

6 Conclusão

Aplicações que estão disponíveis de forma monolítica são pouco flexíveis, visto que há muitas inter-relações e interdependências amarradas em seu código fonte. Isto dificulta a adaptação do sistema às preferências e necessidades dos usuários e à sua evolução. Um pacote de aplicações (*application suite*) oferece um conjunto de programas, normalmente de um mesmo fabricante, com algum nível de integração e uma aparência padronizada. Alguns exemplos de pacotes de aplicações são o Microsoft Office³⁰ e o Netscape Communicator suite³¹. Porém, nos pacotes não há muita flexibilidade (nem sempre o usuário pode decidir quais partes do pacote deseja instalar) e não é possível integrar ferramentas externas. Além disto, nestas aplicações não há necessariamente reuso e modularização do código. A versão 2.0 do AulaNet apresenta-se como um pacote de aplicações, oferecendo um conjunto de serviços relativamente independentes. Entretanto, seu código fonte é fortemente integrado.

A utilização de técnicas de desenvolvimento baseado em componentes é uma forma de desenvolver groupware extensível. Estas técnicas visam desenvolver sistemas modulares compostos de componentes de software, adaptáveis e combináveis na medida da necessidade, tendo em mente futuras manutenções. A utilização de componentes favorece a utilização de código já testado em outras situações e a captura e o encapsulamento do conhecimento e da experiência dos especialistas do domínio.

Greenberg [2006] argumenta que a utilização de toolkits com componentes que encapsulam as complexidades do desenvolvimento de groupware é uma maneira de impulsionar o desenvolvimento desta tecnologia, que ainda não atingiu seu potencial de disseminação e utilização, principalmente nas empresas. De acordo com Greenberg, uma parcela muito pequena dos groupwares

³⁰ <http://office.microsoft.com>

³¹ <http://browser.netscape.com>

desenvolvidos tornou-se produtos comerciais realmente utilizados, apesar de sistemas para grupo terem sido previstos por Douglas Engelbart em 1968 e a área de CSCW ter sido estabelecida ao final da década de 80.

Ao analisar os motivos pelos quais a tecnologia de groupware ainda não deslanchou, Greenberg [2006] argumenta que diferentemente das aplicações desktop de automação de escritório e do hipertexto, que também foram previstas por Engelbart, o desenvolvimento de groupware não dispõe de um ferramental que simplifique a construção, aumente a produtividade e possibilite que programadores não tão experientes prototipem aplicações para grupos. As aplicações desktop e o hipertexto contam com toolkits e ambientes de desenvolvimento que estabelecem uma linguagem própria, utilizada pelos programadores para criar aplicações agrupando e interligando componentes. Com este ferramental, o foco do desenvolvimento é deslocado da criação de algoritmos referentes a problemas de baixo nível para a investigação da interação com o usuário e para o suporte computacional ao domínio em questão [Myers, 1995]. Tendo este ferramental disponível, uma grande quantidade de aplicações é criada, algumas comercialmente, outras por diversão e outras, por estudantes, como exercícios em seus cursos. Isto favorece a diversidade e a criatividade no desenvolvimento.

Ainda de acordo com o mesmo autor, atualmente o desenvolvimento de groupware ainda necessita de programadores altamente capacitados, aptos a lidar com protocolos de rede, soquetes, captura, apresentação, compressão e descompressão de áudio e vídeo, transações distribuídas, sincronização, concorrência de acesso, gerenciamento de sessão, entre outros. Com isto, mesmo questões bem conhecidas e investigadas pela comunidade de CSCW são deixadas de lado, pois o foco do desenvolvimento permanece no baixo nível e não há reuso do suporte computacional à colaboração. Nos sistemas colaborativos desenvolvidos desta maneira, o refinamento iterativo e a prototipação de novas idéias normalmente são dificultados, e o código fica fortemente acoplado. As ferramentas desenvolvidas acabam sendo pouco aderentes às reais necessidades de interação, que para serem descobertas requerem muita experimentação, investigação e prototipação. Greenberg [2006] posiciona o desenvolvimento de groupware no modelo BRETAM [Gaines, 1999], apresentado na Figura 6.1.

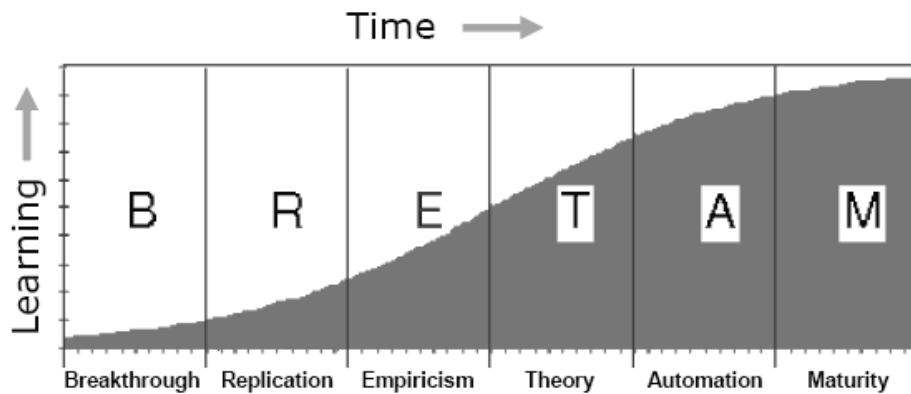


Figura 6.1. Modelo BRETAM para o desenvolvimento de uma tecnologia [Gaines, 1999]

O modelo BRETAM descreve como uma tecnologia evolui ao longo do tempo. A tecnologia inicia-se a partir de uma idéia criativa e uma visão de futuro (fase de *Breakthrough*). As idéias de groupware originaram-se da visão de Engelbart em 1968 e dos artigos seminais da década de 80. A fase de *Replication* ocorre quando as pessoas imitam as idéias umas das outras, reimplementando-as, alterando-as e inovando. Na replicação, estabelece-se uma comunidade de pesquisa, que adquire um corpo de conhecimento sobre os fatores centrais da tecnologia e sua relação com os seres humanos. O processo de construção das ferramentas ainda não é bem estabelecido e há muitas incertezas e retrabalho. De acordo com Greenberg [2006], groupware ainda está nesta etapa. Na fase de *Empiricism*, a tecnologia se torna bem disseminada e é adquirida uma larga experiência na aplicação da tecnologia a situações práticas e reais. Com base na experiência adquirida, são formuladas regras empíricas, que são descritas textualmente (boas práticas, padrões e guidelines) ou implementadas em ferramentas que as encapsulem e instrumentem o desenvolvedor. Ao ser adquirida mais experiência, teorias são desenvolvidas e validadas (fase de *Theory*) e, posteriormente automatizadas (fase de *Automation*). A fase de *Maturity* é atingida quando as tecnologias e teorias são utilizadas rotineiramente de forma transparente e sem questionamentos [Gaines, 1999].

A abordagem proposta nesta tese visa instrumentar o desenvolvedor de groupware com componentes concebidos com base no modelo 3C de colaboração. O desenvolvedor parte do levantamento de requisitos e da análise do domínio e seleciona os componentes necessários para oferecer suporte computacional à dinâmica estabelecida para a colaboração. Os conceitos abordados no modelo 3C

são usados para guiar a especificação do groupware e provêm uma linguagem comum para representar e descrever aspectos da colaboração.

Os conceitos da modelagem do domínio, que neste caso são referentes à colaboração, permeiam as diversas atividades e artefatos do desenvolvimento. A modelagem realizada é mapeada à implementação, suavizando a transição entre as atividades do desenvolvimento, o que favorece um ciclo iterativo e a manutenção do groupware. O modelo também propicia que no processo da engenharia sejam isolados os problemas referentes a cada C de modo a abordar cada elemento separadamente ao refinar o suporte à colaboração.

Greenberg [2006] propõe características, listadas na Tabela 6.1, que bons toolkits devem apresentar para instrumentar adequadamente o desenvolvedor de groupware e auxiliar a tecnologia a sair da fase de replicação do modelo BRETAM. A abordagem e a arquitetura propostas nesta tese atendem a estas características.

Num	Característica
1	Estar embutido em uma plataforma e utilizar linguagem familiar e de uso comum, de modo a facilitar a adoção solução e aproveitar conhecimentos e habilidades
2	Encapsular a complexidade de baixo nível inerente ao desenvolvimento de groupware, como transmissão de dados, compartilhamento, concorrência e gerenciamento de sessões
3	Minimizar a necessidade de tarefas rotineiras não essenciais
4	Encapsular projetos e conceitos bem sucedidos e conhecidos pela comunidade de pesquisa em módulos que possam ser incorporados ao software com pouco esforço de programação
5	Apresentar uma API concisa que encoraja as pessoas a pensar sobre groupware
6	Tornar as coisas simples alcançáveis com poucas linhas de código e tornar as complexas possíveis de serem feitas

Tabela 6.1. Características de bons toolkits [Greenberg, 2006]

A arquitetura e os componentes foram desenvolvidos na linguagem Java e utilizam uma extensão do modelo de componentes JavaBeans, que é largamente difundido e utilizado na literatura e comercialmente (característica 1). Várias das plataformas apresentadas no Capítulo 2 utilizam este modelo [Banavar et al., 1998; Marsic, 1999; Stiemerling et al., 1999; Chabert et al., 1998; Hummes & Merialdo, 2000]. A utilização de plataformas e tecnologias de suporte difundidas facilita a adoção da solução e sua curva de aprendizagem é reduzida. A infra-estrutura de execução, que é montada a partir de uma composição de frameworks de infra-estrutura [Barreto et al., 2005], e os *component frameworks* oferecem diversos serviços de sistema, o que alivia a necessidade de o desenvolvedor escrever código para implementar serviços de baixo nível (característica 2).

Alguns exemplos de serviços providos são persistência, transação declarativa, concorrência de acesso, comunicação remota, logging, instalação, atualização e remoção, autenticação, segurança, tratamento e direcionamento de requisições, validação, tratamento de erros, geração de relatórios, etc. A infra-estrutura também automatiza diversos procedimentos repetitivos do desenvolvimento e da aplicação (característica 3).

A solução encapsula projetos e conceitos bem sucedidos e conhecidos pela comunidade de pesquisa em módulos passíveis de serem incorporados ao software com pouco esforço de programação (característica 4). A abordagem e a arquitetura propostas nesta tese implementam e disponibilizam na forma de componentes diversos aspectos do modelo 3C de colaboração e possibilitam que novos conceitos, ou variações dos existentes, sejam disponibilizados em componentes, que são integráveis às ferramentas colaborativas. O conjunto de componentes disponibilizados apresenta funcionalidades de uma maneira coesa e fracamente acoplada, pela própria natureza dos componentes de software (característica 5). Por fim, Greenberg [2006] afirma que as adaptações simples devem ser conseguidas com poucas linhas de código (característica 6). Os estudos de caso apresentados no Capítulo 5 desta tese mostram que diversas adaptações são realizadas através de customizações ou substituições de componentes por versões mais robustas. Quando estas opções não forem suficientes, um novo componente é desenvolvido e, respeitando-se as interfaces definidas na arquitetura, incorporado e integrado aos componentes existentes.

Ao disponibilizar um conjunto de componentes fundamentados em um modelo de colaboração e construídos de modo a encapsular as dificuldades técnicas de sistemas distribuídos e multi-usuário, é estabelecida uma linguagem através da qual os desenvolvedores pensam e investigam o suporte computacional à colaboração [Whorf, 1956]. Com isto, a colaboração passa a estar presente nas diversas atividades e artefatos do desenvolvimento de groupware. De acordo com Grosz [1996], “Collaboration must be designed into systems from the start; it cannot be patched on” (a colaboração deve ser incorporada no projeto dos sistemas desde o início; não pode ser remendada).

A utilização de um modelo de colaboração e de componentes fundamentados neste modelo favorece a redução da distância semântica entre

modelagem e implementação, de modo que os desenvolvedores pensam em termos da colaboração nas diversas atividades do desenvolvimento. O sistema colaborativo é estruturado a partir de componentes que encapsulam e refletem os conceitos do modelo 3C, mapeando as necessidades da colaboração a arranjos de componentes. Este ferramental possibilita que programadores criem protótipos e investiguem suas idéias, copiando e variando idéias de outros, compondo novos ambientes e ferramentas colaborativas. O desenvolvedor, instrumentado pelos componentes pré-elaborados e pela infra-estrutura de execução, direciona seus esforços à investigação e projeto de um efetivo suporte à colaboração. O perfil deste desenvolvedor não necessita mais ser tão especializado em questões técnicas de baixo nível, o que é muito favorável ao desenvolvimento do ambiente AulaNet. De acordo com o diretor de tecnologia da empresa EduWeb, sr. Luidi Xavier Fortunato, é difícil de encontrar e caro de contratar especialistas na tecnologia de suporte.

O reuso obtido com componentes de software favorece a produtividade e a melhoria da qualidade [Pfleeger, 2001]. Boas soluções e inovações no suporte à colaboração são reusadas por diversos grupos de pesquisa e instituições, o que favorece o refinamento e amadurecimento dos componentes. O AulaNet é disponibilizado com um conjunto padrão de serviços que pode ser estendido para atender às necessidades específicas de cada grupo e situação, bem como à sua evolução.

6.1. Contribuições

A principal contribuição desta tese é a proposta de utilização do modelo 3C de colaboração, originado de um artigo seminal de Ellis et al. [1991], freqüentemente citado e utilizado na literatura, para embasar o desenvolvimento de groupware, instrumentando o desenvolvedor com kits de componentes fundamentados no modelo. Conforme discutido no Capítulo 2 desta tese, esta abordagem é inovadora, e, conforme discutido no Capítulo 5, vem se mostrando viável e tem encontrado boa aceitação nos veículos científicos da área e de áreas afins.

Outra contribuição deste trabalho é o aprofundamento e refinamento do modelo 3C. Diversos trabalhos da literatura e ferramentas utilizadas no meio acadêmico e comercial foram estudados com este propósito. Além disto, o modelo 3C foi consolidado a partir da experiência adquirida nos 8 anos de desenvolvimento e utilização do ambiente AulaNet no suporte computacional à colaboração.

Outra contribuição deste trabalho é a proposição de uma arquitetura baseada em component frameworks e component kits para a construção de groupware. O próprio Collaboration Component Kit é uma contribuição deste trabalho. Como contribuição específica para o projeto AulaNet foram desenvolvidos sua nova arquitetura, com base na arquitetura proposta, e foram desenvolvidos dois de seus serviços. Apesar dos demais serviços não terem sido implementados, eles foram considerados no projeto e implementação da arquitetura. Com base no component kit proposto, os outros serviços serão construídos.

Por fim, o trabalho desenvolvido nesta tese deu origem a diversas publicações e a um curso do Departamento de Informática da PUC-Rio.

6.2. Limitações

A abordagem e o ferramental propostos nesta tese não se aplicam a qualquer groupware. A dinâmica da colaboração e os serviços colaborativos devem ser bem modelados pelo modelo 3C. A abordagem e o ferramental não são voltados a princípio para a implantação ou refinamento do suporte a colaboração em sistemas legados.

Nesta tese, só é tratada a camada de negócio do desenvolvimento de groupware. O desenvolvimento de groupware engloba muitas outras atividades e fatores. Por exemplo, a infra-estrutura de execução e a interface com o usuário também são cruciais para a criação de groupware propício para utilização e manutenção. A camada de infra-estrutura é tratada na dissertação de Celso Gomes Barreto [2006], que é parte deste consórcio de pesquisa. A camada de interface foi tratada apenas na arquitetura técnica, devendo ser investigada na arquitetura de aplicação.

Os estudos de caso desta tese foram restritos às ferramentas de comunicação Conferências e Debate. Mesmo estas ferramentas possuindo diversos elementos de coordenação e cooperação, além dos de comunicação, é necessário avaliar e refinar a engenharia do domínio para ferramentas de coordenação e de cooperação para verificar a abrangência do trabalho.

As dificuldades de aplicar groupware não são tratadas diretamente neste trabalho. Há diversos fatores envolvidos, como questões políticas, conflitos, intrigas, falta de estímulo, individualismos, falta de afinidade, questões culturais, etc. A composição do grupo de trabalho, por exemplo, tem uma influência direta em seu sucesso [Roussos et al., 1997]. Romano et al. [1998] relata que a presença de indivíduos controladores tende a desestabilizar a colaboração, e Borges et al. [2002] investiga os papéis informais que os participantes desempenham em uma reunião, de modo a compor grupos mais efetivos. A dinâmica de trabalho, as ferramentas selecionadas e a composição do ambiente de trabalho são fundamentais para propiciar a colaboração. Estes fatores são especialmente críticos em um ambiente voltado para o ensino-aprendizagem. Neste tipo de ambiente, normalmente não é possível escolher o grupo de trabalho: o grupo é composto pelos alunos que se matriculam no curso. Além disto, os alunos não estão habituados e capacitados a colaborar e os professores não estão habituados e capacitados a planejar dinâmicas que explorem a colaboração nas atividades educacionais. A própria utilização de ambientes educacionais pela web ainda é uma barreira a ser vencida [Lucena & Fuks, 2000].

Outra limitação desta tese é que o modelo 3C não foi investigado em grupos numerosos. Os grupos tratados possuíam menos de 20 participantes. De acordo com Teles [2004] a colaboração é dificultada em grupos numerosos, pois os subgrupos, as vaidades, as inimizades, etc. se intensificam, e é mais difícil para um coordenador organizar, sincronizar e unir o grupo. O suporte computacional à colaboração proposto nesta tese pressupõe equipes de trabalho reduzidas e coesas.

Este trabalho não foi comparado em termos de qualidade de software com outras abordagens. A qualidade de software é tratada pela norma ISO/IEC 9126 e sua equivalente brasileira NBR 13596, que estabelece seis características de qualidade de software: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Alguns destas características são dependentes

da infra-estrutura de execução e da arquitetura técnica adotada. A abordagem proposta nesta tese influencia mais diretamente as características funcionalidade, confiabilidade e manutenibilidade, principalmente as subcaracterísticas adequação, maturidade e modificabilidade, respectivamente. A adequação é favorecida pela disponibilização de serviços e componentes 3C, que propiciam a construção de um groupware mais ajustado às necessidades específicas de colaboração do grupo. A maturidade é favorecida pelo reuso dos componentes de software e pelo encapsulamento do suporte à colaboração. A modificabilidade é favorecida pela modularização e pela capacidade de substituição obtida com a componentização. Para comparar em termos de qualidade a abordagem proposta nesta tese com outras abordagens disponíveis na literatura, seria necessário que grupos externos desenvolvessem o mesmo ambiente, o AulaNet, por exemplo, utilizando as várias abordagens, para depois comparar os resultados. Por restrições de tempo e de recursos, isto não foi tratado no escopo desta tese. Algumas comparações foram feitas com o AulaNet 2.0 e o AulaNet 3.0. Contudo, os dois ambientes foram desenvolvidos em tempos diferentes, por desenvolvedores com perfis diferentes e com tecnologias e ferramentas diferentes, o que impossibilita extrair afirmações conclusivas. Além disto, não foi utilizada uma abordagem orientada para o reuso no AulaNet 2.0.

6.3. Trabalhos Futuros

Pretende-se investigar os aspectos relacionados à construção da interface com o usuário, levando em consideração o modelo 3C de colaboração e a arquitetura proposta nesta tese. Eventualmente, um mesmo componente 3C pode ser associado a mais de um widget de interface. Por exemplo, o componente CategorizationMgr, pode ser associado a um widget que monta a lista de categorias em um SelectBox ou a um widget que disponibiliza a lista de categorias através de Radio Buttons. Esta variedade de componentes de interface organizados em função dos componentes 3C favorece o desenvolvedor, que passa a contar com palhetas de componentes para seleção do que for mais apropriado para suas necessidades. Chung & Dewan [2004] propõem uma abordagem para

mapeamento dinâmico de componentes de interface com o usuário e de negócio, que pode ser levada em consideração nesta extensão da solução.

Os trabalhos futuros desta tese também incluem uma investigação e experimentação maior sobre qualidade de software no escopo da abordagem proposta e a investigação da utilização da solução apresentada em grupos numerosos e heterogêneos. Pretende-se também desenvolver adaptadores do modelo de componentes utilizado no AulaNet para as infra-estruturas de execução dos portalware, de modo a compatibilizar os serviços do AulaNet com os portalware e vice-versa.

Pretende-se também manter a experimentação com a turma da disciplina de Engenharia de Groupware, que utiliza a abordagem e a arquitetura propostas nesta tese para projetar uma extensão ou um novo groupware. Nas próximas edições do curso a documentação dos componentes será aprimorada e os alunos implementarão as propostas, utilizando o ferramental provido. Estas novas experimentações possibilitarão refinar o conjunto de componentes 3C e sua documentação. Para guiar os alunos no desenvolvimento da solução, será utilizado o processo proposto por Pimentel [2006], visando direcioná-los e reduzir a quantidade de falhas.

Os componentes 3C também serão utilizados pelo grupo de desenvolvedores do AulaNet do Laboratório de Engenharia de Software (LES) da PUC-Rio para implementar os demais serviços do ambiente. Depois de implementados, estes serviços serão entregues para o grupo de desenvolvedores da EduWeb, que irá trabalhar no código. Destes desenvolvimentos, espera-se também o refinamento do ferramental proposto.

Como trabalho futuro, conforme discutido no Capítulo 4, também pretende-se desenvolver e investigar a utilização de frameworks de domínio para a geração de aplicações, serviços e componentes de uma mesma família, com o objetivo de aumentar o reuso de código, facilitar a instanciação de componentes similares e consolidar e generalizar o suporte à colaboração.

Um problema ainda em aberto no desenvolvimento baseado em componentes é a construção de efetivos mecanismos de localização de componentes [Gimenes & Huzita, 2005]. O grau de reuso não cresce com o

aumento do número de componentes disponíveis, em parte pela dificuldade de localizá-los [Sametinger, 1997]. Pretende-se investigar como criar uma comunidade de desenvolvimento e de troca de componentes para o AulaNet, definindo maneiras de catalogar e buscar componentes, utilizando os recursos de catálogos e localização de serviços da plataforma web services.

Outro fator a ser investigado é a utilização da tecnologia de aspectos para capturar e explicitar os pontos de variabilidade dos componentes [Kulesza et al., 2004]. Pretende-se investigar também o tratamento do versionamento e o controle de dependências entre versões dos componentes. Os trabalhos futuros também estão direcionados para a criação de um ambiente voltado para o trabalho em grupo e para a definição de uma engenharia de groupware.

6.3.1.

O Ambiente eLabora

O conceito de trabalho está mudando, em parte devido ao ritmo de produção de conhecimento e ao aperfeiçoamento das tecnologias de telecomunicações. Profissionais dedicados ao trabalho intelectualizado são cada vez mais requisitados. Destes trabalhadores são exigidas novas habilidades: eles devem estar aptos a aprender continuamente, a trabalhar em grupo e a transformar criativamente conhecimento em novo conhecimento. Estas necessidades definem uma situação onde seus ambientes de trabalho e de aprendizado se confundem e se misturam [Fuks, 2000].

Pretende-se utilizar a experiência e a infra-estrutura desenvolvida para o AulaNet para a construção de um ambiente voltado para o trabalho em grupo, principalmente para o suporte a projetos colaborativos, onde normalmente há uma rotatividade de participantes e pouco suporte computacional. Este ambiente, já batizado de eLabora [Gerosa et al., 2001], integra as necessidades do ensino-aprendizagem e do trabalho. Com algumas adaptações, principalmente na nomenclatura, é possível imaginar que um curso do AulaNet é correspondente a um projeto do eLabora, uma turma a uma equipe, uma aula a uma etapa de trabalho, etc.

Com o eLabora, espera-se capacitar trabalhadores para enfrentar os desafios da sociedade do conhecimento, trabalhando em um ambiente similar ao que são treinados. O desenvolvimento do eLabora auxiliará também a refinar a arquitetura e os serviços propostos. Além disto, por utilizar a mesma infra-estrutura, um serviço colaborativo desenvolvido para o AulaNet poderá ser disponibilizado no eLabora e vice-versa.

6.3.2.

A Engenharia de Groupware Baseada no Modelo 3C

Conforme discutido no Capítulo 2, há diversas abordagens na literatura que estendem as ferramentas e técnicas da Engenharia de Software de modo específico para o desenvolvimento de groupware. Esta tese está inserida em um contexto de pesquisa que objetiva descrever uma engenharia de groupware baseada no modelo 3C e em técnicas de desenvolvimento baseado em componentes, instrumentando as diversas atividades do desenvolvimento de aplicações colaborativas. A engenharia de groupware é o processo sistemático, disciplinado e instrumentado pelo qual se modela e se desenvolve groupware [Fuks et al., 2002]. A engenharia de groupware une conceitos e tecnologias das áreas de Engenharia de Software e de CSCW para instrumentar o desenvolvimento e evolução de groupware.

Idealmente, um groupware deve ser prototipado [Schrage, 1996], dado sua tendência a falhas [Grudin, 1989]. Por isto, um ciclo baseado no desenvolvimento incremental [Boehm, 1988] mostra-se mais adequado para o desenvolvimento de groupware. Este ciclo é especialmente útil em projetos com requisitos instáveis ou não bem-definidos, que é justamente o caso de groupware. O desenvolvimento incremental com sucessivos testes de aceitação e de usabilidade adapta o groupware na direção das reais necessidades do usuário. Como são necessárias várias iterações e as necessidades não se estabilizam, visto que o grupo continuamente muda suas características, o desenvolvimento baseado em componentes mostra-se uma solução para propiciar a prototipação rápida, a experimentação e a adaptação do suporte computacional à colaboração. A Figura 6.2 ilustra o ciclo de desenvolvimento da engenharia de groupware proposta [Fuks et al., 2005].

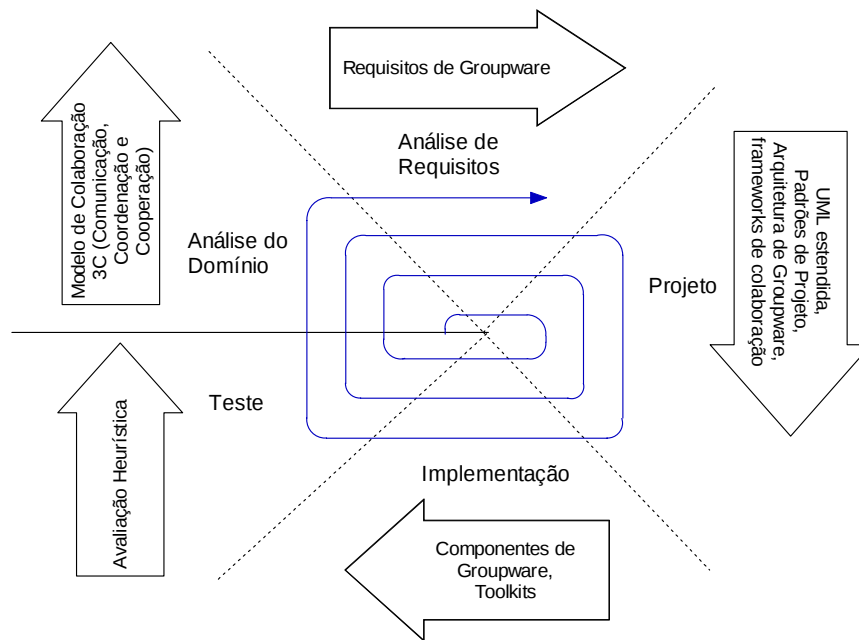


Figura 6.2. Ciclo da engenharia de groupware

O ferramental de suporte das diversas atividades do ciclo de desenvolvimento de groupware é embasado no modelo 3C, o que propicia trabalhar com os mesmos conceitos desde os primeiros momentos de análise até os testes finais, valorizando o desenvolvimento iterativo. Mantendo os mesmos conceitos na análise do domínio, na análise de requisitos, no projeto, na implementação e nos testes cria-se um vocabulário comum, mapeando conceitos e termos da realidade para a implementação, diminuindo a distância semântica e favorecendo a evolução da aplicação.

Considerando a análise do domínio como “o processo de identificar, coletar, organizar e representar a informação relevante de um domínio com base no estudo de sistemas existentes, conhecimento capturado de especialistas do domínio, fundamentação teórica e tecnológica” [Hess et al., 2000], considera-se que o modelo 3C instrumenta o desenvolvedor, que analisa a aplicação de sua ferramenta com base nos conceitos referentes à comunicação, à coordenação e à cooperação. A análise do domínio é utilizada como base para a definição e a especificação dos requisitos e é onde são identificados e organizados os conhecimentos sobre uma classe de problemas [Werner & Braga, 2005].

O levantamento de requisitos é guiado por requisitos genéricos de groupware e pelos requisitos de colaboração levantados durante a análise baseada no modelo 3C. Os requisitos, entretanto, nunca são precisos o suficiente para

possibilitar uma especificação precisa do sistema. O desenvolvimento incremental possibilita avaliar e validar continuamente os requisitos, bem como as decisões de projeto e de implementação. O desenvolvimento passado ajuda a entender o desenvolvimento futuro [Teles, 2004].

Há diferentes técnicas baseadas no modelo 3C que instrumentam o projeto de um groupware. Por exemplo, padrões, para reusar abordagens comuns de projeto e de análise de groupware baseado em componentes 3C; extensões da UML, para representar aspectos específicos de groupware e da colaboração direcionando a integração com a arquitetura; arquiteturas de groupware baseada em *component frameworks*; e frameworks de aplicação para instanciar ambientes, serviços e componentes 3C.

Para a fase de implementação, componentes pré-elaborados de comunicação, de coordenação e de cooperação podem ser plugados nos *component frameworks*, como ilustrado nos estudos de caso desta tese. São disponibilizadas também ao desenvolvedor ferramentas que auxiliam a montagem e customização, como a apresentada por Stiemerling et al. [2001].

Os requisitos e conceitos do modelo 3C também guiam a validação e os testes heurísticos do groupware, focando a atenção do avaliador em aspectos específicos de comunicação, de coordenação e de cooperação. Além das técnicas de avaliação heurística, são utilizadas técnicas tradicionais da Engenharia de Software, como testes unitários, para buscar a presença de erros no sistema.

O ciclo de desenvolvimento de groupware está inserido no contexto de um processo de desenvolvimento de groupware. O processo define as atividades e os artefatos a serem manipulados durante o desenvolvimento, que no caso da engenharia de groupware proposta, são baseados no modelo 3C de colaboração [Pimentel et al., 2005]. O processo prevê a instanciamento da solução e a criação de novos componentes.

6.4. Considerações Finais

A abordagem proposta nesta tese objetiva instrumentar o desenvolvedor de groupware para que ele, a partir das necessidades do trabalho em grupo, estruture

a aplicação colaborativa utilizando componentes fundamentados no modelo 3C de colaboração. Deve ser lembrado, porém, que como apontado por Laurillau & Nigay [2002], trazer a separação do suporte computacional dos diversos elementos da colaboração para a interface com o usuário pode trazer confusão e falhas de entendimento. Os elementos da interface com o usuário devem ser harmoniosamente combinados e posicionados próximos aos objetos que afetam.

Vale ressaltar também que a arquitetura componentizada do AulaNet não garante por si só a qualidade do produto final. O mesmo ocorre com um restaurante, cujo sucesso depende mais do desempenho do cozinheiro do que da qualidade da panela usada para o preparo da comida, dos ingredientes ou do prato no qual ela é servida. O sucesso do groupware é dependente de quem monta o ambiente e do desempenho de quem atua nele. Não basta conhecimento de programação para se projetar um groupware, são necessários também conhecimentos sobre o funcionamento da colaboração em um grupo de trabalho e dos requisitos de usuários e de desenvolvedores. O processo desenvolvido na tese de Mariano Pimentel ameniza esta dependência, pois para desenvolver groupware, o desenvolvedor segue os passos e gera os artefatos pré-estabelecidos [Pimentel et al., 2005].

Espera-se com a abordagem e a arquitetura propostas uma maior manutenibilidade e adaptabilidade do AulaNet. Espera-se, com isto, que a evolução do ambiente não desestruture o código, que passa a ser construído com base nos pilares do modelo 3C: comunicação, coordenação e cooperação.