornece ao usuário as ferramentas necessárias à cooperação entre os projetistas.

3.3 Sistema de Comunicação

O Módulo de Comunicação visa suprir as necessidades básicas de comunicação entre os membros de uma equipe que realizam um trabalho conjunto [03].

Aproveitando o ambiente de cooperação criado e controlado pelo Gerenciador de Cooperação, os usuários da rede (e pertencentes a um projeto) podem trocar pequenas mensagens de forma rápida e simples. A definição de papéis permite também que usuários referenciem outros integrantes do grupo, através da função deles dentro do projeto (por exemplo, um projetista se comunica com o coordenador de projeto sem a necessidade de saber quem ele é).

O mecanismo de e-mail fornecido pelo sistema operacional está sendo adaptado e expandido para o tratamento de documentos multimídia, especificamente o que diz respeito a manipulação (criação, “visualização”, armazenamento e envio) de sons, com o objetivo de tornar mais natural (e poderosa) a troca de informações entre os usuários do ambiente cooperativo. Um elemento sonoro é uma passagem de áudio digitalizada, fornecida ao ambiente por um Servidor de Áudio e manipulado pelo usuário através de uma interface gráfica que, funcionalmente, se assemelha a um gravador portátil.

Anotações associadas a versões de objetos são tratadas independentemente, já que versões de projeto não são atualizáveis. Através delas, projetistas podem propor e solicitar modificações de objetos aos seus criadores.

Por fim, um ambiente de conferência com protocolo preemptivo deve ser incorporado. Nele, todos os participantes têm a mesma visão da janela de comunicação (WYSIWIS¹) e apenas um deles tem o direito de “fala” por vez [03].

4 Conclusão

Uma proposta de Ambiente de Suporte ao Trabalho Cooperativo foi apresentada de forma resumida. O Ambiente tem por objetivo auxiliar grupos de pessoas trabalhando de forma cooperativa, fornecendo mecanismos de colaboração, coordenação e comunicação.

Sua simplicidade no tratamento da cooperação segue a premissa de que não é necessário (nem interessante) mudar a forma de interação entre as pessoas no trabalho em grupo; é necessário sim, entender como isso se processa e permitir a improvisação característica das atividades coletivas [09].

¹ *What You See Is What I See*

Referências

- [01] C.A.T. Aguiar, F. Borges & J. Ceppola, "Sistema Automatizado para Registro e Análise de Sinais EEG", I Congresso Brasileiro de Informática em Saúde, Campinas, 1986
- [02] C.A.T. Aguiar & A.C. Salgado, "Computer-Supported Cooperative Work - Uma Visão Geral", Relatório Técnico, Depto. de Informática - UFPE, Recife, 1992
- [03] L.S. Castro & R. Drummond, "Sistrac: Sistema de Suporte a Trabalho Cooperativo", Dissertação de Mestrado, DCC - Unicamp, Campinas, 1991
- [04] H.T. Chou & W. Kim, "A Unifying Framework for Versions in a CAD Environment", Proceedings of the 12th International Conference on VLDB, Kyoto, Japan, 1986
- [05] M. Dürr & K. Rösch, "A Model for Cooperative Transactions", VII Simpósio Brasileiro de Banco de Dados, Porto Alegre, 1992
- [06] C.A. Ellis, S.J. Gibbs & G.L. Rein, "Groupware: Some Issues and Experiences", Communications of the ACM, Vol 34, Number 1, January 1991
- [07] S.J. Gibbs, "LIZA: An Extensible Groupware Toolkit", CHI'89 Proceedings, Austin, 1989
- [08] I. Greif & S. Savin, "Data Sharing in Group Work", in Computer-Supported Cooperative Work: A book of Readings, Edited by Irene Greif, Morgan Kaufmann Publishers, 1988
- [09] J. Grudin, "Why CSCW Applications Fail: problems in the Design and Evaluation of Organizational Interfaces", CSCW'88 Proceeding, Portland, Oregon, 1988
- [10] J. Grudin, "CSCW : Introduction", Communications of the ACM, Vol 34, Number 12, December 1991
- [11] C. Iochpe, et al., "Um Modelo de cooperação baseado em características de objetos e em ordens parciais de passos de projeto", Relatório de Projeto, CPGCC - UFRGS, 1991
- [12] C. Iochpe, et al., "Suporte à Cooperação em Ambiente de Projeto Assistido por Computador", VI Simpósio Brasileiro de Banco de Dados, Manaus, 1991
- [13] W. Käfer, "A Framework for version-based Cooperation Control", Research Report, Universität Kaiserslautern, Alemanha, 1990 (submetido a publicação)
- [14] A.M.B. Martiniano, "Gerenciamento de Versões de Objetos", Relatório Técnico, Depto de Informática - UFPE, Recife, 1992
- [15] M.H. Nodine, et al., "Cooperative Transaction Hierarchies", Technical Report, no. CS-90-03, Brown University, 1990
- [16] M.H. Nodine, et al., "Synchronization and Recovery in Cooperative Transactions", Proceedings of the 4th International WorkShop on Persistent Object System Design, Implementation and Use, 1990
- [17] A.H. Skarra, "Localized Correctness Specifications for Cooperative Transactions in an Object-Oriented Database, Office Knowledge Engineering, vol. 4, 1990