

Suporte por Computador ao Desenvolvimento Cooperativo de Software: Classificação e Propostas

Renata Mendes de Araujo

(renata,mdias)@cos.ufrj.br

Márcio de Sousa Dias

Marcos Roberto da Silva Borges

mrborges@nce.ufrj.br

COPPE - Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro
Caixa Postal 6851 Rio de Janeiro - RJ
21945-970

Núcleo de Computação Eletrônica
Instituto de Matemática
Universidade Federal do Rio de Janeiro
Caixa Postal 2324 Rio de Janeiro - RJ
20001-970

Resumo: O objetivo deste trabalho é o de apresentar um esquema de classificação dos principais aspectos envolvidos no apoio ao desenvolvimento cooperativo de software. Partindo da visão do processo de desenvolvimento como um processo contínuo de entendimento, essa classificação se baseia em quatro aspectos necessários para seu suporte: a comunicação, o registro da memória de grupo, os mecanismos de percepção e a coordenação. A partir desta classificação, descrevemos as abordagens de suporte à cooperação utilizadas por ambientes e ferramentas propostas até o momento atual.

Palavras-Chave: Desenvolvimento Cooperativo de Software, CSCW, *groupware*, ambiente de engenharia de software.

1. Introdução

Já não se questiona mais a característica cooperativa de processos de desenvolvimento de software, uma vez que este processo é marcado por um grande número de interações. Além disso, a natureza e a forma de realização destas interações varia ao longo da execução do processo. Cada vez mais se reconhece a necessidade de melhorar a qualidade destas interações como ponto fundamental para o aumento da qualidade dos produtos gerados e para promover o agenciamento desta produção.

A construção de um produto de software é um processo que envolve grupos de atores com interesses e visões distintas mas em direção a um só objetivo. Coordenar estes interesses e possibilitar que estes atores trabalhem harmonicamente pode parecer apenas uma questão gerencial. Mas, nota-se que toda uma filosofia de trabalho cooperativo é preciso ser aplicada para que todos os atores envolvidos neste processo possam ter garantias quanto a alcançarem seus objetivos individuais gerando, ao mesmo tempo, um produto útil comum.

As principais questões que norteiam a pesquisa em engenharia de software como reuso, modelagem de processos de software, qualidade de software, entre outras, se beneficiam diretamente dos resultados obtidos com a pesquisa em relação ao suporte ao desenvolvimento cooperativo, uma vez que todas elas compreendem a participação de grupos em sua realização. O próprio modelo de capacitação de software - CMM (*Capability Maturity Model*) - define que, conforme as organizações sobem para níveis mais altos de maturidade, as tarefas de projeto passam de tarefas envolvendo o desenvolvimento individual e isolado para o trabalho em equipes [PATN95].

Embora se reconheça a necessidade de suporte à cooperação durante o desenvolvimento de software, ainda não é possível garantir um suporte eficiente e que promova a convergência deste processo. Buscando soluções para esta questão, um contingente da comunidade de pesquisa em engenharia de software tem procurado aplicar os conceitos e resultados obtidos com as pesquisas que envolvem a área de Trabalho Cooperativo Suportado Por Computador

(CSCW) no contexto de desenvolvimento de software, na tentativa de suportá-lo através de técnicas, ambientes e ferramentas cooperativas [GIBB89].

Compreender as questões que envolvem a participação dos diversos elementos de uma equipe de software é o primeiro passo para prover recursos eficientes para seu suporte. Assim, o objetivo deste trabalho é o de apresentar um esquema para classificação dos principais aspectos envolvidos no apoio ao desenvolvimento cooperativo de software. Partindo da visão do processo de desenvolvimento como um processo contínuo de entendimento, essa classificação se baseia em quatro aspectos necessários para seu suporte: a comunicação, o registro da memória de grupo, os mecanismos de percepção e a coordenação.

Baseados nesta classificação, descrevemos as abordagens de suporte à cooperação utilizadas por ambientes e ferramentas propostas recentemente na literatura. A análise destas abordagens segundo o esquema proposto nos permite avaliar o panorama atual deste nicho de pesquisa, levantando quais questões estão sendo atacadas e quais estão sendo negligenciadas.

Na seção a seguir apresentamos os aspectos de suporte ao desenvolvimento cooperativo de software considerados em nosso estudo, definindo um esquema de classificação das abordagens de suporte existentes. A partir daí, apresentamos e discutimos as abordagens propostas atualmente para este suporte, concluindo com um panorama geral das tendências na aplicação de aspectos colaborativos no processo de desenvolvimento de software.

2. Aspectos de Suporte ao Desenvolvimento Cooperativo de Software

O principal objetivo da engenharia de software é o de estabelecer mecanismos para um desenvolvimento de software com qualidade e produtividade. A qualidade do software está intimamente relacionada com a questão de **entendimento** entre os participantes do processo de desenvolvimento, ou ainda, à convergência de seus pontos de vista. A produtividade, por sua vez, depende de que o entendimento seja alcançado de forma organizada, breve e eficiente.

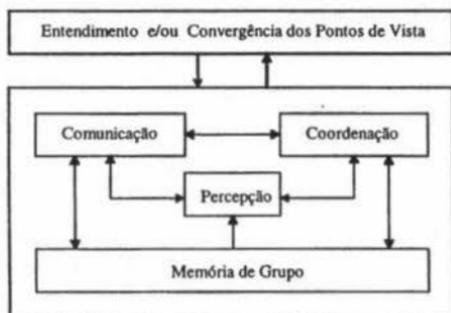


Figura 1 - Esquema Conceitual dos Aspectos de Suporte por Computador ao DCS

Segundo Daft e Lengel [DAFT86], o entendimento entre pessoas está baseado nos conceitos de *incerteza* e *equivocalidade*, cujos graus se objetiva diminuir. Enquanto a incerteza se refere à ausência de informação, a equivocalidade diz respeito à ambigüidade, isto é, a existência de interpretações conflitantes sobre os assuntos tratados pelas pessoas.

O desenvolvimento de software pode ser visualizado como um processo contínuo de redução dos graus de incerteza e equivocalidade, onde a resolução de conflitos e a convergência de pontos de vista de seus participantes determinam seus resultados.

Os membros de uma equipe de desenvolvimento - usuários e desenvolvedores - vivem em ambientes de trabalho complexos definidos por suas áreas de especialização. Quando se reúnem em equipes, a profundidade de experiências e conhecimento de cada um aparece como

partes da base de conhecimento do grupo. Mas, estas diferenças em especialização podem emerrar o processo de entendimento como se estivessem falando línguas diferentes. Para superar este obstáculo, a equipe deve construir uma estrutura para compartilhar seus conhecimentos para que a colaboração entre eles não seja prejudicada por problemas de mal entendimento.

Para suportar eficientemente o processo de colaboração, portanto, é preciso investir em entendimento. Quatro questões concorrem para viabilizar o suporte a este entendimento: a **comunicação** entre os participantes envolvidos no processo, a **coordenação** de suas atividades, o registro do conhecimento comum pela **memória de grupo** e a **percepção** do grupo em relação ao contexto de trabalho sendo realizado. Estas questões não podem ser consideradas monoliticamente, uma vez que se encontram intimamente dependentes e interrelacionadas uma às outras [ARAU97a]. Um diagrama descrevendo o relacionamento entre estes aspectos pode ser visualizado na Figura 1. Neste diagrama, as setas indicam que o suporte a um determinado aspecto pode influenciar ou contribuir para o suporte a outros aspectos do esquema.

Na literatura pesquisada, encontramos descrições de ambientes que se preocupam em oferecer suporte ao processo de colaboração ao longo de todo o processo, englobando todas as atividades básicas e propostas de ferramentas voltadas para o suporte a atividades específicas. Nas seções que se seguem, apresentamos os recursos oferecidos por cada uma destas propostas para o suporte às questões definidas no esquema de classificação. Estes recursos encontram-se relacionados na Tabela 1.

Aspectos	Abordagens de Suporte
Memória de Grupo	<ol style="list-style-type: none"> 1. Representação do conhecimento através de hipertextos 2. Modelos de argumentação 3. Formalização de conceitos 4. Gerência de configuração 5. Armazenamento em bases de objetos
Percepção	<ol style="list-style-type: none"> 1. Recursos de interface 2. Mecanismos de consulta e navegação 3. Uso de anotações 4. Uso de notificação
Comunicação	<ol style="list-style-type: none"> 1. Flexibilidade de modos de interação 2. Uso de espaços de trabalho compartilhados 3. Comunicação indireta através da memória de grupo 4. Uso de sistemas de mensagens
Coordenação	<ol style="list-style-type: none"> 1. Modelagem de processos 2. Encenação de processos 3. Modelagem de usuários 4. Controle de concorrência

Tabela 1- Resumo das abordagens para o suporte ao desenvolvimento cooperativo de software

3. Abordagens de Suporte à Memória de Grupo

O registro dos artefatos de software produzidos ao longo de um projeto é uma questão que norteia as pesquisas em ambientes de desenvolvimento de software há algum tempo. Quando se fala em suporte ao processo de colaboração entre elementos de uma equipe de desenvolvimento, este registro torna-se ainda mais vital. Neste contexto, a modelagem dos dados a serem armazenados se amplia para comportar outras informações provenientes da interação em grupo além dos artefatos gerados. Além disso, o arranjo de informações em bases distribuídas cresce em dimensões para o suporte a interações remotas.

A forma de captura do conhecimento, conforme realizada na maioria dos processos de desenvolvimento, se concentra na preservação de documentos nas mais variadas formas que representam os produtos gerados. Este conhecimento pode ser encarado como o conhecimento formal e é nele que o grupo se baseia como memória de trabalho.

Entretanto, o conhecimento dito informal, que é, a grosso modo, a descrição do processo pelo qual os produtos foram criados, compreendendo o registro das idéias, fatos, questões, pontos de vista, conversas, discussões, decisões etc, que aconteceram no decorrer do processo e acabaram por defini-lo, é difícil de ser capturado. Esta racionalização (*rationale*) do processo e outras formas de conhecimento informal devem estar intimamente relacionados aos artefatos sendo produzidos pois um se baseia no outro para darem sentido ao processo.

Isto significa dizer que é preciso preservar o contexto das interações realizadas ao longo do processo. Este contexto apresenta a forma de uma rede de informações que incluem fatos, hipóteses, restrições, decisões e suas razões, o significado de conceitos e, é claro, os documentos formais, formando aquilo que é denominado de memória de grupo [CONK96].

A memória de grupo, portanto, é o registro de todo o processo de interação do grupo (*memória do processo*), incluindo a comunicação realizada e os passos desencadeados; e os produtos gerados (*memória do produto*). É sobre os elementos armazenados na memória de grupo que usuários, desenvolvedores e gerentes irão travar suas comunicações, coordenar suas atividades e ter acesso ao conhecimento comum.

3.1 Representação da Informação

Podemos destacar o papel definitivo da metáfora de hipertexto como tecnologia para a organização e disseminação da memória de grupo. É através de interfaces hipertextuais que usuários e desenvolvedores irão construir uma interface de conhecimento comum, irão registrar suas idéias, estabelecer comunicação e registrar a evolução do produto gerado [ARAU97a].

A maioria dos ambientes e ferramentas estudadas armazenam em uma base hipertextual o material produzido durante o processo de construção (os componentes de software), como o ambiente OZWeb [OZWE96] e *Conversation Builder* [KAPL92].

Algumas propostas se preocupam em expandir esta base de artefatos com as contribuições informais. O ambiente ICARO [LICE96], por exemplo, faculta a geração automática de documentos HTML a partir da documentação de projeto gerada com o uso de uma ferramenta CASE. A partir desta documentação em HTML, o ambiente permite que discussões sejam desencadeadas e registradas. O ambiente *Beyond-Sniff* [BISC94] permite a associação de anotações geradas por seus usuários aos artefatos de software. O ambiente *Evolving Artifact* (EVA) [OSTW95] suporta a criação de um novo tipo de artefato de software que integra documentação baseada em hipertextos com protótipos. EVA provê um tipo de ligação executável que, quando selecionada, ocasiona a execução de um protótipo do sistema. Usando EVA, usuários e desenvolvedores podem criar uma documentação baseada em hipertexto descrevendo a aplicação e seus requisitos; desenvolvedores constroem protótipos que serão embutidos no contexto apropriado da documentação; e usuários podem ter acesso e interagir com os protótipos e registrar seus comentários e críticas ao hipertexto.

Outras abordagens se preocupam em armazenar apenas a interação informal. Neste caso, os produtos de software gerados são armazenados em outras bases, podendo estar associados aos elementos gerados durante a interação através de ligações externas. Esta abordagem é usada pela ferramenta Quorum [ARAU94], pelo ambiente ARCoPAS [CAVA94] e pela ferramenta SISCO [BELL95] [BELL96].

3.2 Modelos de Argumentação

Aliado aos recursos de organização e navegação oferecidos pela tecnologia de hipertextos, reconhece-se a necessidade de prover recursos para captura da memória coletiva. Esta captura tem sido comumente realizada através do registro de justificativas de projeto baseado em modelos de argumentação.

Este é o caso do ambiente CPCE (*Collaborative Process-Centered Environment*) [CPCE94], da ferramenta Quorum [ARAU94], do ambiente ARCoPAS [CAVA94], do ambiente para projeto participativo de Cavalcanti e Borges [CAVA96] e da ferramenta SISCO [BELL95]. No CPCE, que é um ambiente centrado em controle de processos, a encenação do modelo de processo se baseia num modelo de discussão que é uma extensão do modelo sugerido por Potts. O ambiente proposto por Cavalcanti e Borges utiliza a técnica do *Inquiry Cycle Model*. A ferramenta Quorum, o ambiente ARCoPAS e a ferramenta SISCO propõem o uso do modelo de argumentação IBIS.

Segundo Twidale, Rodden e Sommerville [TWID93], o registro de discussões e justificativas de projeto (*design rationale*) apresenta problemas no que diz respeito a dificuldades de aceitação por parte dos usuários, por os obrigarem a decidir previamente a natureza de cada contribuição que lançarem.

Os estudos realizados no decorrer do desenvolvimento da ferramenta DNP (*Designers' Notepad*) [TWID93], confirmam que há uma grande variação das atividades de projeto tanto entre usuários e pelos mesmos usuários ao longo do tempo e de acordo com as circunstâncias. Esta variação mostra a necessidade de dotar a ferramenta com flexibilidade para suportar esta variação. No caso do DNP, esta flexibilidade está em evitar a delegação de semântica a entidades e ligações entre elas, permitindo que sejam utilizadas livremente pelos usuários nos estágios iniciais do projeto, quando ainda não se pode exigir muita formalização de conceitos. Esta liberdade permite que a estrutura de conceitos apropriada surja durante o projeto após as entidades terem sido levantadas e criadas.

3.3 Formalização de Conceitos

Algumas propostas de ferramentas cooperativas atacam a questão do registro do conhecimento através da formalização de conceitos. Mais do que meros dicionários de dados, estas ferramentas auxiliam a coleta/geração e organização de conceitos e vão além, reconhecendo a necessidade de suporte à discussões como mecanismo para promover o refinamento destes conceitos.

A ferramenta *Conflict-Resolution Supporting Tool* [IOCH95][MARA96], por exemplo, tem o objetivo de auxiliar equipes de desenvolvimento a identificar conflitos de linguagem entre os membros do grupo, para evitar que erros de conceituação possam comprometer a qualidade do processo de desenvolvimento. Ao invés de se preocupar em suportar a resolução dos conflitos de linguagem, a ferramenta enfatiza o suporte à identificação destes conflitos através da análise dos diálogos e documentos que descrevem a aplicação ou o projeto do sistema.

O entendimento entre participantes do projeto e suporte ao trabalho criativo em conjunto é um dos objetivos da ferramenta DNP (*Designers' Notepad*) [TWID93]. Esta ferramenta procura oferecer mecanismos para suportar a liberdade de expressão em relação a conceitos (necessária principalmente nos estágios iniciais do projeto de software) e uma maior formalização destes conceitos conforme o projeto evolui. Trata-se, portanto de um processador de idéias, envolvendo a geração, organização e refinamento de conceitos e o suporte à argumentação em relação às definições.

3.4 Gerência de Configuração

A quase totalidade das abordagens para controle da configuração do software ou controle de versões tendem para uma estrutura onde são armazenadas as alterações consecutivas a um determinado componente de software, facultando aos participantes visualizar a versão de um determinado componente no tempo com maior flexibilidade.

Nesta categoria, estão as propostas da linguagem VTML (*Versioned Text Markup Language*) [VITA96], que pretende ser uma linguagem de marcação de textos para controle de versões de documentos; o ambiente EPOS (*Expert System for Program and System Development*) [EPOS96], onde uma versão não é considerada como um objeto explícito mas como o resultado de determinadas mudanças que foram aplicadas à base de dados; o ambiente PROSOFT [REIS96], onde uma versão é um objeto composto pelo objeto original e as modificações ocorridas; e o ambiente proposto por Magnusson e Guerraoui [MAGN96], onde o histórico de alterações de um programa é armazenado através de estruturas hierárquicas denominadas grafos de evolução, que descrevem as versões do sistema e os deltas relativos às alterações realizadas entre versões.

Algumas abordagens se preocupam com o registro dos motivos que levaram às alterações entre versões, como os ambientes Quorum [ARAU94] e ARCoPAS [CAVA94].

3.5 Armazenamento de Dados

Procuramos observar, na propostas de ambientes cooperativos de software, as soluções utilizadas para o armazenamento das informações em bases de dados. Surpreendentemente (ou não), a maioria das propostas estudadas sugerem o uso de bases de objetos como repositório de informações.

O ambiente SPADE [BAND93] e o ambiente CPCE [CPCE94] apostam em bancos de dados orientados a objetos comerciais como: O₂ e GemStone, respectivamente. Outros ambientes construíram seus próprios gerenciadores de objetos como o OZWeb [OZWE96], EPOS [EPOS96] e Coo (*Cooperation & Coordination*) [CANA94]. O ambiente *Conversation Builder* [KAPL92] usa um mecanismo para persistência de objetos mas, planejavam a evolução para o uso de bancos de dados orientados a objetos.

O ambiente proposto por Magnusson e Guerraoui [MAGN96] armazena todas as informações sobre a configuração do sistema em construção em arquivos Unix utilizando um formato especial (*Treefile*).

4. Abordagens de Suporte à Percepção

O conceito de percepção pode ser definido como a contextualização das atividades individuais através da compreensão das atividades realizadas por outras pessoas [BORG96]. Com o uso de mecanismos de percepção, é possível a um membro do grupo ter noção do contexto onde está inserido seu trabalho e o trabalho da equipe.

Os mecanismos de percepção são essenciais para o suporte de grupo, na medida em que transformam interações irregulares em interações consistentes e perceptíveis no decorrer do tempo. São mecanismos que possibilitam que os participantes da interação mantenham-se atualizados sobre eventos importantes, contribuindo para que suas atividades sejam realizadas de forma mais consciente e, conseqüentemente, mais eficazes.

No que diz respeito ao suporte à cooperação para grupos de desenvolvimento de software, é preciso prover recursos para que todos os membros tenham a noção do contexto de suas atividades dentro do contexto geral do processo, para que consigam perceber o andamento das

atividades sendo realizadas por outros membros e para compreender como os resultados gerados pelas atividades alheias podem ser conjugados aos seus para chegarem mais rapidamente ao resultado final.

A noção de presença de outros participantes é o tipo mais comum de informação oferecido para os membros do grupo. Este tipo de percepção está relacionado a mecanismos de notificação dos membros que estão ativos a cada momento. O conhecimento da presença de cada membro define os canais de comunicação disponíveis para cada interação.

Uma outra forma de percepção diz respeito à posição dos participantes dentro do trabalho ou atividade sendo realizada. Essa informação provê ao grupo o seu contexto de trabalho, mostrando quem realiza o que a cada momento.

O estado ou situação de execução de uma atividade também são informações úteis para prover percepção. Em alguns casos, pode ser necessário permitir a visualização da execução das atividades alheias.

4.1 Recursos de Interface

Para a contextualização de cada participante em relação aos outros membros da equipe, algumas ferramentas oferecem paradigmas de percepção como cursores personalizados, WYSIWIS e teleapontamento, conforme proposto, por exemplo, no ambiente CSD [CSD96].

4.2 Consulta e Navegação através da Memória de Grupo

Outras ferramentas se baseiam em consultas ou navegação pelo espaço de trabalho a procura de informações sobre a interação e as atividades realizadas por cada membro.

A ferramenta CHAT, provida pelo EPOS [EPOS96], permite que os usuários obtenham informações sobre espaços de trabalho, componentes e relações. Utilizando o navegador do espaço de trabalho, os usuários podem checar se alguém está conectado, quais componentes foram recuperados da base (*check out*), quando o espaço de trabalho foi criado, quem o criou e com que objetivos.

4.3 Anotações

O ambiente *Beyond-Sniff* [BISC94] baseia seu mecanismos de percepção nas anotações geradas pelo grupo e registradas em sua base. As consultas podem ser feitas a partir dos artefatos de software armazenados ou podem ser formuladas consultas baseadas nos atributos das anotações (tipo, data, artefato relacionado etc). Usuários podem ser notificados quanto à ocorrência de determinadas anotações ou mesmo serem notificados de quando uma determinada anotação foi lida. Anotações podem, ainda, ser endereçadas a usuários específicos, podem ser inseridas manualmente ou podem ser geradas automaticamente por ferramentas.

4.4 Notificações

Mecanismos de notificação são importantes para a percepção, permitindo que um membro do grupo tome conhecimento de eventos ocorridos durante a interação.

O ambiente *Conversation Builder* [KAPL92] tem como objetivo principal ser um ambiente "ativo". Isto significa que o ambiente tem conhecimento sobre as ações realizadas pelos usuários do sistema e pode usar este conhecimento para prover ajuda sensível ao contexto, guiar usuários pelo processo de desenvolvimento e manter os usuários a par das ações de outros. *Conversation Builder* auxilia o usuário para que tenha noção das tarefas em que está

engajado, as relações entre estas tarefas, as ações que pode realizar em cada uma, a estrutura da base de componentes e as ações relevantes de outros membros.

5. Abordagens de Suporte à Comunicação

Quando pensamos em possibilidades de comunicação, a palavra-chave torna-se ligação. Não se pode esperar cooperação entre partes sem que haja uma ligação entre elas. A comunicação entre os membros de uma equipe de desenvolvimento reside na existência e potencialidade das ligações entre eles. Estas ligações podem ser definidas como técnicas e/ou canais de comunicação que permitem aos usuários e desenvolvedores trocar informações [KEIL95]. Esses canais de comunicação compreendem, por exemplo, mecanismos de troca de mensagens, reuniões e fóruns de discussão.

Além desta comunicação direta, é possível estabelecer um canal de comunicação indireto através da memória de grupo, onde a construção progressiva e o compartilhamento do conhecimento comum podem ser considerados como interfaces de comunicação em potencial. Outra forma de comunicação indireta é suportada por mecanismos de percepção, uma vez que a contextualização de um participante dentro do trabalho do grupo também estabelece uma ligação deste com os demais membros.

A coordenação também contribui e influencia a comunicação, na medida em que se pode definir protocolos de interação do grupo, com o objetivo de tornar a comunicação mais eficiente.

5.1 Modos de Interação

Quanto aos modos de interação, várias das ferramentas e ambientes estudados não discutem este aspecto ou mesmo mencionam como pretendem promover a interação entre seus participantes. Pelo o que nos foi possível analisar, privilegia-se a interação remota [CSD96][ARAU94] e assíncrona [ARAU94][CPCE94][CAVA94][BISC94][BELL95], talvez por ser o modo de interação que suscite maiores novidades de pesquisa e flexibilize as oportunidades de interação entre os membros da equipe de projeto.

Por outro lado, muitas propostas de ambientes de suporte ao projeto cooperativo de software se baseiam na definição de uma estrutura de integração de ferramentas. Assim, o modo de interação entre seus usuários estaria vinculado ao modo de interação imposto por cada ferramenta.

Nota-se claramente, ainda, uma tendência a flexibilizar o modo de interação de acordo com a natureza da atividade sendo realizada, conforme apontado por Cavalcanti e Borges em sua proposta de ambiente para projeto participativo de software [CAVA96], no ambiente *Flecse (Flexible Environment for Collaborative Software)* [DEWA93] e no ambiente *OZWeb* [OZWE96]. Isto significa que os ambientes procuram oferecer uma infraestrutura de interação onde passar de interações síncronas para interações assíncronas, bem como ter acesso ao contexto de trabalho de forma remota ou localmente, seja uma transição possível e facilitada.

5.2 Espaços de Trabalho Compartilhados

Segundo Twidale, Rodden e Sommerville [TWID93], espaços de trabalho compartilhados são os recursos mais utilizados para suporte ao projeto cooperativo de software. Nesta abordagem, aos desenvolvedores é oferecido uma superfície compartilhada, onde podem expressar suas idéias e/ou construir seus produtos.

A ferramenta DNP (*Designers' Notepad*) [TWID93] oferece um espaço gráfico compartilhado como meio para representação de idéias durante conversas e discussões realizadas durante as atividades de projeto. Da mesma forma, o ambiente para suporte ao projeto participativo de software de Cavalcanti e Borges [CAVA96] oferece ferramentas para co-edição de cenários e modelos de domínio da aplicação em construção. O ambiente Flecse [DEWA93] oferece uma ferramenta para edição cooperativa de código. O ambiente CSD (*Cooperative Software Development*) [CSD96] segue a mesma filosofia, oferecendo um espaço para edição de componentes de software e realização de conferências, baseado em várias mídias (texto, gráfico e áudio). Este tipo de espaço compartilhado, com uso de recursos gráficos, é comumente utilizado para interações síncronas.

5.3 Comunicação Indireta através da Memória de Grupo

Espaços de comunicação indireta podem compreender bases de informações compartilhadas com o registro das interações realizadas por cada participante. Esta forma de compartilhamento de trabalho é comumente utilizada em interações assíncronas conforme proposto nos ambientes Quorum [ARAU94], SISCO [BELL95], EPOS [EPOS96], ARCoPAS [CAVA94] e *Beyond-Sniff* [BISC94].

5.4 Sistemas de Mensagens

Sistemas de mensagens são recursos amplamente utilizados para o suporte à comunicação em ambientes cooperativos, principalmente para estabelecer canais de comunicação informal e paralela à atividade em andamento.

Este é o caso dos ambientes ICARO [LICE96] e RASP (*Regatta Automated Software Process*) [SWEN93] que utilizam o sistema de mensagens para a delegação e negociação de realização de tarefas entre os gerentes de projeto e os outros membros da equipe.

O ambiente EPOS [EPOS96], por sua vez, possui uma ferramenta - CHAT (*Conflict Handling Toolkit*) - baseada num sistema de mensagens para apoio à resolução de conflitos ocorridos durante a execução das atividades do processo. As mensagens podem ser enviadas pelos próprios usuários ou podem ter sido geradas pelo sistema na ocorrência de conflitos.

Já a ferramenta Quorum [ARAU94] utiliza um sistema de mensagens para promover interações marginais aos processos de tomada de decisão, como: solicitação de informações, lançamento de comentários, solicitação de participação de membros do grupo, comunicação de eventos, ocorrências e detalhes, envio de lembretes etc.

6. Abordagens de Suporte à Coordenação

Construir software cooperativamente exige um gerenciamento e acompanhamento constante das atividades sendo realizadas pelo grupo como um todo e daquelas realizadas individualmente por cada participante. É preciso, portanto, prover suporte para a execução das tarefas do grupo e das tarefas individuais de cada membro através de mecanismos de definição, visualização e acompanhamento do encaminhamento do processo.

A palavra-chave relacionado à coordenação, portanto, refere-se a acompanhamento. Especificar como a interação se dará, definir regras e limites, estipular responsabilidades e controlar e acompanhar a execução de tarefas são questões que precisam ser apoiadas durante o processo. A memória de grupo é responsável por manter essas informações disponíveis.

A comunicação e a percepção podem ser importantes para o acompanhamento de tarefas ao contribuírem para que as questões de coordenação sejam resolvidas e decisões neste escopo

sejam tomadas. Desta forma, os membros da equipe estarão envolvidos com o próprio acompanhamento do processo de trabalho, tornando a coordenação mais eficiente.

6.1 Modelagem de Processos de Software

Esta é uma questão de grande ênfase em relação a ambientes de suporte ao projeto cooperativo de software. Muitas são as propostas para prover a modelagem e encenação de processos de software levando em consideração o aspecto cooperativo do desenvolvimento.

Não sendo possível garantir a existência de um modelo de processo universal para suporte ao desenvolvimento de software, há uma evidente inclinação a oferecer a possibilidade de customização de processos para cada contexto de projeto através de linguagens para modelagem e representação de processos.

Alguns ambientes oferecem uma linguagem de modelagem de processos exclusiva, como o ambiente SPADE (*Software Process Analysis, Design and Enactment*) [BAND93], que define a linguagem SLANG (*Spade Language*), baseada em Redes de Petri de alto nível. O ambiente EPOS [EPOS96] segue a mesma idéia através da linguagem SPELL (*Software Process Evolutionary Language*), baseada num modelo de objetos contendo as classes que descrevem as entidades envolvidas na modelagem do processo. Outros, como o ambiente OZWeb [OZWE96], permitem o uso de qualquer paradigma para modelagem de processos.

Em algumas propostas, como em EPOS [EPOS96] e *Conversation Builder* [KAPL92], há também a preocupação com a modelagem de meta-processos, a definição de metodologias de modelagem e a criação de bibliotecas de modelos de processo reutilizáveis.

A definição de meta-processos e sua customização também são as preocupações do ambiente CPCE [CPCE94]. Sua proposta enfoca o processo através de um modelo de discussão onde questões, posições e argumentos são apresentados em relação a artefatos sendo manipulados pelos participantes ou em relação aos passos definidos para a execução da atividade em andamento. Portanto, processo e meta-processo se confundem numa mesma estrutura de argumentação. Processos são especializados (instanciados) a partir do modelo geral de processos e podem ser customizados para atender a requisitos específicos de determinados contextos de colaboração. Alterações no processo são realizadas através do lançamento de questões quanto ao meta-processo.

Uma filosofia similar segue o ambiente *Conversation Builder* [KAPL92]. Neste, as atividades realizadas são consideradas como "conversas", seguindo a filosofia de Winograd e Flores de "conversa para ação". Um protocolo de conversação provê um conjunto de regras que governam o comportamento das conversas instanciadas sob aquele protocolo, tais como as atividades que podem ser realizadas, por quem serão realizadas, sob quais circunstâncias e quais seus efeitos.

Numa tentativa de flexibilizar ainda mais a modelagem de processos, o ambiente RASP [SWEN93] se baseia na automação e reengenharia de processos de software, permitindo que membros de equipes tomem parte do planejamento e redefinição do processo. RASP evita o problema da complexidade na modelagem de processos por não exigir que os modelos estejam completos para serem encenados.

6.2 Encenação de Processos de Software

As propostas para o controle da execução dos processos de software focalizam principalmente a questão da evolução dinâmica do processo (alterações no processo no decorrer de sua encenação).

No ambiente SPADE [BAND93], por exemplo, as atividades podem ser instanciadas dinamicamente durante a encenação do processo. Analogamente, no ambiente OZWeb [OZWE96], os modelos de processo são encenados em uma máquina de *workflow* oferecida pelo ambiente que permite a associação dinâmica de usuários a atividades, permitindo, inclusive, a delegação de tarefas durante a execução do processo.

Outras questões são abordadas com o objetivo de livrar a encenação de processos de um controle rígido e restritivo. O ambiente OZWeb, por exemplo, tem a preocupação em tolerar inconsistências na encenação, quer que seja possível operar com informações incompletas e permitir fragmentações no processo. O projeto do ambiente OZWeb apresenta, ainda, um modelo para ambientes descentralizados centrados em processos. A tentativa é a de explorar a modelagem e a encenação de colaboração entre grupos por processos independentes, autônomos e possivelmente pré-existent, diferentemente do suporte intra-grupos comumente oferecido por ambientes.

A mesma preocupação com execução flexível pauta os objetivos do ambiente Coo. Uma característica central deste ambiente surge do fato de que existem limites para uso de abordagens puramente baseadas no conhecimento prévio de todas as possibilidades de interação no decorrer do processo. A proposta do ambiente Coo está em basear a encenação num modelo de transação seguro que, no momento em que o conhecimento sobre o processo seja insuficiente, possa definir um modo de sincronização padrão para sua encenação [GODA94].

6.3 Modelagem de Papéis

Em todos os ambientes e ferramentas pesquisados, há a preocupação na modelagem de seus usuários e delegação de responsabilidades aos diferentes tipos de usuários modelados.

Alguns ambientes/ferramentas se preocupam em classificar explicitamente os usuários em grupos através de suas características, responsabilidades e papéis em relação ao processo [OZWE96] [ARAU94] [CAVA94] [BELL95]. Outros ambientes, como o EPOS [EPOS96] e *Conversation Builder* [KAPL92], também associam usuários à execução de atividades do processo mas não parecem se preocupar com a classificação e modelagem de usuários em papéis.

6.4 Controle de Concorrência

O modelo tradicional de controle de concorrência em ambientes para múltiplos usuários baseia-se nos conceitos de transação e chaveamento (bloqueio). O problema desta abordagem, quando utilizada em ambientes para suporte a colaboração, está em colocar uma visão particular da cooperação. Esta visão de controle rígido é muitas vezes inaceitável dada a riqueza de padrões de interação em desenvolvimento cooperativo de software. Em muitos momentos do processo de desenvolvimento, por exemplo, um usuário precisa ter conhecimento de que um dado componente está sendo utilizado por outro membro da equipe e pode ainda querer ver o que este membro executa sobre o componente, como se estivesse "olhando por sobre seus ombros".

Com o intuito de preservar os aspectos de percepção das interações, um requisito importante para ferramentas cooperativas é o de que o uso por múltiplos usuários deve ser explícito ao invés de prevenido ou "disfarçado". Isto requer um modelo de controle de acesso a informações e ferramentas diferente dos modelos tradicionais. Basicamente, as propostas procuram oferecer um controle mais otimista aos objetos armazenados e a possibilidade de solução de conflitos através de negociações entre os usuários.

Pensando nisso, o ambiente EPOS [EPOS96] busca oferecer suporte para planejamento, levantamento e tratamento de conflitos de concorrência, ao invés de evitar que estes ocorram e coloquem restrições à forma de trabalho em grupo.

O ambiente ARCoPAS [CAVA94] usa uma abordagem otimista, possibilitando a existência de cópias individuais da base de informações compartilhadas, onde os membros da equipe de projeto podem realizar alterações e incluir seus comentários pessoais sem alterar a base comum.

O *Beyond-Sniff* [BISC94] oferece controle otimista e pessimista para acesso aos artefatos armazenados. O controle pessimista significa que o acesso é controlado através de mecanismos de chaveamento. O controle otimista permite que os desenvolvedores trabalhem autonomamente em cópias do projeto que serão fundidas posteriormente, gerando uma nova versão do mesmo.

O ambiente proposto por Magnusson e Guerraoui [MAGN96] também aposta no controle otimista de versões. Contudo, em alguns casos, podem ser definidas linhas "protegidas", ou seja, versões do software que só podem ser editadas por um grupo de usuários pré-definido.

No ambiente Coe [CANA94], quando ocorrem conflitos de acesso a objetos, são iniciadas discussões entre os agentes humanos associados ao processo em conflito. Todos os agentes direta ou indiretamente envolvidos com esta tarefa são potencialmente indicados para a negociação.

7. Síntese da Aplicação do Esquema de Classificação

Usando o esquema proposto, foi possível analisar as abordagens utilizadas pelas propostas para o suporte à colaboração de desenvolvimento de software. Um resumo desta análise é apresentado na Tabela 2.

Podemos observar a existência de dois ramos de pesquisa na área de desenvolvimento cooperativo de software. Um deles diz respeito à construção de ferramentas isoladas, voltadas para o suporte a aspectos específicos ou a fases isoladas do processo de construção de software. O outro ramo focaliza a construção de ambientes cooperativos, com ênfase na definição de um núcleo principal de controle de atividades através de abordagens de modelagem de processos, onde ferramentas podem ser acopladas e seu uso gerenciado. Nos dois ramos observa-se que a área apresenta ainda poucos resultados quanto ao uso dessas ferramentas e ambientes.

O primeiro ramo de pesquisa tende a ter uma maior preocupação com aspectos de interface, comunicação, percepção e coleta da memória de grupo; enquanto que o segundo se concentra em aspectos de comunicação, registro da memória de grupo e coordenação de tarefas.

As ferramentas/ambientes seguem abordagens bem similares para o gerenciamento da memória de grupo. Estas abordagens se concentram nas questões que envolvem o registro de informações em bases hipertextuais. A maioria também reconhece a necessidade de registro da memória informal, usando para isto modelos de argumentação. Mas, ainda há o que se discutir em relação aos modelos de argumentação utilizados até então como forma de captura do conhecimento informal.

Nota-se claramente que a questão da percepção tem sido pouco estudada neste contexto. As soluções se concentram em mecanismos de consulta e navegação pela base de informações armazenadas na memória de grupo e poucas propostas relativas a recursos de interface são apresentadas. Estas metáforas seguem as propostas sugeridas na área de CSCW em escopo

geral. Pesquisas quanto a necessidade ou não de definição de metáforas específicas para interação entre grupos de desenvolvimento de software precisam ser mais elaboradas.

Ferramenta/Ambiente	Memória de Grupo					Percepção				Comunicação				Coordenação				Suporte ao Processo
	1	2	3	4	5	1	2	3	4	1	2	3	4	1	2	3	4	
<i>Beyond-Sniff</i>																		Codificação
<i>Magnusson e Guerraoui</i>																		Projeto e Codificação
<i>Flece</i>																		Codificação/Inspeção/ Teste
<i>CSI</i>																		Inspeção de software
<i>ARCoPAS</i>																		Recuperação de Projeto
<i>Cavalcanti e Borges</i>																		Levantamento de Requisitos
<i>Conflict-Resolution Tool</i>																		Levantamento de Requisitos
<i>Designers' Notepad</i>																		Levantamento de Requisitos
<i>ICARO</i>																		Suporte à discussões
<i>Quorum</i>																		Suporte à discussões
<i>SISCO</i>																		Suporte à reuniões
<i>RASP</i>																		Coordenação de atividades
<i>Evolving Artifact (EVA)</i>																		Suporte global ao processo
<i>OZWeb</i>																		Suporte global ao processo
<i>CSD</i>																		Suporte global ao processo
<i>EPOS</i>																		Suporte global ao processo
<i>Conversation Builder</i>																		Suporte global ao processo
<i>CPCE</i>																		Suporte global ao processo
<i>PROSOFT</i>																		Suporte global ao processo
<i>SPADE</i>																		Suporte global ao processo
<i>Coo</i>																		Suporte global ao processo

Tabela 2 - Resumo das propostas analisadas

Quanto ao aspecto de canais de comunicação, percebemos que as conferências são o principal recurso utilizado pelas ferramentas/ambientes. Consideramos como conferência as interações que envolvem a comunicação remota ou local, síncrona ou assíncrona entre indivíduos acerca de um foco comum, seja este um espaço compartilhado de trabalho ou uma base de discussões.

O suporte à coordenação está maciçamente concentrado na modelagem e encenação de processos e modelagem de usuários. Contudo, evidencia-se a necessidade de atender a questões específicas no que se refere à dinamicidade do processo. As abordagens para modelagem e encenação parecem seguir as propostas tradicionais de linguagens de modelagem, mas com recursos específicos voltados para o suporte à interação de grupos e para as atividades de desenvolvimento de software.

Não cabe neste trabalho a realização de críticas e comparações entre as ferramentas/ambientes analisados. Isto porque o estudo está baseado em propostas publicadas na literatura, que não são necessariamente completas ou não descrevem as propostas em todos os seus aspectos. Além disso, nosso estudo está restrito às propostas que se consideram claramente voltadas

para o apoio ao desenvolvimento colaborativo de software. Ferramentas ou ambientes genéricos, embora possam ser adaptados para utilização em contextos de desenvolvimento de software, não fizeram parte de nosso foco de pesquisa neste trabalho.

8. Conclusão

Neste trabalho, apresentamos uma classificação das questões que consideramos como básicas para o apoio ao projeto cooperativo de software. Partindo da visão do processo de desenvolvimento como um processo contínuo de resolução de conflitos e convergência de pontos de vista, quatro aspectos foram estabelecidos como necessários ao suporte a este processo de busca de entendimento: a oferta de canais de comunicação, a abordagem de registro da memória de grupo, os mecanismos de contextualização e percepção e os protocolos e padrões de coordenação.

Uma vez que todas as questões que pautam a pesquisa em engenharia de software podem se beneficiar do suporte à cooperação, o esquema apresentado neste trabalho contribui principalmente como elemento de organização e compreensão dos aspectos envolvidos com este suporte.

Uma outra contribuição do nosso esquema conceitual é a possibilidade de utilizá-lo como um *framework* genérico para a especificação e modelagem de aplicações voltadas para o apoio ao desenvolvimento colaborativo de software. Sua utilização como *framework* para modelagem está sendo utilizada em [DIAS97].

Conforme apresentado, o instrumento de classificação proposto é útil para a análise dos esforços de pesquisa nesta área, permitindo avaliar em quais aspectos a pesquisa tem se concentrado e quais aspectos têm sido negligenciados, e, desta forma, orientando melhor o foco de trabalhos nesta área.

Apesar do apelo à necessidade de suporte à cooperação em processos de software, dificilmente são encontrados ferramentas/ambientes comerciais com características especificamente voltadas para este suporte. Mais do que isto, das propostas analisadas, poucos são os resultados de sua utilização em estudos de caso reais e em projetos de longa duração. Desta forma, ainda não se pode afirmar se estas abordagens de suporte à cooperação apoiam eficientemente o desenvolvimento de software. Isso nos faz acreditar no potencial acadêmico das pesquisas nesta área, uma vez que os aspectos sociais e cooperativos em desenvolvimento de software ainda apresentam questões a serem descobertas e outras a serem resolvidas.

Esta área apresenta ainda um grande potencial econômico, pois eficiência e produtividade podem ser alcançadas com uma maior socialização do processo de trabalho. O apoio cooperativo traz, então, novas perspectivas ao suporte à engenharia de software, na busca de um processo de trabalho menos rígido, mais ágil e produtivo.

9. Referências

- [ARAU94a] Araujo, R.M. (1994) Quorum - um SSDG para o desenvolvimento de software Tese de Mestrado COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação Agosto.
- [ARAU94b] Araujo, R.M., Borges, M.R.S. (1994) Quorum - um SSDG para o desenvolvimento de software VIII Simpósio Brasileiro de Engenharia de Software Curitiba Outubro.
- [ARAU97a] Araujo, R.M., Xexéo, G.B. (1997) Placing participatory software design in light of cognitive ecology World Multiconference on Systemics, Cybernetics and Informatics Caracas Venezuela.

- [BAND93] Bandinelli, S. Braga, M. Fuggetta, A. Lavazza, L. (1993) Cooperation in the SPADE environment: a case study Proceedings of Workshop on Computer Supported Cooperative Work, Petri Nets and Related Formalisms Chicago Junho.
- [BELL95] Bellasai, G. Borges, M.R. Fuller, D.A. Pino, J.A. (1995) SISCO: a tool to improve meetings productivity First CYTED-RITOS International Workshop on Groupware Lisboa Portugal.
- [BELL96] Bellasai, G. Borges, M.R. Fuller, D.A. Pino, J.A. (1996) An IBIS-based model to support group discussions in Proceedings of the 1996 IFIP International Workshop on the Office of the Future, Tucson, Arizona, Abril págs. 39-53.
- [BISC94] Bischofberger, W.R. Kofler, T. Mätzel, K. Schäffer, B. (1994) Computer-supported cooperative software engineering with Beyond-Sniff UBILAB Technical Report 94.9.1 Union Bank of Switzerland Zurich.
- [CAVA94a] Cavalcanti, M.C.R. (1994) ARCoPAS: um ambiente para recuperação cooperativa do projeto arquitetônico de sistemas Tese de Mestrado COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação Setembro.
- [CAVA94b] Cavalcanti, M.C.R. Borges, M.R.S. (1994) ARCoPAS: um ambiente para recuperação cooperativa do projeto arquitetônico de sistemas VIII Simpósio Brasileiro de Engenharia de Software Curitiba Outubro.
- [CAVA96] Cavalcanti, M.C.R. Borges, M.R.S. (1996) Participatory software design - specifying a methodology and a support environment unpublished paper Object Technology Laboratory School of Engineering Santa Clara University Fevereiro.
- [DAFT86] Daft, R.L. Lengel, R.H. (1986) Organizational information requirements, media richness and structural design Organization Science, 32/5: 554-571.
- [DEWA93] Dewan, P. Riedl, J. (1993) Toward computer-supported concurrent software engineering Computer January págs. 17-27.
- [DIAS97] Dias, M.S. Borges, M.R.S. (1997) Suporte ao desenvolvimento cooperativo de projeto de software orientado a objetos Workshop de Teses em Engenharia de Software Fortaleza Outubro.
- [GIBB89] Gibbs, S. (1989) CSCW and Software Engineering, Object-Oriented Development Tschritzs, D. (ed), Centre Universitaire d'Informatique Université de Genève Julho págs. 31-40.
- [IOCH95] Iochpe, C. (1995) Improving requirements analysis through cscw Primeiro Seminário Internacional CYTED-RITOS sobre Sistemas Cooperativos Lisboa Portugal Setembro págs.29-37.
- [KAPL92] Kaplan, S. M. Tolone, W. J. Carroll, A. M. Bogia, D. P. Bignoli, C. (1992) Supporting collaborative software development with ConversationBuilder V Symposium on Software Development Environments págs. 11-20.
- [KEIL95] Keil, M. Carmel, E. (1995) Customer-developer links in software development CACM vol. 38 n. 5 págs.33-44.
- [LICE96] Licea, G. Favela, J. (1996) ICARO: a web-based environment for collaborative software development Segundo Seminário Internacional CYTED-RITOS sobre Sistemas Cooperativos Setembro Puerto Varas Chile págs.23-30.
- [MAGN96] Magnusson, B. Guerraoui, R. (1996) Support for collaborative object-oriented development International Symposium on Parallel and Distributed Computing Systems (PDCS'96), Dijon, September 1996.
- [MASH93] Mashayekhi, V. Drake, J. M. Tsai, W. Riedl, J. (1993) Distributed, collaborative software inspection IEEE Software September págs. 66-75.
- [PATN95] Patnayakuni, R. Patnayakuni, N. (1995) A socio-technical approach to CASE and the software development process Proceedings of the 1st Americas Conference on Information Systems Pittsburgh Pennsylvania USA Agosto págs. 6-8.

- [SWEN93] Swenson, K. D. (1993) Regatta project: a tool for business process reengineering Proceedings of the First International Conference in Technologies and Theories for Human Cooperation, Collaboration and Coordination - Applica'93 Março..

10. Webliografia

- [ARAU97b] Araujo, R.M. Dias, M.S. Borges, M.R.S. (1997) A Framework for the Classification of Computer Supported Collaborative Design Concepts <http://www.cos.ufrj.br/~renata/Memoria/framework.htm> acesso em 29/07/97.
- [BORG96] Borges, M.R.S. (1996) Awareness mechanisms for SISCO <http://www.nce.ufrj.br/~mborges/sisco3/aware/sumario2.htm> acesso em 03/05/97.
- [CANA94] Canals, G. Charoy, F. Godart, C. Molli, P. (1994) P-Root & COO: building a cooperative software development environment <http://gille.loria.fr:7000/see95/see95.html> atualização em 31/08/94 acesso em 12/12/96.
- [CONK96] Conklin, J. (1996) Designing organizational memory: preserving intellectual assets in a knowledge economy Corporate Memory Systems, Inc. <http://www.zilker.net/business/info/pubs/desom/> Setembro.
- [CPCE94] (1994) A collaborative process-centered environment kernel <http://www.loria.fr/~jloncham/caise.ps.gz> acesso em 13/12/96 publicado em CAISE'94.
- [CSD96] (1996) CSD Home Page <http://www.fgd.fhg.de/~marcos/csd.html> acesso em 16/11/96.
- [DNP96] (1996) The Designers' NotePad <http://www.comp.lancs.ac.uk/computing/users/mbt/dnp/dnp.html> acesso em 22/12/96
- [EPOS96] (1996) EPOS group home page <http://www.idt.unit.no/~epos/> acesso em 10/12/96.
- [GODA94] Godart, C. Canals, G. Charoy, F. Molli, P. (1994) An introduction to cooperative software development in COO International Conference on Systems Integration ICIS'94 também disponível em <http://gille.loria.fr:7000/icsi94/ICSI94/ICSI94.html> acesso em 13/12/96.
- [MARA96] Maranhão, A. S. Iochpe, C. (1996) Proposta de identificador de palavras-chave para ferramenta de suporte à resolução de conflitos de linguagem I Semana Acadêmica do CPGCC/UFRGS - Instituto de Informática Setembro <http://www.inf.ufrgs.br/~deamar/seminar.htm> acesso em 24/12/96.
- [OSTW95] Ostwald, J. (1995) Supporting collaborative design with representations for mutual understanding CHI'95 Proceedings Doctoral Consortium disponível em http://www.acm.org/sigchi/chi95/Electronic/documnts/doctoral/jod_bdy.html acesso em setembro/1996.
- [OZWE96] (1996) OZWeb project summary <http://www.psl.cs.columbia.edu/edcs/> acesso em 19/11/96.
- [TWID93] Twidale, M. Rodden, T. Sommerville, I. (1993) The designers' notepad: supporting and understanding cooperative design http://www.comp.lancs.ac.uk/computing/users/mbt/dnp/dnp_ecscw93.ps acesso em 22/12/96.
- [VITA96] Vitali, F. Durand, D.G. (1996) Using versioning to support collaboration on the WWW <http://cs-pub.bu.edu/students/grads/dgd/version.html> acesso em 06/01/97.