

FUKS, H., RAPOSO, A. B., GEROSA, M.A., PIMENTEL, M. G., LUCENA, C. J. P. Suporte à Coordenação e à Cooperação em uma Ferramenta de Comunicação Textual Assíncrona: Um Estudo de Caso no Ambiente AulaNet. **I Workshop Brasileiro de Tecnologias para Colaboração - WCSCW 2004**, p.173-180. Ribeirão Preto, SP, Brasil, 2004. (Em conjunto com o WebMedia & La-Web Joint conference 2004).

Suporte à Coordenação e à Cooperação em uma Ferramenta de Comunicação Textual Assíncrona: Um Estudo de Caso no Ambiente AulaNet

Hugo Fuks¹, Alberto Barbosa Raposo², Marco Aurélio Gerosa¹,
Mariano Gomes Pimentel¹, Carlos José Pereira de Lucena¹

¹Laboratório de Engenharia de Software (LES)

²Grupo de Tecnologias em Computação Gráfica (TecGraf)

Depto. de Informática – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

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

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

Abstract

Collaborative systems are especially prone to failure and should be flexible in order to be adapted to the group characteristics and to the work processes evolution. The new architecture of the AulaNet learning environment was developed using component-based techniques to provide the necessary flexibility. A case study regarding the forum service of the environment is presented to show how a tool can be adapted to accompany the evolution of the course dynamics.

Resumo

Sistemas colaborativos são particularmente propícios a falhas e devem ser flexíveis para serem adaptados às características do grupo e à evolução dos processos de trabalho. A nova arquitetura do ambiente AulaNet foi construída baseada em componentes de forma a prover a flexibilidade necessária. Para exemplificar a evolução dos processos de trabalho e o suporte arquitetural dado, é apresentado um estudo de caso com o serviço de fórum do ambiente.

1. Introdução

Sistemas colaborativos são especialmente vulneráveis a falhas [Grudin, 1989]. Eles envolvem aspectos multidisciplinares em sua construção e são difíceis de aplicar e testar. 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, é praticamente impossível para o desenvolvedor prever todas as possibilidades de utilização e dar suporte à maioria delas. Ainda que isto fosse possível, tornaria o sistema excessivamente configurável e, conseqüentemente, difícil de utilizar.

Uma arquitetura componentizada possibilita que o coordenador do grupo acople e desacople componentes de forma a montar uma ferramenta adequada a seus propósitos. Mais do que 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 do grupo e da dinâmica do curso.

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 adaptação do serviço de conferências do ambiente a partir das observações e necessidades extraídas de um dos cursos do ambiente. Na seção seguinte o ambiente AulaNet é apresentado sucintamente, juntamente com o modelo de colaboração 3C, que norteou e foi refinado a partir da evolução do ambiente. Na Seção 3, a arquitetura é apresentada, na Seção 4 o estudo de caso é abordado, a Seção 5 discute a Engenharia de Groupware e a Seção 6 conclui o artigo.

2. O Ambiente AulaNet

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

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.

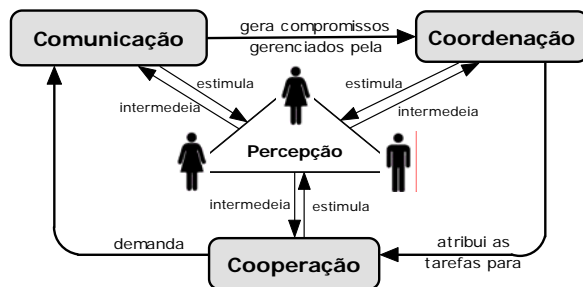


Figura 1. Diagrama do Modelo de Colaboração 3C.

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.

Interligando estas atividades estão as tarefas. A tarefas originam-se dos compromissos e conhecimento negociados durante a comunicação e são realizadas durante a cooperação. A coordenação atua para garantir que as tarefas sejam executadas da maneira correta, no tempo certo e com os recursos necessários. Através da percepção os participantes obtêm feedback e feedthrough e se mantêm cientes do que está acontecendo, tornando possível antecipar as necessidades e auxiliar o trabalho dos colegas [Gerosa, Fuks & Lucena, 2003].



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

Os serviços de colaboração do ambiente AulaNet são organizados em serviços de comunicação, de coordenação e de cooperação, conforme pode ser observado na Figura 2.

3. A Nova Arquitetura do 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 as mesmas 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]. *Component frameworks* definem as invariantes e os protocolos de conexão e comunicação entre os componentes. “Sem uma arquitetura adequada, a construção de groupware e sistemas interativos em geral é difícil de obter, o software resultante é difícil de manter e o refinamento iterativo é dificultado.” [Calvary, Coutaz & Nigay, 1997].

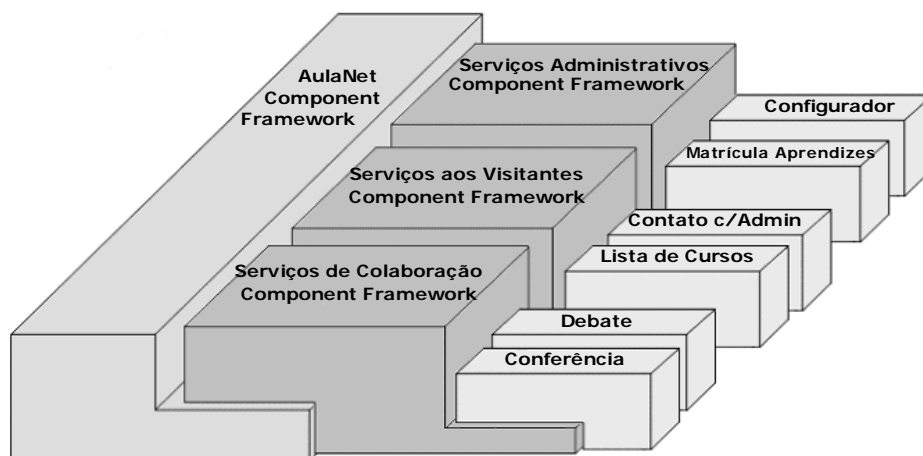


Figura 3 - Arquitetura geral do ambiente AulaNet (a título de clareza, apenas alguns serviços foram incluídos).

Na nova arquitetura do ambiente AulaNet, exibida na Figura 3, 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.

Os serviços colaborativos são utilizados pelos docentes e aprendizes para dar suporte às atividades de aprendizagem colaborativa; os serviços administrativos são usados pelo administrador do sistema para gerenciar e configurar as funcionalidades e dados do servidor; e, por fim, os serviços de visitantes são disponibilizados para visitantes do ambiente, como o FAQ, contato com administrador, notificação de erro, listagem de cursos, etc.

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 infraestrutura 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

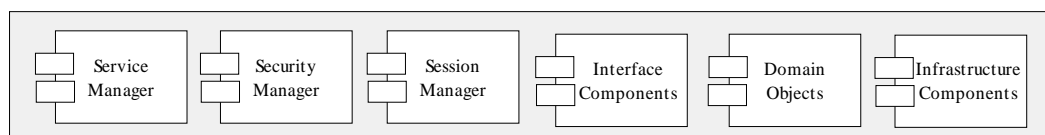


Figura 4 - AulaNet component framework



Figura 5. Interface das Conferências.

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.

O J2EE 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 Conferências

Os serviços do ambiente AulaNet também são desenvolvidos utilizando uma arquitetura baseada em *component frameworks*, como pode ser observado no exemplo da Figura 6. Há uma estrutura comum implementada pelo *collaboration framework*, que define o esqueleto do serviço. Plugados a este framework, estão os *communication*, *coordination* e *cooperation component frameworks*, 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 arquitetura das Conferências, exibida na Figura 6, pode-se notar que foram usados três componentes: um componente de comunicação, um de coordenação e um de cooperação. O Componente de Comunicação

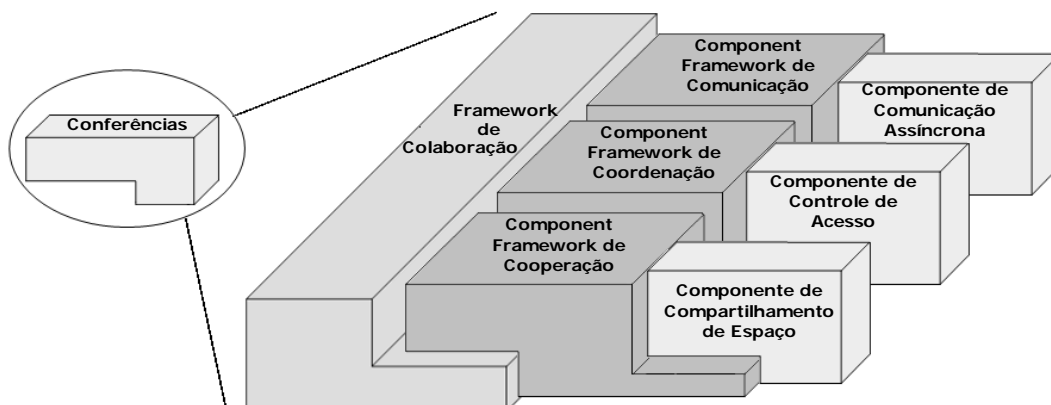


Figura 6. Implementação do serviço colaborativo Conferências.

Assíncrona implementa as facilidades de comunicação, como envio de mensagens, estruturação do discurso, categorização de mensagens, etc. O Componente de Controle de Acesso implementa o trancamento de recursos, e é utilizado na Conferência para que o professor possa desabilitar a conferência, tornando-a somente leitura. O Componente de Compartilhamento de Espaço implementa as funcionalidades de exibição e manipulação dos objetos compartilhados, que no caso das Conferências, são as mensagens.

Na Figura 5, a interface das Conferências é apresentada. Pode-se notar que nas Conferências ocorre a estruturação de mensagens em árvore (as mensagens filhas estão aninhadas com relação à mensagem mãe). A comunicação em si é tratada pelo Componente de Comunicação Assíncrona, enquanto a exibição da estrutura é tratada pelo Componente de Compartilhamento de Espaço. Ambos lidam com as mensagens na forma de árvore.

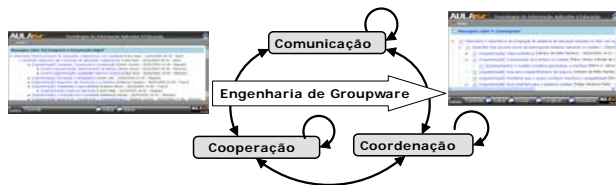


Figura 7. Evolução das Conferências.

Na evolução das dinâmicas dos cursos aplicados pelo ambiente, sentiu-se que a ferramenta necessitava de novos recursos de coordenação e de cooperação. Com base na Engenharia de Groupware [Fuks, Raposo & Gerosa, 2002][Fuks, Gerosa & Pimentel, 2003], foram desenvolvidos novos componentes para prover novas facilidades.

4.1. A Evolução da Coordenação nas Conferências

Uma das necessidades observadas nos cursos do ambiente, principalmente no curso TIAE [Fuks, Gerosa & Lucena, 2002], foi a conceituação das mensagens enviadas pelos aprendizes, visando melhorar a participação qualitativa nas Conferências. O *Componente de Avaliação* foi desenvolvido e acoplado ao *component framework* de coordenação das Conferências de forma a possibilitar: a conceituação individual de cada mensagem e a elaboração e registro de comentários visíveis apenas para o autor, para toda a turma ou apenas para o professor (Figura 8). O conceito dado a cada mensagem passou a ser exibido juntamente com seus outros dados.

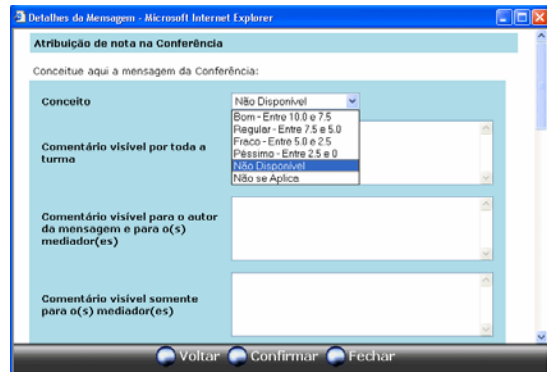


Figura 8. Conceituação de Mensagens nas Conferências.

Também foram desenvolvidos relatórios de acompanhamento da participação (Figura 9) para possibilitar ao professor obter informações de percepção sobre as contribuições dos aprendizes. Estas informações são referentes à qualidade e a quantidade de participações de cada um. Os relatórios também são visíveis para os aprendizes para que eles possam acompanhar seu progresso e compará-lo com o de seus colegas.

Participante	Conferências 8.99 (6)	Debate 8.08 (2)	Conceito Médio 8.57 (10)
Alberto	Bom / 8.68	Ativo(a) / 6.67	8.34
Alexandre	Bom / 9.69	Muito Ativo (a) / 8	7.41
Andre	Bom / 9.56	Muito Ativo (a) / 10	7.74
Andréa	Bom / 8.12	Ativo(a) / 6	7.88
Bernardo	Bom / 9.17	Muito Ativo (a) / 7.86	8.47
Bruno	Bom / 9.73	Muito Ativo (a) / 10	9.64

Figura 9. Acompanhamento da Participação.

4.2. A Evolução da Cooperação nas Conferências

Em alguns cursos do ambiente, o número de mensagens enviadas para uma mesma conferência chega próximo de 100. Isto dificulta a visualização e a navegação pela estrutura de mensagens. Resolveu-se então alterar o Componente de Compartilhamento de Espaço, que cuida da exibição da árvore de mensagens, sem alterar a maneira como os participantes se comunicam, e conseqüentemente, o Componente de Comunicação Assíncrona.

Na nova versão do Componente de Compartilhamento de Espaço, é possível expandir e colapsar as ramificações das árvores de mensagens, conforme pode ser observado na Figura 10.

Além disto, o novo Componente de Compartilhamento de Espaço possibilita marcar várias mensagens para aplicar sobre elas uma mesma operação, fazer buscas por categoria, data, texto ou autor, e exibir estatísticas por categoria, nível, participante, tamanho, data e horário.

5. A Engenharia de Groupware

Idealmente, um groupware deve ser prototipado, dado sua tendência a falha [Grudin, 1989]. Por isto, um ciclo de desenvolvimento incremental mostra-se mais adequado. O desenvolvimento progressivo com sucessivos testes de aceitação e de usabilidade molda o groupware para as reais necessidades do usuário. Como são necessárias várias iterações e as necessidades nunca se estabilizam, pois o grupo muda suas características, seus participantes, seus objetivos, etc., o desenvolvimento baseado em componentes mostrou-se viável para facilitar a moldagem do groupware e a prototipação rápida visando testar conceitos e interações. A Figura 11 mostra o ciclo de desenvolvimento da Engenharia de Groupware proposta.

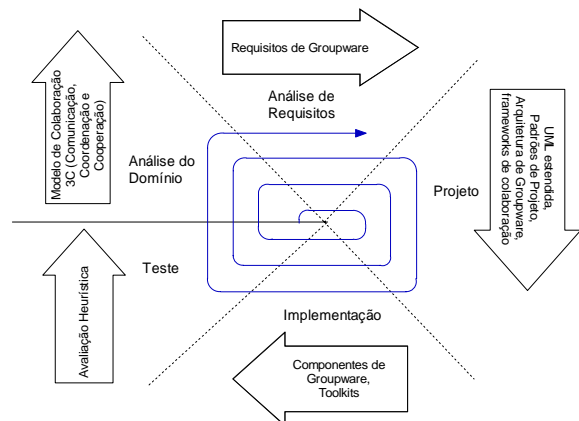


Figura 11. Ciclo da Engenharia de Groupware.

O modelo 3C guia a engenharia de groupware no sentido de poder atacar isoladamente os problemas de comunicação, de coordenação e de cooperação. Tendo suporte nas demais fases do ciclo aos três 3Cs torna-se possível trabalhar com os mesmos conceitos desde os primeiros momentos de análise e até os testes finais. Isto valoriza e facilita o desenvolvimento incremental, onde continuamente se muda de atividade. Mantendo a mesma notação e conceitos na análise do domínio, na análise de requisitos, no projeto, na implementação e nos testes cria-se um vocabulário comum, diminui o gap semântico, trazendo conceitos e termos da realidade para a implementação, e suaviza a transição entre as fases.

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], percebe-se que o modelo 3C instrumenta o desenvolvedor, que pode analisar a aplicação de sua ferramenta com base nos conceitos e teorias referentes à comunicação, à coordenação e à cooperação.

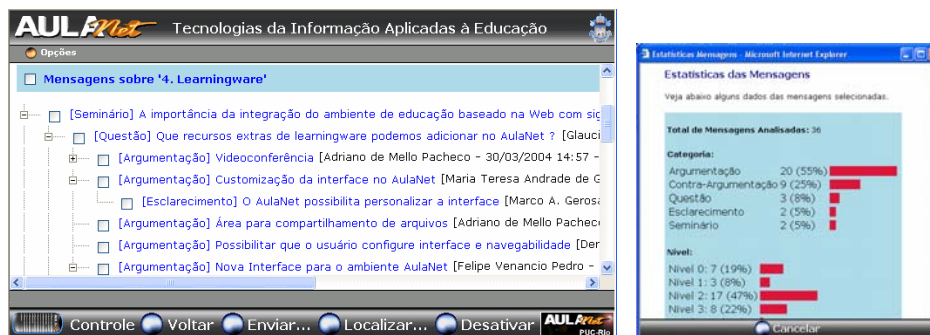


Figura 10. Novo espaço compartilhado das Conferências.

O levantamento de requisitos pode ser guiado por requisitos genéricos de groupware [Schmidt & Rodden, 1996] e pelos requisitos de colaboração levantados durante a análise baseada no modelo 3C. Porém vale lembrar que os requisitos 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 também ajuda a entender o desenvolvimento futuro.

Há diferentes técnicas baseadas no modelo 3C que podem instrumentar o projeto de um groupware. Por exemplo, padrões de projeto de groupware [Groupware Patterns Swiki, 2003], para reusar abordagens comuns de projeto; extensões da UML, para representar aspectos específicos de groupware; arquiteturas de groupware e frameworks de colaboração [Tietze, 2001]. Na seção anterior ilustramos o uso de frameworks e componentes baseados no modelo 3C que facilitam a tarefa do programador, que pode reusar dados, funcionalidades e estruturas providas pelos frameworks, deixando para a infra-estrutura o suporte de alguns aspectos recorrentes em aplicações multi-usuário, como sincronização de dados, distribuição e gerenciamento de recursos e comunicação intercomponentes.

Para a fase de implementação, componentes pré-elaborados de comunicação, de coordenação e de cooperação podem ser plugados nos component frameworks, como mostrado no estudo de caso. Toolkits [Roseman & Greenberg, 1997] e outros tipos de componentes [Blois & Becker, 2002] são alternativas para o desenvolvimento de aplicações colaborativas.

Dado sua natureza interativa complexa, teste de groupware ainda não alcançou a sua maturidade. Os requisitos e conceitos do modelo 3C podem ajudar a validar e testar heurísticamente o software, focando a atenção do avaliador a aspectos específicos de comunicação, de coordenação e de cooperação.

6. 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.

Os refinamentos do serviço Conferências, ilustrados neste artigo, mostrou o mapeamento das necessidades surgidas da evolução da dinâmica dos cursos do ambiente, 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 e não funcionalmente separados.

7. 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, Marco Aurélio Gerosa nº 140103/02-3 e Alberto Raposo nº 305015/02-8. Mariano Gomes Pimentel recebe bolsa da CAPES.

8. 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

- Calvary, G., Coutaz, J., Nigay, L. (1997): From Single-User Architectural Design to PAC*: a Generic Software Architectural Model for CSCW. *Proceedings of CHI'97*, Atlanta, March 1997, pp 242-249.
- 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., 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, ISSN 0268-0513, Cartafax Pub., pp. 23-38.
- Fuks, H., Gerosa, M.A. & Pimental, M.G. (2004), "Projeto de Comunicação em Groupware: Desenvolvimento, Interface e Utilização", *XXII Jornada de Atualização em Informática*, Anais do XXIII Congresso da Sociedade Brasileira de Computação, V2, Cap. 7, ISBN 85-88442-59-0, pp. 295-338, 2003.
- Fuks, H., Raposo, A.B. & Gerosa, M.A. (2002), "Engenharia de Groupware: Desenvolvimento de Aplicações Colaborativas", *XXI Jornada de Atualização em Informática*, Anais do XXII Congresso da Sociedade Brasileira de Computação, V2, Cap. 3, ISBN 85-88442-24-8, pp. 89-128.
- Gerosa, M.A., Fuks, H. & Lucena, C.J.P. (2003): Analysis and Design of Awareness Elements in Collaboration Digital Environments: A Case Study in the AulaNet Learning Environment. *The Journal of Interactive Learning Research*, V. 14, No. 3, AACE, USA, pp. 315-332.
- Grudin, J. (1989): Why Groupware Applications Fail: Problems In Design And Evaluation. *Office: Technology And People*, vol. 4, no. 3, pp. 245-264.
- Groupware Patterns Swiki (2003), <http://swiki.darmstadt.gmd.de/gw-patterns>
- 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.
- Hess, H. et al. (2000): "A Domain Analysis Bibliography", Carnegie Mellon University/Software Engineering Institute, <http://www.sei.cmu.edu/publications/documents/90-reports/90.sr.003.html>, Special Report -90-SR-3.
- Laurillau, Y. & Nigay, L. (2002): Clover architecture for groupware, Proceedings of the ACM Conference on Computer Supported Cooperative Work, Louisiana, USA, p. 236 - 245
- Roseman, M. & Greenberg, S. (1997): Building Groupware with GroupKit, Tcl/Tk Tools, Harrison, M. (ed), Chapter 15, pp 535-564, O'Reilly Press.
- Schmidt, K. & Rodden, T. (1996): Putting it all Together: Requirements for a CSCW Platform. In: Shapiro, D., Tauber, M., Traummüller, R. (eds.): The Design of Computer Supported Cooperative Work and Groupware Systems. North Holland, Holland, p. 157-176
- Schmidt and P. Zaraté (2002): *Cooperative Systems Design (Frontiers In Artificial Intelligence and Applications Vol. 74)*. IOS Press, Amsterdam, pp. 88-103.
- Szyperski, C. (1999): Component Software: Beyond Object-Oriented Programming. Addison-Wesley.
- Tietze, D.A. (2001): A Framework For Developing Component-Based Co-Operative Applications. Ph.D. Dissertation, Computer Science, Technischen Universität Darmstadt, Germany.