

Uma Arquitetura para o Desenvolvimento de Ferramentas Colaborativas para o Ambiente de Aprendizagem AulaNet

Marco Aurélio Gerosa¹, Alberto Barbosa Raposo²,
Hugo Fuks¹, Carlos José Pereira de Lucena¹

¹Laboratório de Engenharia de Software (LES), Departamento de Informática

²Grupo de Tecnologias em Computação Gráfica (Tecgraf), Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Rua Marquês de São Vicente, 225, Rio de Janeiro, RJ, 22453-900, Brasil

{gerosa, hugo, lucena}@inf.puc-rio.br, abraposo@tecgraf.puc-rio.br

Abstract. *Collaborative systems should be flexible enough to adapt to group characteristics and to work processes evolution. In this paper, the new architecture of the AulaNet learning environment is presented. This architecture was developed using component-based techniques to provide the necessary flexibility. A case study is presented, showing how the component-based architecture accommodates the changes needed to support the evolution of a course dynamics.*

Resumo. *Sistemas colaborativos devem ser suficientemente flexíveis para serem adaptados às características do grupo e à evolução dos processos de trabalho. Neste artigo é apresentada a nova arquitetura do ambiente AulaNet, construída com técnicas de desenvolvimento baseado em componentes para prover a flexibilidade necessária. É apresentado um estudo de caso com um dos serviços do ambiente, para mostrar como a ferramenta pode ser adaptada para acompanhar a evolução da dinâmica do curso.*

1. Introdução

Sistemas colaborativos são especialmente vulneráveis a falhas [Grudin, 1989]. Eles envolvem aspectos multidisciplinares em sua construção e são difíceis de aplicar e de testar. Isto é especialmente verdade em ferramentas colaborativas para o ensino-aprendizagem, que normalmente ainda contam com o complicador de envolverem uma turma heterogênea, de curta duração e muitas vezes despreparada para a utilização da tecnologia no ensino-aprendizagem.

Para reduzir este problema, os sistemas colaborativos precisam ser desenvolvidos e refinados iterativamente, de forma que a realimentação fornecida pelo seu uso seja usada para guiar a evolução e a adaptação da ferramenta. Como cada grupo utilizando uma ferramenta tem características e objetivos distintos, fica muito difícil para o desenvolvedor prever todas as possibilidades de utilização possíveis e dar suporte à maioria delas. Ainda que isto fosse possível, tornaria o sistema excessivamente configurável e difícil de utilizar.

Uma outra solução para o problema seria fornecer APIs (*Application Program Interface*) de programação para que o docente pudesse desenvolver, a partir de primitivas previamente desenvolvidas, a ferramenta que mais lhe conviesse. Porém, o docente normalmente não domina a arte da programação e não possui recursos para que outras pessoas desenvolvam para eles.

Uma solução meio-termo é oferecer um software componentizado para que o docente possa plugar e desplugar componentes de forma a montar uma ferramenta adequada a seus propósitos. E mais do isto, pode experimentar diversas configurações e, a partir da realimentação obtida com a utilização da ferramenta, adaptá-la e refiná-la, de forma a acompanhar a evolução da dinâmica do seu curso e do próprio grupo, visto que este também evolui.

A nova arquitetura do ambiente AulaNet foi projetada visando esta flexibilidade. O docente poderá montar sua ferramenta sem ter que escrever código para isto. Isto também é interessante para instituições que utilizam o ambiente, visto que poderão adaptá-lo e desenvolver novos componentes para seus propósitos específicos.

Neste artigo é apresentada a nova arquitetura do ambiente AulaNet, juntamente com um estudo de caso que exemplifica a necessidade de adaptar a ferramenta à evolução da dinâmica do curso e das características da turma. O estudo de caso é a evolução do serviço Debate do ambiente. O serviço evolui a partir das observações e necessidades extraídas do curso Tecnologias de Informação Aplicadas à Educação (TIAE). Na seção seguinte o ambiente AulaNet e o curso TIAE são sucintamente apresentados, juntamente com o modelo de colaboração 3C, que norteou e foi refinado a partir da evolução do ambiente e do curso. Na Seção 3, a arquitetura é apresentada, na Seção 4 o estudo de caso é abordado e a Seção 5 conclui o artigo.

2. A Colaboração no Ambiente AulaNet

O AulaNet é um ambiente baseado numa abordagem groupware para o ensino-aprendizagem na Web que vem sendo desenvolvido desde 1997 pelo Laboratório de Engenharia de Software da Universidade Católica do Rio de Janeiro (PUC-Rio). O AulaNet é um freeware, e está disponível nas versões em português, inglês e espanhol em <http://groupware.les.inf.puc-rio.br> e <http://www.eduweb.com.br>.

2.1. O Modelo de Colaboração 3C

O modelo de colaboração adotado no ambiente AulaNet é baseado na idéia de que para colaborar, um grupo tem que exercer três atividades principais: comunicar-se, coordenar-se e cooperar [Ellis et al, 1991], conforme pode ser observado na Figura 1.

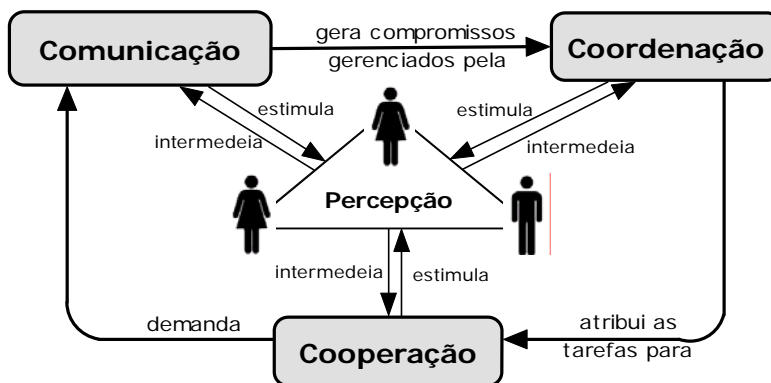


Figura 1. Diagrama do Modelo de Colaboração 3C.

A comunicação, em um grupo de trabalho, envolve a negociação de compromissos e conhecimento. Através da coordenação, o grupo lida com conflitos e se organiza de maneira a evitar que os esforços de comunicação e de cooperação sejam perdidos. A cooperação é a operação conjunta dos membros do grupo em um espaço compartilhado.



Figura 2. Classificação dos serviços do ambiente AulaNet em função do modelo 3C.

Nas primeiras versões do ambiente AulaNet, seus recursos eram classificados em serviços administrativos, de avaliação e didáticos, que é uma abordagem comum em ferramentas educacionais [Edutools, 2004]. Para incentivar a colaboração no ambiente, os serviços foram reorganizados com base no modelo de colaboração 3C [Fuks, 2000], sendo atualmente classificados como serviços de comunicação, de coordenação e de cooperação, conforme pode ser observado na Figura 2.

2.2. O Curso Tecnologias de Informação Aplicadas à Educação

O curso TIAE (Tecnologias de Informação Aplicadas à Educação) é ministrado desde 1998 como uma disciplina do Departamento de Informática da PUC-Rio, totalmente a distância pelo ambiente AulaNet. O objetivo do curso é fazer os aprendizes colaborarem usando as tecnologias de informação, tornando-se educadores baseados na Web [Fuks, Gerosa & Lucena, 2002]. O curso visa construir uma rede de aprendizagem [Harasim et al., 1997] onde o grupo aprende, primordialmente, através das interações dos participantes em atividades colaborativas.

O curso TIAE é organizado em tópicos, sendo um tópico abordado a cada semana, durante oito semanas. Os aprendizes lêem os conteúdos selecionados sobre o tópico, realizam pesquisas de aprofundamento, e participam de uma discussão sobre questões específicas sobre o tópico em estudo.

A discussão é realizada durante três dias consecutivos através do serviço Conferências do AulaNet, que funciona como um fórum de discussão que possibilita o encadeamento e categorização de mensagens [Fuks et al., 2002a]. Após a discussão na conferência, o tópico em estudo é encerrado com a realização de um debate síncrono, com duração de uma hora, pela ferramenta de bate-papo Debate do AulaNet.

O papel de transmitir informações e conduzir as argumentações, que geralmente é atribuído aos mediadores de um curso, no TIAE é compartilhado com os aprendizes. Em cada conferência, um aprendiz desempenha o papel de seminarista, elaborando um seminário e três questões a partir das quais a turma desenvolve a argumentação ao longo da semana. Durante este período de argumentação, o seminarista fica responsável por animar e manter a dinâmica da conferência.

Nos debates do curso TIAE, um aprendiz previamente selecionado desempenha o papel de moderador, tornando-se responsável por: coordenar a sessão propondo questões para discussão; manter o foco na questão proposta evitando que a discussão se disperse ou tome rumos inadequados; cuidar para que o debate não ocorra num ritmo muito exagerado ou monótono; e

coordenar os outros aprendizes estimulando a participação de todos. Para apoiar o moderador na coordenação do debate, atuam os mediadores do curso que, dentre outras funções, fazem a abertura e o encerramento formal de cada debate, e avaliam a participação dos aprendizes.

3. A Nova Arquitetura do Ambiente AulaNet

Um sistema colaborativo tem que ser suficientemente flexível para ser adaptado às características do grupo e à evolução dos processos de trabalho. Apesar de não haver uma maneira de prever todos os requisitos de um sistema colaborativo, diversos sistemas compartilham algumas características.

Este cenário é propício à aplicação de técnicas de desenvolvimento baseado em componente, que provêem a flexibilidade necessária em projetos com requisitos instáveis [Szyperski, 1999]. Os serviços de um sistema colaborativo podem ser vistos como componentes de groupware que podem ser plugados e desplugados do sistema [Blois & Becker, 2002]. A arquitetura do sistema é composta de *component frameworks*, que definem as invariantes e os protocolos de conexão e comunicação entre componentes.

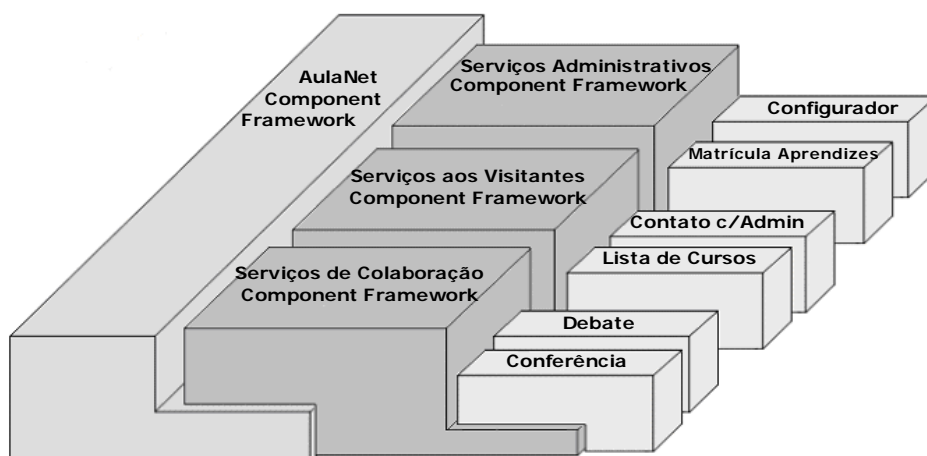


Figura 3. Arquitetura geral do ambiente AulaNet (a título de clareza, apenas alguns serviços foram incluídos).

Na arquitetura do ambiente AulaNet, o *AulaNet component framework* implementa as funcionalidades comuns a todos os serviços, como o gerenciamento da comunicação interserviço e o compartilhamento de dados. Atualmente, há três famílias de serviços no AulaNet: serviços colaborativos, administrativos e de visitantes, que correspondem aos *component frameworks* que lidam com as características específicas de cada serviço (Figura 3). A primeira família de serviço é utilizada pelos docentes e aprendizes para dar suporte às atividades de aprendizagem colaborativa; a segunda é usada pelo administrador do sistema para gerenciar e configurar as funcionalidades e dados do servidor; e a terceira agrega as funcionalidades que são disponibilizadas para visitantes do ambiente, como o FAQ, contanto com administrador, notificação de erro, listagem de cursos, etc.

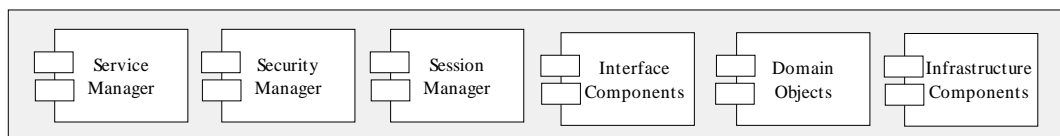


Figura 4. AulaNet component framework.

O *AulaNet component framework* (Figura 4) contém seis componentes principais: *Service Manager*, *Security Manager*, *Session Manager Interface Component Library*, *Domain Object Library* e *Infrastructure Component Library*.

O *Service Manager* é responsável pelo controle de versão e de nome dos serviços, instalação de novos serviços, comunicação entre serviços, localização remota, etc. O *Security Manager* lida com identificação, autenticação e controle de acesso dos usuários. O *Session Manager* mantém persistente um conjunto de valores entre chamadas de um mesmo usuário. A *Interface Component Library* provê os elementos de interface que são utilizados nos serviços, facilitando a atualização, a padronização, a consistência e a alteração de toda a aparência do ambiente. Esta biblioteca também é responsável por gerenciar os textos exibidos na interface, pois o AulaNet possibilita trabalhar com múltiplas linguagens e personalização dos textos. A *Domain Object Library* provê o compartilhamento dos objetos do modelo de domínio entre os serviços, possibilitando seu reuso e a persistência de objetos e ferramentas. Esta biblioteca contém mecanismos de replicação e de controle de concorrência. Para facilitar a implementação dos serviços e para agrupar as funcionalidades de infra-estrutura comuns aos serviços, a arquitetura do AulaNet contém a *Infrastructure Component Library*, que dá suporte ao envio de mensagens, tratamento de erros, entre outros.

Uma estratégia de cliente magro foi adotada na arquitetura do AulaNet, ou seja, a maior parte do processamento acontece no servidor, deixando para o cliente a tarefa de exibir a interface com o usuário. Esta estratégia é especialmente útil em um ambiente de aprendizagem, visto que seus usuários não são necessariamente de áreas relacionadas com a tecnologia, e esta abordagem requer mínima instalação e configuração na estação cliente. Utilizando navegadores Web, os usuários interagem com o AulaNet e seus serviços. Alguns serviços, como o Debate, requerem alguns componentes executando no lado cliente. Nestes casos, a arquitetura provê mecanismos de comunicação cliente-servidor através de *applets* rodando na máquina do cliente.

O J2EE¹ é utilizado como base para a arquitetura do AulaNet, o que viabiliza a integração de componentes e a distribuição em mais de um servidor, melhorando a escalabilidade do ambiente. A conectividade remota é implementada através de tecnologias do J2EE, como JNDI, LMS e RMI. Além disto, o J2EE provê recursos para a autenticação de usuário, controle de permissão e gerenciamento de sessão, servindo como base para os *component frameworks*.

4. Estudo de Caso do Serviço Debate

Os serviços do ambiente AulaNet também são desenvolvidos utilizando uma arquitetura baseada em *component frameworks*, como pode ser observado no exemplo da Figura 5. Há uma estrutura comum implementada pelo *framework de colaboração*, que define o esqueleto do serviço. Plugados a este framework, estão os *component frameworks de comunicação, de coordenação e de cooperação*, que dão suporte à comunicação, coordenação e cooperação, respectivamente. Frameworks OO são utilizados para implementar componentes, que se plugam no respectivo *component framework* e disponibilizam as funcionalidades específicas do serviço.

Na Figura 5 pode ser observada uma versão anterior do serviço Debate do ambiente AulaNet, denominada Mediated Chat 1.0 (MC1). Esta versão é implementada por um componente de comunicação, que cuida da comunicação síncrona com outros usuários, e um componente de cooperação, que implementa o espaço compartilhado.

¹ <http://java.sun.com/j2ee>

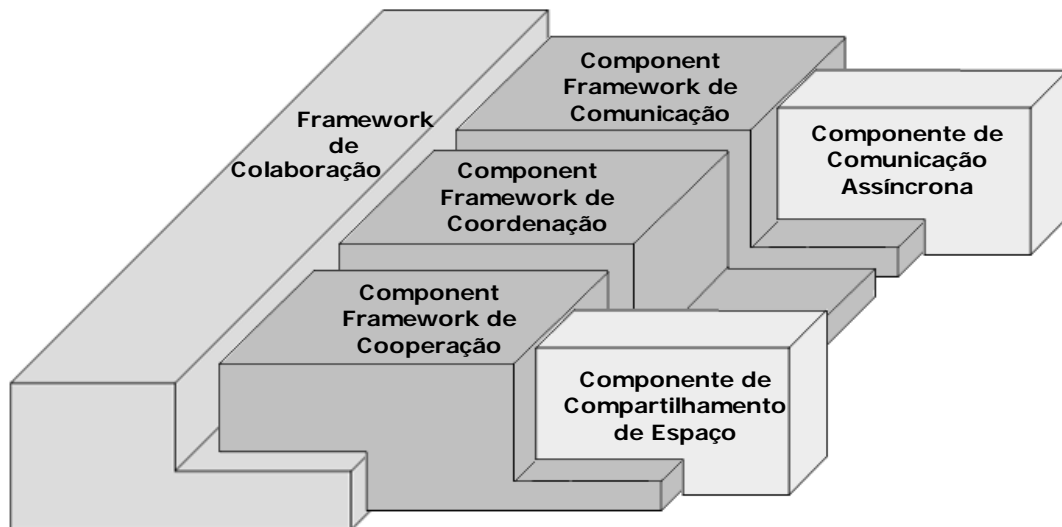


Figura 5. Implementação do serviço colaborativo Debate: Mediated Chat 1.0.

Na Figura 6, a interface do MC1 é apresentada. É uma ferramenta de Chat usual, com uma caixa de texto onde os participantes digitam suas mensagens e elementos de percepção, onde mensagens e a lista de participantes presentes são exibidas.

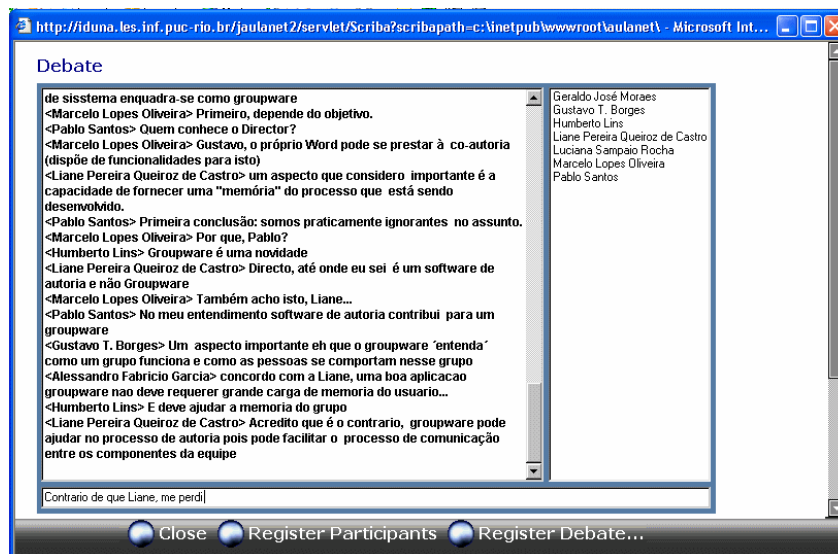


Figura 6. Interface do mediador para o MC1.

Na evolução do curso TIAE, o procedimento adotado na atividade semanal realizada no Debate foi alterada e, infelizmente, a ferramenta MC1 não provia mais um suporte adequado. Anteriormente o assunto era discutido livremente, sendo que quem mantinha o foco e guiava a discussão era o aprendiz moderador do debate. A Figura 7 apresenta o novo fluxo de tarefas da atividade (foi utilizada a notação proposta por [Raposo & Fuks, 2002]).

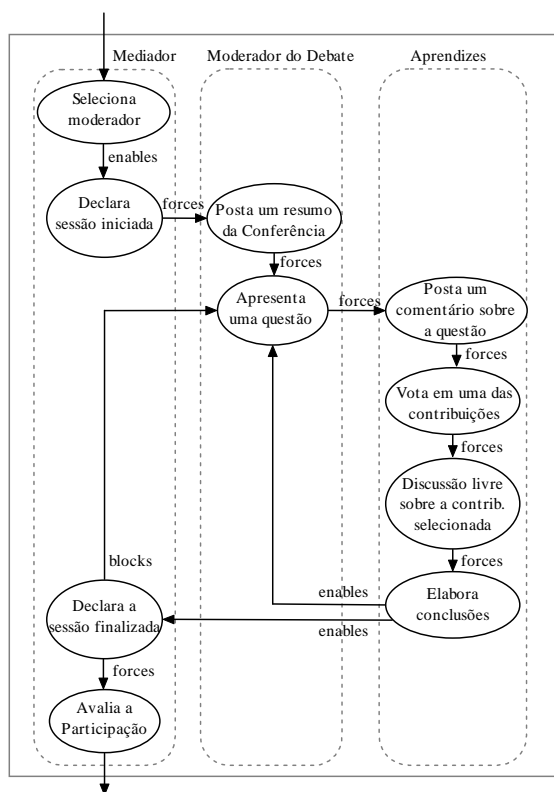


Figura 7. Fluxo de tarefas de um debate.

Na nova dinâmica, o mediador escolhe o moderador e declara a sessão iniciada. O moderador por sua vez é incumbido de postar um resumo do que foi discutido na conferência ao longo da semana e iniciar um loop de apresentar questões e discuti-las, até que o mediador declara a sessão finalizada, para então avaliar a participação dos aprendizes. Para cada questão discutida, há uma rodada onde cada aprendiz posta um comentário sobre a questão e outro onde se vota em qual contribuição será discutida, para então entrar num período de discussão livre.

Porém implementar esta seqüência de atividades no Debate MC1 trazia diversos complicadores que a inviabilizavam [Pimentel, Fuks & Lucena, 2003]. Diversas pessoas teclavam conjuntamente com o moderador, mensagens chegavam após a questão ter sido considerada concluída, havia uma perda de tempo muito grande para organizar as propostas de discussão e a votação, etc. Com isto, sentiu-se a necessidade de incrementar o serviço Debate com novos mecanismos de coordenação que auxiliassem nesta tarefa. Também sentiu-se necessidade de alguns elementos de percepção no espaço compartilhado para indicação do que estava acontecendo em cada momento. Estas necessidades se refletem na coordenação e na cooperação. A parte referente à comunicação do Debate não precisou ser alterada, visto que a troca de mensagens em si permaneceu a mesma.

Implementou-se então os novos componentes para o Debate. O componente de cooperação referente ao espaço compartilhado foi estendido com novos elementos de percepção, como a visualização do tema em discussão, o horário de envio da mensagem, a identificação dos mediadores e da ordem de participação. O componente de coordenação implementado dá suporte ao bloqueio do canal de comunicação e a utilização de técnicas de conversação. A nova arquitetura do serviço pode ser observada na Figura 8.

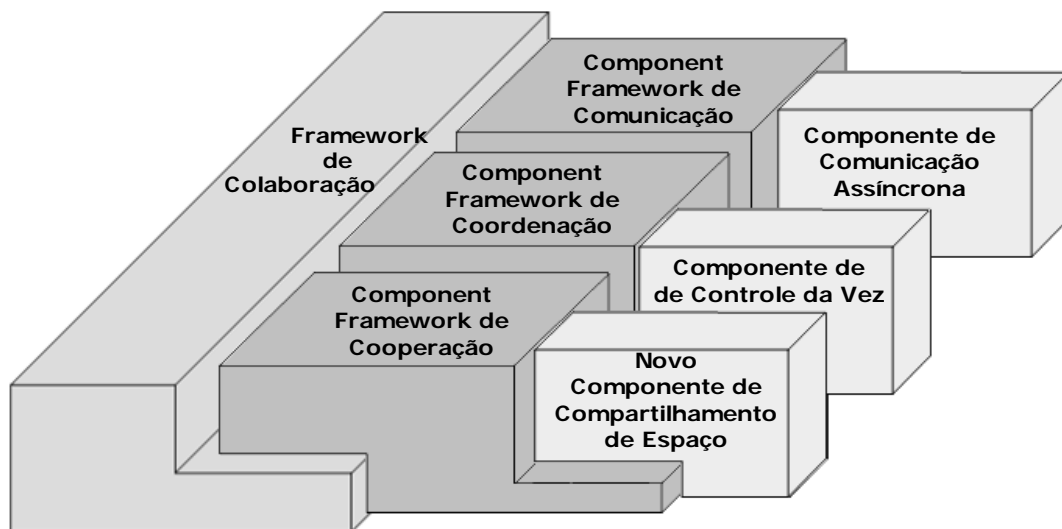


Figura 8. Implementação do serviço colaborativo Debate: Mediated Chat 2.0.

Esta nova versão do serviço Debate foi chamada de Mediated Chat 2.0 (MC2). A interface desta nova versão pode ser observada na Figura 9.

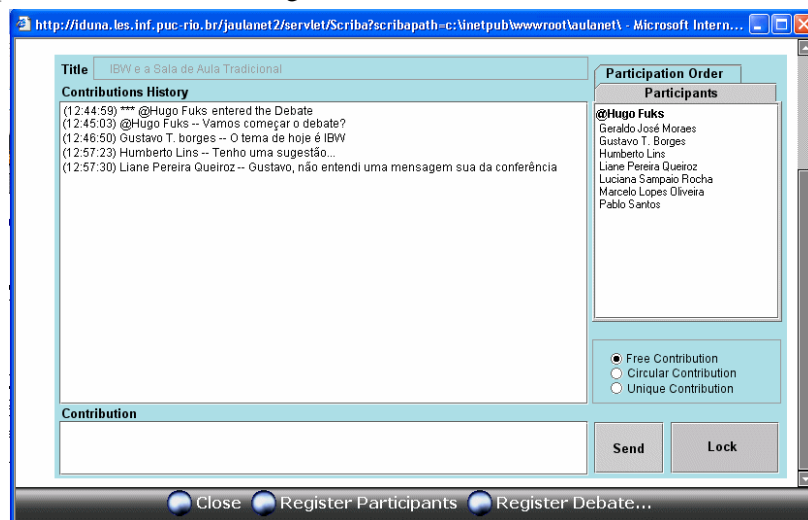


Figura 9. Interface do mediador para o MC2.

As técnicas de conversação implementadas nesta versão do Debate foram: discussão livre, discussão circular e contribuição única [Rezende, Fuks & Lucena, 2003]. Para executar a tarefa de postar um comentário sobre a questão é utilizada a técnica de contribuição circular, deixando o canal de comunicação habilitado para um participante por vez. Para a tarefa de votar em uma contribuição para ser discutida, é utilizada a técnica contribuição única, onde cada um só pode contribuir uma única vez. Nas tarefas do mediador e do moderador do debate, o canal é bloqueado para os demais aprendizes. Nas demais tarefas utiliza-se a discussão livre.

5. Conclusão

O desenvolvimento baseado em componente torna possível a construção de sistemas colaborativos flexíveis. O ambiente AulaNet provê um conjunto padrão de serviços que pode ser adaptado para cada servidor. Isto também torna possível o desenvolvimento de novos serviços,

mesmo sem saber a implementação interna do AulaNet, basta respeitar as interfaces e protocolos de comunicação entre o ambiente e seus serviços. Com isto, é possível encapsular ferramentas que não foram originalmente desenvolvidas para o AulaNet implementando algumas classes que fazem a intermediação da comunicação.

O Modelo de Colaboração 3C define três tipos de serviços que um sistema colaborativo pode dar suporte. Os conceitos abordados no modelo podem ser usados para guiar a especificação funcional e provêem uma linguagem comum para representar e descrever os aspectos da colaboração do grupo. A utilização de *component frameworks* facilita o trabalho dos programadores, que podem reutilizar e estender as estruturas de dados e funcionalidades providas pelos frameworks, deixando para a infra-estrutura dar suporte a aspectos específicos de uma aplicação multi-usuário colaborativa, como a sincronização de dados, compartilhamento de recursos distribuídos, comunicação intercomponentes, etc. O re-projeto do serviço Debate, ilustrado neste artigo, mostrou o mapeamento das necessidades surgidas da evolução da dinâmica do curso, em novos componentes que plugados à arquitetura do ambiente possibilitam flexibilizar e experimentar o suporte à colaboração.

Deve ser lembrado, porém, que como apontado por [Laurillau & Nigay, 2002], apesar de trazer a separação de aspectos do modelo de colaboração para a arquitetura do software é uma boa escolha, fazer o mesmo com 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 eles afetam.

6. Agradecimentos

O Projeto AulaNet é parcialmente financiado pela Fundação Padre Leonel Franca e pelo Ministério da Ciência e Tecnologia através do projeto Sistemas Multi-Agentes para a Engenharia de Software (ESSMA) bolsa nº 552068/2002-0. Também é financiado pelas bolsas individuais do CNPq: Carlos José Pereira de Lucena nº 300031/92-0, Hugo Fuks nº 303055/02-2, Alberto Barbosa Raposo nº 305015/02-8 e Marco Aurélio Gerosa nº 140103/02-3.

Referências

- Blois, A.P.T.B. & Becker, K.A. (2002): Component-based Architecture to Support Collaborative Application Design. *8th International Workshop on Groupware (CRIWG)*. Lecture Notes in Computer Science Vol. 2440. Springer-Verlag, p. 134-146
- Edutools (2004): <http://www.edutools.info> (date 28/09/2004)
- Ellis, C.A., Gibbs, S.J. & Rein, G.L. (1991): Groupware - Some Issues and Experiences. *Communications of The ACM*, vol. 34, no. 1, pp. 38-58.
- Fuks, H. (2000), "Aprendizagem e Trabalho Cooperativo no Ambiente AulaNet", *Revista Brasileira de Informática na Educação*, N6, Abril 2000, ISSN 1414-5685, Sociedade Brasileira de Computação, pp 53-73.
- Fuks, H., Gerosa, M.A. & Lucena, C.J.P. (2002), "The Development and Application of Distance Learning on the Internet", *Open Learning Journal*, V. 17, No. 1, February 2002, Cartafax Pub, pp. 23-38.
- Fuks, H., Gerosa, M.A. & Lucena, C.J.P. (2002a), "Usando a Categorização e Estruturação de Mensagens Textuais em Cursos pelo Ambiente AulaNet", *Revista Brasileira de Informática na Educação*, V. 10, No. 1, Abril 2002, ISSN 1414-5685, Sociedade Brasileira de Computação, pp. 31-44.
- Grudin, J. (1989): Why Groupware Applications Fail: Problems In Design And Evaluation. *Office: Technology And People*, vol. 4, no. 3, pp. 245-264.

- Harasim, L., Hiltz, S. R., Teles, L., & Turoff, M. (1997) "Learning networks: A field guide to teaching and online learning", 3rd ed., MIT Press, 1997.
- Laurillau, Y. & Nigay, L. (2002): Clover architecture for groupware, Proceedings of the ACM Conference on Computer Supported Cooperative Work, Louisiana, USA, p. 236 - 245
- Pimentel, M.G., Fuks, H. & Lucena, C.J.P. (2003), "Debati, debati... aprendi? Investigações sobre o papel educacional das ferramentas de bate-papo" WIE 2003 - IX *Workshop de Informática na Escola*, Anais do XXIII Congresso da Sociedade Brasileira de Computação, V5, Campinas-SP, 2 a 8 de agosto de 2003. pp. 167-178.
- Raposo, A.B. & Fuks, H. (2002): Defining Task Interdependencies and Coordination Mechanisms For Collaborative Systems. In M. Blay-Fornarino, A.M. Pinna-Dery, K. Schmidt and P. Zaraté (eds): *Cooperative Systems Design (Frontiers In Artificial Intelligence and Applications Vol. 74)*. IOS Press, Amsterdam, pp. 88-103.
- Rezende, J.L., Fuks, H. & Lucena, C.J.P. (2003), "Aplicando o Protocolo Social através de Mecanismos de Coordenação embutidos em uma Ferramenta de Bate-Papo". *XIV Simpósio Brasileiro de Informática na Educação - SBIE 2003*, 12 a 14 de Novembro de 2003, ISBN: 85-88442-70-1, Rio de Janeiro - RJ, pp. 55-64.
- Szyperski, C. (1999): *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley.