

1 Introdução

Ao desenvolver groupware, o desenvolvedor depara-se com inúmeros desafios. Além de entender de seu domínio de aplicação, ou seja, colaboração, este deve lidar com aspectos de infra-estrutura de baixo nível, como o gerenciamento de transações, a persistência dos dados da aplicação e a validação dos dados de entrada do usuário. O desenvolvimento de groupware já é difícil por si só devido a seu caráter multidisciplinar e a heterogeneidade dos diversos grupos de trabalho (Gerosa et al., 2004) e não deve ser complicado por questões de baixo nível. A introdução frameworks de infra-estrutura como o Hibernate (2005), o Spring (2005) e o JSF (2005) possibilita que o desenvolvedor de groupware concentre-se em seu domínio e lide com estas questões a partir de uma perspectiva mais alto nível.

O AulaNet (Lucena & Fuks, 2000) é um groupware (Fuks et al., 2005) usado no ensino/aprendizagem colaborativo. Este é desenvolvido no Laboratório de Engenharia de Software (LES) da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). O objetivo da pesquisa que resultou nesta dissertação de mestrado é elaborar uma arquitetura técnica para groupware que ofereça uma base de serviços independentes de domínio, tomando como estudo de caso o ambiente AulaNet.

Esta introdução apresenta conceitos fundamentais sobre groupware. Também são apresentadas as arquiteturas do AulaNet 2.1 e 3.0. Por fim, são descritas as teses consorciadas a este trabalho e a organização do texto.

1.1. Groupware e o Modelo 3C

Groupware é software que apóia a interação entre indivíduos que trabalham para a realização de um objetivo comum (Rezende, 2003). Trabalhando colaborativamente, pelo menos potencialmente, pode-se produzir melhores resultados do que individualmente (Fuks et al., 2002).

O modelo 3C é usado para analisar a colaboração. Segundo este modelo, para colaborar, indivíduos devem se comunicar, coordenar e cooperar de forma adequada. A Figura 1.1 esquematiza o modelo 3C.

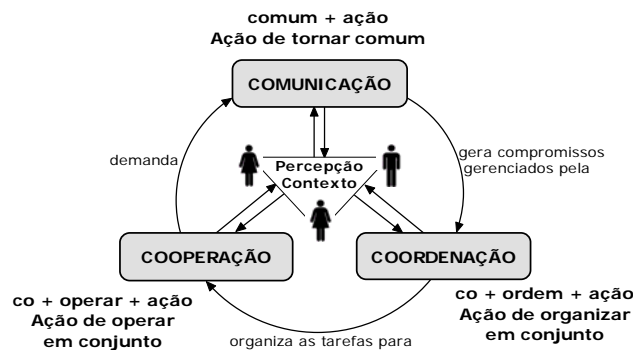


Figura 1.1 – O Modelo 3C (Pimentel et al., 2005)

Para que os objetivos sejam atingidos, um grupo precisa se comunicar, o que gera compromissos. Para garantir que estes sejam cumpridos e lidar com eventuais conflitos que possam surgir, é preciso que os compromissos gerados pela comunicação sejam coordenados. Coordenados, os integrantes do grupo podem cooperar, operando em conjunto e realizando o trabalho colaborativo. A cooperação demanda comunicação e o ciclo recomeça.

Para desenvolver groupware de qualidade, é necessário entender de colaboração (Pimentel et al., 2005). Utilizando o modelo 3C pode-se mapear características de um serviço em cada um dos “Cs” e, desta forma, melhorar a ferramenta através da modificação de um dos “Cs”. Pode-se também identificar deficiências de um groupware a partir da análise da colaboração baseada no modelo 3C, acrescentando ou removendo serviços de comunicação, coordenação e cooperação (Pimentel, 2006) de acordo com a necessidade do grupo de trabalho.

1.1.1. Comunicação

Participantes de uma equipe de trabalho devem se comunicar para conseguir realizar tarefas relacionadas, que necessitem de negociação ou que se encontram parcialmente descritas (Rezende, 2003). A comunicação em groupware pode ser efetuada de diversas formas. Alguns exemplos de ferramentas de comunicação disponíveis no AulaNet são o Debate e a Mensagem Instantânea, ferramentas de

comunicação síncrona, e a Conferência e o Correio para Turma, ferramentas de comunicação assíncrona. As ferramentas de comunicação síncrona privilegiam a interação entre os participantes, já que o tempo entre o envio de uma mensagem e a resposta do receptor é curto. Já as ferramentas de comunicação assíncronas valorizam a reflexão, pois os participantes têm mais tempo para formular suas mensagens.

Segundo o modelo 3C de colaboração, a comunicação é dita bem sucedida quando a mensagem tem o seu significado entendido pelo receptor (Fuks et al., 2005). A comunicação é realizada com o objetivo de gerar compromissos, porém uma mensagem mal interpretada pode gerar conflitos.

A leitura da mensagem altera o conhecimento do receptor e gera compromissos, que resultam em ações. Para ter indicações de que a mensagem foi compreendida o emissor deve observar as ações e o discurso do receptor (Fuks et al., 2005). Uma vez gerados os compromissos, é necessário coordenar o grupo para garantir o cumprimento dos compromissos.

1.1.2. Coordenação

Para que um grupo cumpra uma tarefa de forma eficiente, para que o trabalho em grupo produza um resultado melhor que a soma dos trabalhos individuais, os membros do grupo precisam ser coordenados (Fussell et al., 1998). A coordenação organiza o grupo para evitar que esforços de comunicação e cooperação sejam perdidos e que as tarefas sejam realizadas na ordem correta, no tempo correto e cumprindo as restrições e objetivos (Raposo et al., 2001). Alguns exemplos de ferramentas de coordenação disponíveis no AulaNet são o serviço de Avisos e o Acompanhamento de Participação.

A coordenação de tarefas envolve a pré-articulação, o gerenciamento das atividades e a pós-articulação. Durante a pré-articulação são realizadas todas as ações necessárias de preparação para a colaboração. Dentre estas tarefas incluem-se a definição dos objetivos, o mapeamento dos objetivos em tarefas, a seleção dos participantes e a distribuição das tarefas entre os participantes. Normalmente a fase da pré-articulação termina antes do início da cooperação. O gerenciamento das atividades consiste no controle das interdependências das tarefas executadas

para alcançar o objetivo. A pós-articulação ocorre após a conclusão das tarefas e envolve uma análise dos resultados e a documentação do processo colaborativo (Fuks et al., 2005).

Trabalhar colaborativamente demanda esforço para a coordenação dos membros de um grupo. Sem coordenação, parte dos esforços de comunicação não são aproveitados na cooperação. Para que o grupo possa operar em conjunto (cooperar) de forma satisfatória é necessário que os compromissos assumidos na comunicação entre os participantes sejam coordenados (Rezende, 2003).

1.1.3. Cooperação

Cooperação é o ato de operar em conjunto. Membros de um grupo cooperam, manipulando e organizando informação, construindo e refinando objetos de cooperação como documentos de texto, planilhas, gráficos entre outros. Mecanismos de cooperação provêm meios para gerenciar estes artefatos no espaço compartilhado, através do gerenciamento de versões, controle de acesso, busca, etc (Fuks et al., 2005). Alguns exemplos de ferramentas de cooperação disponíveis no AulaNet são o serviço de Aulas e o serviço de Co-Autoria.

O registro da informação na forma de objetos de cooperação visam aumentar o entendimento entre pessoas reduzindo a incerteza, relacionada à ausência de informação, e à equivocidade, relacionada à ambigüidade e a existência de informações conflitantes (Fuks et al., 2002).

Ao salvar as informações trocadas, o grupo pode armazenar a história de uma discussão que resultou em uma decisão. Contando com a memória coletiva, em um tempo futuro esta história pode ser consultada, retomando os motivos que levaram decisão.

1.2. O AulaNet

Dá-se o nome de learningware a um groupware que dá suporte a aprendizagem colaborativa (Santoro et al., 1999). O AulaNet é um learningware baseado na web desenvolvido pelo Laboratório de Engenharia de Software (LES)

da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) desde junho de 1997. Participam de seu desenvolvimento alunos de graduação, mestrado e doutorado que também o usam para aplicar suas pesquisas e escrever artigos, monografias, dissertações e teses.

Atualmente o AulaNet está disponível em português, inglês e espanhol. Seu código fonte é fechado, mas sua distribuição é feita gratuitamente pela empresa EduWeb (2005). Esta empresa também realiza customizações no AulaNet para uso nos seus diversos clientes.

Os serviços do AulaNet são categorizados segundo sua classificação de acordo com o modelo 3C em serviços de comunicação, coordenação e cooperação.

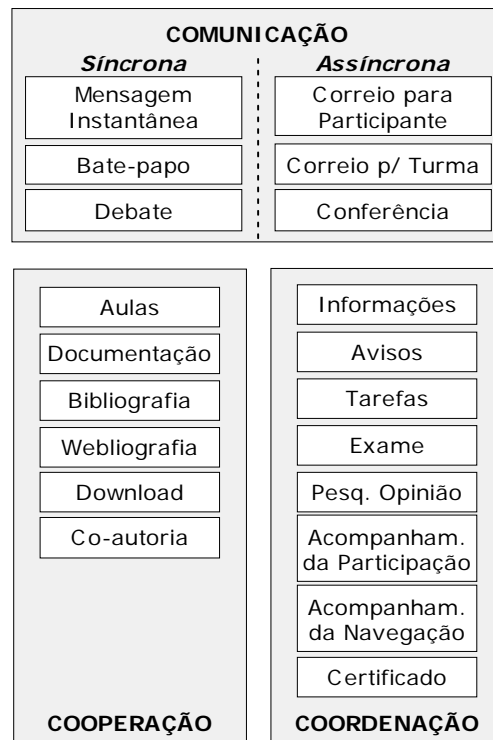


Figura 1.2 – Classificação dos Serviços do AulaNet segundo o Modelo 3C (Pimentel et al., 2005)

Os serviços de comunicação oferecem ferramentas com as quais os aprendizes dos cursos do AulaNet podem se comunicar de forma síncrona ou assíncrona. Os serviços de coordenação provêm serviços usados na gerência e controle das atividades de grupo. Os serviços de cooperação incluem serviços que

oferecem suporte ao conteúdo bibliográfico dos cursos, co-autoria entre aprendizes e outros.

Recentemente iniciou-se o desenvolvimento do AulaNetM, uma extensão dos serviços do AulaNet para dispositivos móveis. Esta iniciativa tem como objetivo explorar o uso do m-learning no AulaNet (Filippo et al., 2005a, b). A Figura 1.3 mostra a tela principal do AulaNet 3.0 e a Figura 1.4 a lista de mensagens da conferência do AulaNetM.



Figura 1.3 – AulaNet no Desktop.



Figura 1.4– AulaNetM em um PDA.

Ao longo dos 8 anos que o AulaNet vem sendo desenvolvido, foram escritos 4 livros ou capítulos de livros, 15 artigos em revistas, 57 artigos em conferências e workshops, 2 teses de doutorado, 9 dissertações de mestrado e 5 monografias de fim de curso relacionadas de alguma forma ao AulaNet. Além disso, o AulaNet está presente em várias universidades no Brasil, como a PUC-Rio, a Federal do Rio de Janeiro, a Federal da Bahia entre outras, e em várias universidades do resto do mundo, dentre elas Universidade Tecnológica do Panamá e Universidade da Madeira (Portugal). O AulaNet também é usado em empresas, entre elas a Rede Globo, NEXTEL e SENAC.

Os mesmos desenvolvedores do AulaNet mantêm os cursos Engenharia de Groupware, onde ele é usado como repositório de informações e o curso Tecnologias de Informação Aplicada A Educação (TIAE), onde ele é usado integralmente em um curso totalmente à distância. Estes cursos são ministrados semestralmente e estão disponíveis para alunos da graduação e pós-graduação da PUC-Rio. A experiência com estes cursos fornece subsídios usados na melhoria constante do AulaNet.

O AulaNet é um learningware desenvolvido pelo LES desde 1997 e é usado em diversas empresas e universidades. O AulaNet também é usado para testar

hipóteses de pesquisas em diversos trabalhos de alunos da graduação, mestrado e doutorado da PUC-Rio. Atualmente, a versão distribuída do AulaNet é a 2.1. Sua arquitetura é descrita na seção a seguir.

1.2.1. A Arquitetura do AulaNet 2.1

O AulaNet é desenvolvido desde junho de 1997. Inicialmente seu código era escrito em LUA (Ierusalimschy et al., 1996). O código evoluiu rapidamente e então decidiu-se utilizar a linguagem de programação Java, de modo a melhorar a integração com o mercado de desenvolvedores e aproveitar a robustez, portabilidade e recursos desta plataforma. O AulaNet foi então totalmente reformulado e sua versão 2.0 escrita em Java foi lançada.

Desde então, a sua arquitetura não sofreu alterações significativas. Atualmente, a versão de produção do AulaNet é a 2.1 que possui a mesma arquitetura da 2.0. Esta é baseada no paradigma cliente-servidor e é fortemente dependente da tecnologia Scriba (Fuks et al., 2003). A Figura 1.5 exibe o esquema da arquitetura do AulaNet 2.1.

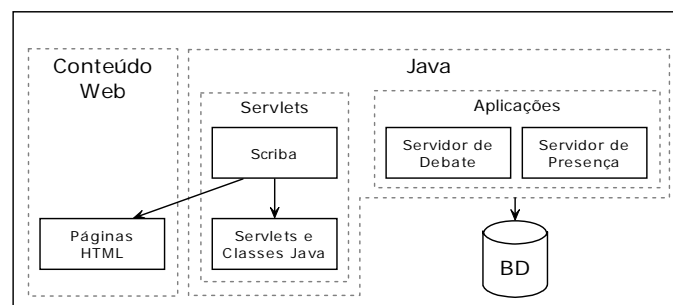


Figura 1.5 – Arquitetura do AulaNet 2.0 (Barreto et al., 2005).

A tecnologia Scriba é constituída por uma linguagem de script que é embutida em arquivos HTML e um Servlet que interpreta os arquivos HTML. O Scriba possibilita, dentre outras coisas, chamadas a classes Java, acesso a banco de dados e definições de variáveis.

A maior parte das páginas HTML/Scriba está relacionada a classes Java, que acessam o banco de dados, realizam algum processamento e constroem a representação gráfica do resultado. O código que representa a lógica da aplicação

não é separado do código que representa a interface com o usuário e encontra-se espalhado tanto nas páginas HTML/Scriba quanto nas classes Java.

Os principais problemas desta arquitetura são a baixa modularidade, o uso de um paradigma procedural, a não aderência a tecnologias e boas práticas atuais e a não separação da lógica da aplicação da interface com o usuário.

O código do AulaNet 2.1 vem sendo desenvolvido ao longo destes anos através de prototipação. Isto, aliado a rotatividade dos desenvolvedores do projeto, com o passar do tempo resultou em um código pouco modular e não padronizado. Como consequência, o código do AulaNet hoje apresenta um baixo grau de reuso.

Apesar de usar a linguagem Java, que é orientada a objetos, o código presente no AulaNet 2.1 assemelha-se à abordagem funcional, sem que seja utilizado uma modelagem de classes. Desta forma, muitas das vantagens inerentes ao paradigma OO não são aplicáveis ao AulaNet 2.1.

Muitas das tecnologias e das boas práticas presentes hoje não estavam presentes quando o AulaNet 2.1 foi desenvolvido. Um exemplo disto é a tecnologia JSP, que surge em meados de 1999 e oferece recursos semelhantes à tecnologia Scriba. O uso de tecnologias proprietárias como o Scriba dificulta a integração com outras equipes de desenvolvimento, como a EduWeb, que precisa treinar cada novo funcionário nesta tecnologia.

Por fim, a falta de uma separação clara entre a lógica da aplicação e a interface com o usuário dificulta o reuso, pois a lógica de negócios não pode ser reutilizada separadamente.

A soma destes problemas resultou em um AulaNet cuja atividade de manutenção é muito cara e cuja entrada de novos desenvolvedores ou integração com outros grupos de desenvolvedores é difícil. Para solucionar estes problemas, o consórcio de teses Groupware@LES propôs uma nova arquitetura para o AulaNet.

1.3. Consórcio de Pesquisa

Para investigar o desenvolvimento de groupware e sua aplicação no desenvolvimento do AulaNet 3.0, nosso grupo de pesquisa Groupware@LES consorciou três trabalhos: esta dissertação de mestrado propõe a integração de

frameworks na constituição da arquitetura técnica do AulaNet; Gerosa (2006), em sua tese de doutorado, propõe a montagem de groupware a partir da agregação de serviços e componentes baseados no Modelo 3C de Colaboração; e Pimentel (2006), em sua tese de doutorado, propõe um processo de desenvolvimento de groupware usando o Modelo 3C de Colaboração em diferentes etapas do processo. Estes trabalhos consorciados reduzem a distância semântica entre a implementação e os conceitos do domínio referentes à colaboração, o que favorece a manutenção e a evolução do groupware. Com o objetivo de contextualizar os três trabalhos, esta seção é replicada na introdução da dissertação e das teses resultantes. As subseções seguintes resumem cada trabalho.

1.3.1.

Agregando Frameworks de Infra-Estrutura em uma Arquitetura Baseada em Componentes: Um Estudo de Caso no Ambiente AulaNet

No desenvolvimento de um groupware, o projetista se depara com desafios em diferentes níveis: entender do domínio e lidar com questões de infra-estrutura. O desenvolvimento de groupware é difícil devido ao seu caráter multidisciplinar e à heterogeneidade dos diversos grupos de trabalho. O desenvolvedor deveria se concentrar mais nos aspectos funcionais utilizando uma infra-estrutura que trate as questões técnicas.

Nesta dissertação, foi elaborada uma arquitetura técnica multicamadas que faz uso do padrão MVC (Fowler, 2002) e que integra frameworks de infra-estrutura (Fayad & Schmidt, 1997; Fayad et al. 1999a, b; Fayad & Johnson, 2000). A abordagem multicamadas com o padrão MVC proporciona a separação entre a lógica da aplicação e a interface com o usuário, considerada uma boa prática de design de software (Fowler, 2002). Frameworks de infra-estrutura proporcionam uma maneira de lidar com as questões de baixo nível como persistências de dados, controle de transações, segurança, etc.

O diagrama esquematizado na Figura 1.6 mostra a arquitetura técnica proposta para o AulaNet 3.0. As setas indicam o fluxo de controle da aplicação, os retângulos representam classes, os círculos representam as interfaces, e as linhas pontilhadas representam a divisão entre as camadas. Esta arquitetura, baseada na

Arquitetura de POJOs descrita por Johnson (2002, 2004), é organizada nas seguintes camadas: apresentação, negócios e recursos.

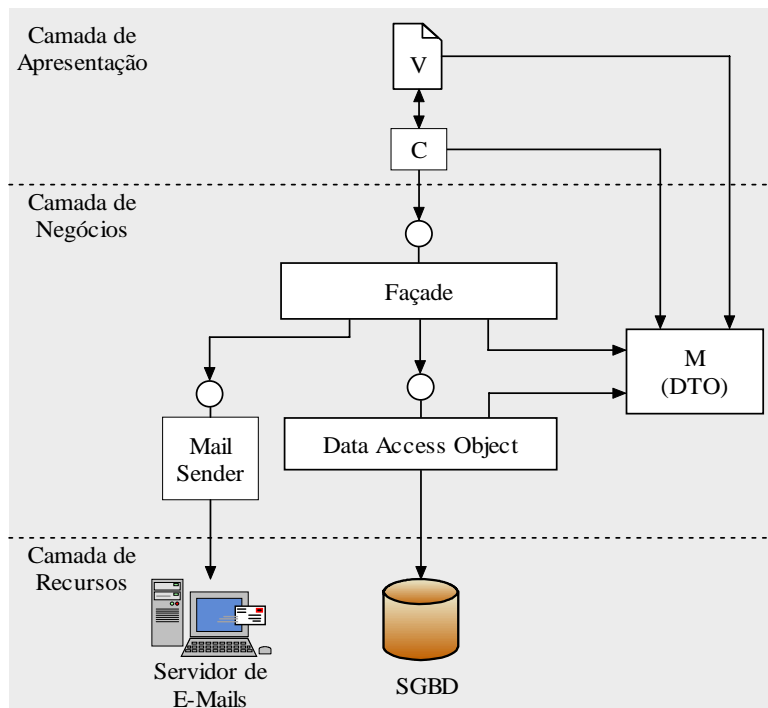


Figura 1.6 – Arquitetura Técnica do AulaNet 3.0.

A camada de recursos relaciona os recursos externos necessários para que a aplicação seja executada. Na arquitetura do AulaNet 3.0 estão previstos o uso de banco de dados relacional (SGBD) e um servidor de e-mails.

A camada de negócios implementa a lógica da aplicação utilizando POJOs (POJO, 2005) (Plain Old Java Objects).

“O termo [POJO] foi cunhado enquanto eu [Martin Fowler], Rebecca Parsons e Josh MacKenzie estávamos nos preparando para uma conferência em Setembro de 2000. Na palestra estávamos levantando os vários benefícios de codificar a lógica de negócios usando objetos Java comuns em vez de usar Beans de Entidade [EJB]. Questionávamos por que as pessoas eram tão contra usar objetos comuns em seus sistemas, e concluímos que era pela falta de um nome pomposo para os objetos simples. Então inventamos um, e o termo pegou muito bem.” (POJO, 2005)

Na camada de negócios, o modelo (representado no diagrama da Figura 1.6 pela letra M do MVC) é implementado por classes que realizam o padrão de projetos *Data Transfer Object* (Fowler, 2002), usadas para transportar os dados das entidades de negócio entre camadas. O acesso à base de dados é encapsulado através de classes que implementam o padrão de projetos *Data Access Objects* (DAO) (Alur et al., 2001), o que possibilita variar a maneira de persistir as classes

do modelo sem que seja preciso reescrever o código cliente. Mail Sender é a classe para enviar e-mails que, de forma similar ao DAO, encapsula o acesso ao servidor de e-mails. A lógica de negócios é exposta para a camada de apresentação através de um *Facade* (Gamma et al., 1995), que é o padrão de projeto para prover uma interface para acesso às funcionalidades de um serviço.

A camada de apresentação expõe a lógica de negócios ao usuário-final. Na arquitetura do AulaNet 3.0, esta camada é composta pelo controlador (representado no diagrama da Figura 1.6 pela letra C do MVC) e por páginas JSP que implementam a camada de Visão (representado no diagrama da Figura 1.6 pela letra V do MVC). O controlador chama os métodos do *Facade*, acessando a camada de negócios. Os DTOs resultantes de operações são passados à visão que exibe as informações ao usuário.

Frameworks de infra-estrutura foram selecionados e acrescentados a esta arquitetura para prover persistência de dados, gerenciamento de transações entre outros aspectos. Estes frameworks possibilitam ao desenvolvedor tratar estes aspectos com uma visão em alto nível, concentrando-se em seu domínio de aplicação, no caso, colaboração.

1.3.2. Desenvolvimento de Groupware Componentizado com base no Modelo 3C de Colaboração

Um groupware é composto de ferramentas colaborativas como Fórum, Agenda, Documentação, etc. Estas ferramentas, disponíveis em diversas aplicações groupware, compartilham funcionalidades relativas ao suporte computacional à colaboração, tais como canal de comunicação, gerenciamento de participantes, registro de informações, etc.

Na tese de Gerosa (2006), para dar suporte ao desenvolvimento de groupware, foram estabelecidos dois níveis de componentização. O primeiro nível é constituído de serviços colaborativos que, por sua vez, são montados com componentes 3C (segundo nível) que implementam funcionalidades relacionadas à colaboração. Estes componentes são distribuídos em *component kits* organizados em função do modelo 3C de colaboração para que desenvolvedores montem aplicações colaborativas. Nesta abordagem, cada serviço usa componentes de comunicação, coordenação e de cooperação independentemente da classificação

3C do serviço. Foi aplicado um método de Engenharia do Domínio para elaborar o conjunto de componentes. Os componentes são iterativamente refinados em função da realimentação obtida com o desenvolvimento dos serviços do AulaNet 3.0 e em função de estudos de caso variando as configurações do suporte à colaboração.

Component frameworks (Szyperki, 1997) são usados para oferecer suporte ao gerenciamento e à execução dos componentes. Conforme apresentado na Figura 1.7, na tese de Gerosa (2006) foi elaborado um *component framework* para cada nível de componentização (serviço e componente 3C). Os serviços são acoplados no Service Component Framework, e os componentes 3C são acoplados no Collaboration Component Framework. Estes *component frameworks* são responsáveis por tratar a instalação, remoção, atualização, ativação, desativação, localização, configuração e monitoramento de componentes. O Service Component Framework gerencia as instâncias dos serviços e a ligação com os componentes de colaboração correspondentes. O Collaboration Component Framework gerencia as instâncias dos componentes de colaboração, que são provenientes do Collaboration Component Kit. Algumas funcionalidades dos *component frameworks* são recorrentes, sendo então elaborado um *framework* para instanciar os *component frameworks*. Este tipo de *framework* é chamado de *component framework framework* (CFF) (Szyperki, 1997, p.277). Um *component framework framework* é visto como um *component framework* de segunda ordem, onde seus componentes são *component frameworks* (Szyperki, 1997, p.276). Na arquitetura da aplicação, o *component framework* de segunda ordem foi denominado Groupware Component Framework Framework.

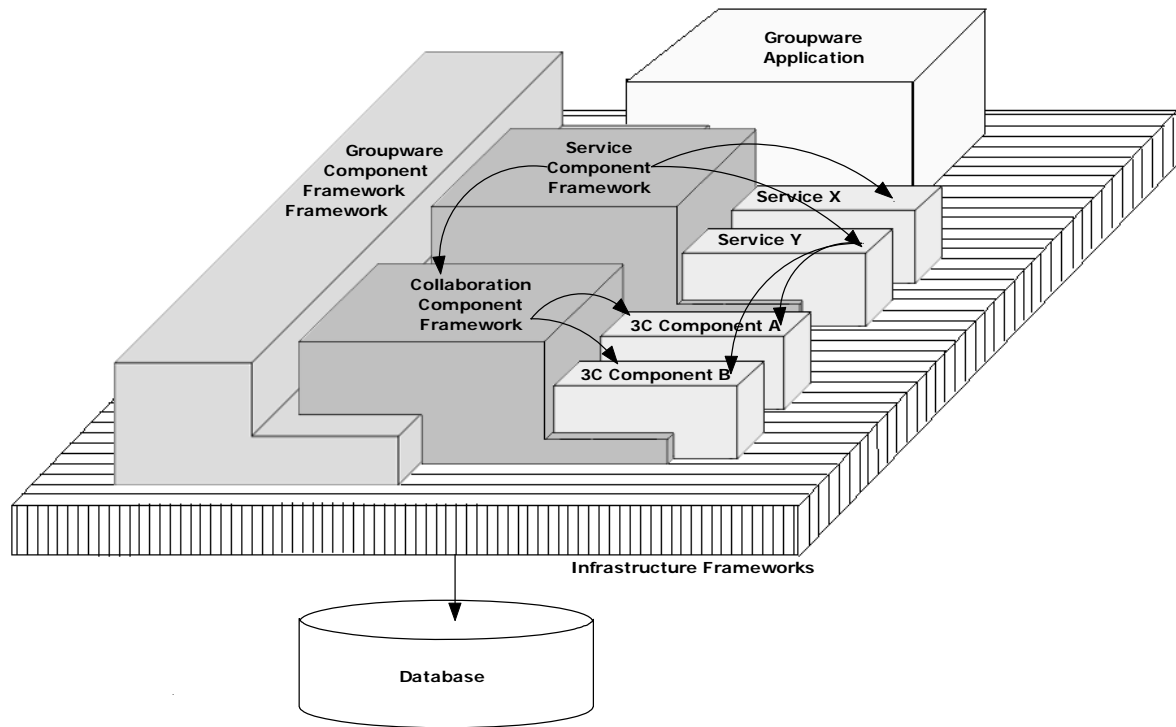


Figura 1.7. A arquitetura de aplicação proposta

Os *component frameworks*, serviços e componentes 3C oferecem suporte computacional aos conceitos do modelo 3C de colaboração, instrumentando o desenvolvimento da camada de negócio. A arquitetura de aplicação proposta estrutura os componentes do domínio, representando um projeto lógico de alto nível independente da tecnologia de suporte (D’Souza & Wills, 1998). Já os aspectos de infra-estrutura, tratados nesta dissertação, são independentes do domínio de aplicação.

Os componentes da arquitetura de aplicação são implementados segundo a arquitetura técnica. Os serviços do AulaNet são criados com um único *Facade* que expõe as operações deste serviço para a camada de apresentação. Os componentes de colaboração por sua vez, podem utilizar vários DTOs e DAOs, dependendo da complexidade do componente. Estes componentes podem ainda usar “código cola” (Szyperski, 1997) e adaptadores (D’Souza & Wills, 1998) para possibilitar a integração com componentes e outros sistemas que não são compatíveis por construção.

1.3.3. RUP-3C-Groupware: um Processo de Desenvolvimento de Groupware baseado no Modelo 3C de Colaboração

Os *frameworks* de infra-estrutura selecionados nesta dissertação se encarregam de soluções para aspectos de infra-estrutura de baixo nível, visando possibilitar o desenvolvedor se concentrar nos aspectos funcionais. Os *kits* de serviços e componentes 3C elaborados por Gerosa (2006) fornecem os elementos para compor um groupware. O processo elaborado na tese de Pimentel (2006) estabelece os passos a serem seguidos na montagem do groupware, pois ainda que se construa uma aplicação groupware para um grupo com uma determinada dinâmica, com o tempo surgem novas situações onde são identificados novos problemas. A aplicação necessitará ser modificada para não se manter inadequada.

Um processo organiza, em linhas gerais, uma seqüência de passos onde são incorporadas diretrizes e boas práticas que, quando seguidas, levam à produção de um software razoável (Sommerville, 2003; Beck, 2004; Philippe, 2003). Coexistem abordagens diferentes para o desenvolvimento de software, dentre elas, o desenvolvimento baseado em componentes, que é uma estratégia recente que tem se tornado cada vez mais usada (Sommerville, 2003; Gimenes & Huzita, 2005). Seguindo esta abordagem, tornaram-se conhecidos processos como Catalysis (D’Souza e Wills, 1998), *UML Components* (Cheesman & Daniels, 2001) e RUP – Rational Unified Process (Philippe, 2003). O processo formalizado na tese de Pimentel (2006), denominado RUP-3C-Groupware, também faz uso da abordagem baseada em componentes, estendendo o RUP para o desenvolvimento específico de aplicações groupware.

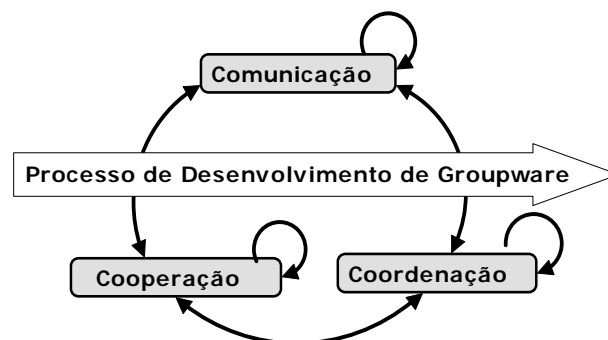


Figura 1.8. Foco para o desenvolvimento de uma versão da aplicação groupware com base no Modelo 3C de Colaboração

No processo proposto, o Modelo 3C de Colaboração é usado nas diferentes etapas do processo: na análise de domínio para classificação das aplicações groupware e de seus elementos; na construção de componentes (Gerosa, 2006); e no foco dado para o desenvolvimento de cada versão. De acordo com essa prática, a aplicação groupware é desenvolvida resolvendo um problema de comunicação, de coordenação ou de cooperação, um a cada versão ao longo do ciclo de desenvolvimento, esquematizado na Figura 1.8.

1.4. Organização da Escrita

O texto desta dissertação está estruturado em 7 capítulos e 1 apêndice.

No capítulo 2 são apresentados conceitos importantes sobre frameworks. Este capítulo apresenta definições, classificações e outras características de frameworks pesquisadas na literatura.

No capítulo 3 a camada de negócios da arquitetura do AulaNet 3.0 é detalhada. É feita uma breve descrição do modelo de componentes EJB, levantando suas vantagens, desvantagens e a perspectiva para seu futuro. Neste capítulo, após uma análise e comparação criteriosa, são escolhidos o framework Hibernate (2005) e o Spring (2005) para serem acrescentados à camada de negócios do AulaNet 3.0.

No capítulo 4 o foco é a camada de apresentação. O grande número de web frameworks disponíveis no mercado demanda uma análise mais detalhada entre os concorrentes desta categoria de framework. Ao término da comparação, o web framework JavaServer Faces (JSF, 2005) é incorporado a camada de apresentação do AulaNet 3.0.

O capítulo 5 retoma a descrição da arquitetura do AulaNet 3.0 apresentada neste capítulo, complementando-a com os frameworks analisados tanto no capítulo 3 quanto no capítulo 4. Também é apresentado neste capítulo como a arquitetura do AulaNet 3.0 provê suporte a dispositivos móveis e que novos frameworks podem ser incorporados a arquitetura, tomando como estudo de caso o framework de agentes Jade (2005). Este capítulo mostra ainda os benefícios que o paradigma de agentes de software pode trazer ao AulaNet.

O capítulo 6 apresenta as conclusões finais e as contribuições desta pesquisa enquanto que o capítulo 7 apresenta as referências bibliográficas. O apêndice A é adicionado com a transcrição dos e-mails trocados com outros pesquisadores, que gentilmente cederam dados tornando esta pesquisa mais precisa. O apêndice B fornece detalhes sobre o método utilizado para comparação de frameworks utilizando critérios não-técnicos.

Ao longo desta dissertação são apresentadas várias listagens de código. Para destacar as listagens do texto da dissertação são usadas caixas especiais. A Figura 1.9 mostra um exemplo destas caixas.

```
30: public class HelloReaders {  
  
31:     public static void main(String args[]) {  
32:         System.out.println("Olá, leitores!");  
33:     }  
  
14: }
```

Figura 1.9 – Exemplo de uma Listagem de Código.

O início de cada linha é precedido de uma numeração, para facilitar a referência no texto. Linhas em branco não são numeradas. Dentro de uma seção a numeração sempre começa do fim da numeração da listagem anterior. Por exemplo, a listagem acima começa na linha 30, então se supõem que haveria uma listagem anterior cuja última linha seria 29. Numerando as linhas desta forma o texto torna-se mais claro, pois citações de linhas não se confundem quando há mais de uma listagem muito próxima. As listagens de código seguem as convenções de código definidas no Sun Java Code Conventions (SJCC, 2006) sempre que possível, mas eventualmente podem fugir do padrão para tornar a listagem mais clara. Alguns detalhes, como declarações e importações de pacotes são omitidos pelo bem da clareza.

Toda a pesquisa que resultou nesta dissertação foi baseada em pesquisas teóricas e aplicações práticas. Um protótipo do serviço da Conferência do AulaNet foi implementado utilizando os principais frameworks apresentados nesta dissertação. As listagens de código presentes nesta dissertação são baseadas em código extraído destes protótipos, simplificadas para fins didáticos.

Nesta dissertação, os frameworks também são analisados sob um ponto de vista não-técnico. São comparados os seguintes critérios: disponibilidade de documentação, de suporte e de ferramentas compatíveis, grau de aceitação no mercado e disponibilidade de profissionais aptos a trabalhar com cada framework.

Apesar de ser muito difícil obter resultados absolutos para estes critérios, Raible (2005) apresenta um método para colher indícios. A quantidade de documentação pode ser checada através de uma busca por livros sobre o framework na página da livraria Amazon e por artigos disponíveis na web sobre o framework através da ferramenta de busca Google. A disponibilidade de suporte é verificada contabilizando o volume de mensagens trocadas nas listas de discussões oficiais de cada framework. Dessa forma, obtém-se um indício do grau de atividade da comunidade que oferece suporte ao framework. A compatibilidade com ferramentas é verificada analisando um conjunto de ferramentas, verificando a documentação destas uma a uma e checando se são compatíveis com o framework. O grau de aceitação no mercado é verificado através de buscas em sites de empregos, procurando por vagas que solicitem habilidades em cada framework. Por fim, a disponibilidade de profissionais é verificada através de buscas realizadas em sites que hospedam currículos. São contabilizados os currículos de profissionais que se declaram aptos a trabalhar com cada framework.