

Um groupware baseado no ambiente AulaNet desenvolvido com componentes

Marco Aurélio Gerosa, Leonardo Magela Cunha, Hugo Fuks & Carlos J. P. Lucena
[gerosa, leocunha, hugo, lucena]@inf.puc-rio.br

Laboratório de Engenharia de Software – LES, Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

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

RESUMO

Na sociedade da informação as pessoas têm que aprender a trabalhar cooperativamente com as novas tecnologias desenvolvidas. O ambiente AulaNet é uma proposta para este tipo de aprendizagem. Mas como seria um ambiente para o trabalho cooperativo? Neste artigo é proposto o projeto do eLabora, um ambiente para o gerenciamento do trabalho em grupo baseado no AulaNet, a ser desenvolvido numa abordagem de framework baseado em componentes.

Palavras-chaves: Componentes de Software, *groupware*, *learningware*.

ABSTRACT

In an information society, people must learn to work in groups with new communication and information technologies. With the AulaNet environment, developed using a prototyping approach, it is possible to carry out this type of cooperative learning. In this article, it is proposed the eLabora project, a component-based framework approach for the management of work groups based upon the AulaNet environment experience.

Keywords: Software components, groupware, learningware.

1. Introdução

O AulaNet [Lucena99] é um ambiente para a criação, aplicação e administração de cursos baseados na Web. Seu desenvolvimento vem se realizando desde junho de 1997 no Laboratório de Engenharia de Software (LES) da PUC-Rio (Pontifícia Universidade Católica do Rio de Janeiro). Ele se baseia nas relações de trabalho cooperativo que se manifestam nas interações dos aprendizes com seus instrutores, com outros aprendizes e com os conteúdos didáticos. O AulaNet adota uma abordagem de *groupware* [Coleman95], ou seja, ele é um sistema para suporte ao trabalho em grupo. Com ele o docente pode produzir os conteúdos através de suas ferramentas de trabalho, como o processador de texto, ou reutilizar conteúdos já existentes em formato digital, deixando a gerência e implementação da navegação para o ambiente. Além disso, também são oferecidos diversos mecanismos de comunicação, coordenação e cooperação que podem ser utilizados nos cursos de forma o torna-los mais interativos e estruturados.

Os serviços oferecidos pelo AulaNet são organizados de acordo com o princípio que para aprender em grupo, um indivíduo tem que compartilhar idéias (se comunicar), estar em

sintonia com os membros do grupo (se coordenar), e realizar as tarefas satisfatoriamente (cooperar) [Fuks99]. Estes serviços são colocados à disposição do docente na criação do curso e durante sua aplicação, possibilitando a seleção e configuração de quais serviços se tornarão disponíveis aos aprendizes.

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 ser aptos a aprender continuamente, a trabalhar em grupo e a transformar de forma criativa conhecimento em novo conhecimento. Essas necessidades criam uma situação onde seus ambientes de trabalho e de aprendizado se confundem e se misturam. Percebendo isso e sentindo a necessidade de re-projetar o ambiente AulaNet, propõe-se neste artigo a construção de um novo ambiente que integre as necessidades do ensino-aprendizagem e do trabalho. Desta forma, espera-se capacitar trabalhadores para enfrentar os desafios da sociedade do conhecimento e para que possam trabalhar num ambiente similar ao que são treinados.

Este novo ambiente, a ser denominado eLabora, será projetado de forma a facilitar a instanciação de outros groupware, como o AulaNet, e a possibilitar o trabalho de equipes externas em novos serviços. Para isso este novo ambiente será desenvolvido usando uma abordagem de framework baseado em componentes. Esta abordagem possibilitará um maior reuso de software, mais rapidez na integração de novos serviços ao ambiente e a possibilidade de equipes externas desenvolverem novas funcionalidades [Szyperski99].

2. A arquitetura AulaNet 2.0

Todos os mecanismos do AulaNet são classificados em mecanismos de comunicação, coordenação e cooperação. Os mecanismos de comunicação fornecem as facilidades que tornam possível a troca e o envio de informações, como discussão textual assíncrona no estilo de fórum (**Grupo de Interesse**); conferência síncrona textual no estilo de *chat* (**Debate**); troca instantânea de mensagens (**Contato com os Participantes**); e correio eletrônico com o instrutor (**Contato com os Docentes**) e com a turma (**Grupo de Discussão**). Os mecanismos de coordenação fornecem os meios para a gerência do tempo e da competência do grupo, incluindo ferramentas de notificação (**Avisos**); de coordenação básica do fluxo do curso (**Plano de Aulas**); de avaliação (**Tarefas e Exames**); e de acompanhamento (**Acompanhamento da Participação**). Os mecanismos de cooperação fornecem os meios para a aprendizagem cooperativa [Harasim97] e de co-autoria (**Co-autoria de Docente e Co-autoria de Aprendiz**), além de uma lista de referências do curso (**Bibliografia e Webliografia**) e de conteúdos transferíveis (**Download**).

A versão 2.0 do AulaNet utiliza uma arquitetura arquitetura cliente–servidor na World Wide Web baseada em servlets. Os servlets são classes Java instanciadas pelo servidor Web para receber as requisições dos usuários, tratá-las e gerar dinamicamente as páginas de resposta [Siyan97]. O AulaNet foi desenvolvido com base na tecnologia Scriba [Blois99], que através de um servlet intermedia toda a comunicação do cliente com o servidor. O Scriba oferece uma linguagem própria a ser embutida nos arquivos HTML que permite, entre outras facilidades, acesso a banco de dados, definição de variáveis para armazenamento temporário de dados e chamadas a classes implementadas em Java. As classes Java agrupam as funções mais específicas e complexas da aplicação.

O servidor AulaNet é composto de um módulo central, que interage com o navegador do cliente através da Internet e de módulos complementares que realizam funções específicas. O módulo central do AulaNet é formado por um conjunto de classes Java, páginas HTML com código Scriba embutido e um banco de dados ODBC. As páginas também possuem código Javascript [Goodman01] e HTML dinâmico [Goodman98] para autenticação de campos e definição da interface com o usuário. Todos os dados manipulados pelo módulo central são armazenados em um banco de dados relacional. A comunicação entre as classes e o banco de dados é realizada através da ponte JDBC-ODBC, permitindo o uso de banco de dados compatíveis com o padrão SQL e ODBC.

3. Por que mudar?

O Scriba foi especialmente criado para ser a tecnologia de base para a construção do ambiente AulaNet. Os comandos existentes permitem a montagem de páginas a partir de valores armazenados em bancos de dados, a definição de variáveis e chamadas a classes Java. Algumas vezes são necessários outros comandos que já existem em tecnologias similares, como o JSP [JSP01]. Quando o Scriba foi desenvolvido, a Sun Microsystems ainda não havia distribuído esta tecnologia, que apesar de ser similar ao Scriba, tem outras funcionalidades interessantes [Blois00]. Apesar da possibilidade de extensão da linguagem Scriba, isso desviaria o foco do desenvolvimento. Além disso, a utilização de uma tecnologia proprietária pode, muitas vezes, dificultar a integração de novos membros à equipe.

Outro problema notado com a utilização do Scriba foi a disseminação de código por arquivos HTML e classes Java. Na arquitetura atual do AulaNet os arquivos HTML funcionam como moldes, com comandos Scriba e JavaScript, que geram dinamicamente as páginas, recorrendo a classes implementadas em Java para as funções mais complexas. Essa disseminação de código entre as diferentes linguagens e arquivos dificulta a manutenção do ambiente. Quando um problema é identificado, é necessário verificar se o mesmo ocorre no arquivo molde, checando as tags HTML, as instruções JavaScript e o código do Scriba. Não sendo este o caso, é necessário verificar o problema nas classes Java chamadas pelo arquivo molde e nas classes chamadas pelas próprias classes.

As classes Java implementam as funções específicas de cada arquivo molde. Desta forma, apesar de terem sido desenvolvidas utilizando uma linguagem orientada a objetos, estas classes se comportam com uma biblioteca de funções, o que se assemelha ao paradigma funcional [Cousineau98]. O acesso aos dados do sistema é espalhado pelo código. De forma que quando ocorre alguma alteração no banco de dados, como por exemplo a inclusão de um campo em uma das tabelas, é necessário buscar e atualizar todas as classes e arquivos HTML onde esta tabela é usada. O paradigma de orientação a objetos resolveria esta dificuldade, encapsulando os dados na forma de objetos [Pressman00].

A versão atual do AulaNet é voltada para educação. Pretende-se que o ambiente acomode diferentes conceitos de *groupware* e não somente os educacionais. Outras propostas de serviços estão sendo elaboradas, como coordenação baseada em *workflow*, utilização de conteúdos educacionais distribuídos [Torres01] e utilização de agentes de software.

O AulaNet vem sendo desenvolvido por prototipação. Seus desenvolvedores são alunos do doutorado, mestrado e graduação da PUC-Rio que, além de mantê-lo, usam-no em suas teses, dissertações e monografias, implementando e testando os conceitos de seus trabalhos. Com isso o AulaNet cresceu e suas funcionalidades foram implementadas na medida da necessidade, e agora tornou-se necessária uma reestruturação do projeto.

4. O eLabora

Visando solucionar os problemas da arquitetura atual do ambiente AulaNet será desenvolvido um novo groupware, o ambiente eLabora, que a partir de uma configuração adequada de seus componentes, instanciará o AulaNet. Para facilitar esta instanciação, e outras que possam surgir, o eLabora será desenvolvido numa abordagem de framework baseado em componentes.

Segundo [Szypersky99] um *component framework* é uma entidade de software que suporta componentes de acordo com certos padrões e permite que instâncias desses componentes sejam conectadas a ele. Ele estabelece “condições ambientais” para as instâncias dos componentes e regula as interações entre elas. *Component frameworks* podem ser utilizados sozinhos, podem criar uma “ilha” para certos componentes, ou ainda podem cooperar com outros componentes ou *component frameworks*. Logo, é natural modelar *component frameworks* como componentes. A arquitetura do eLabora será então constituída de um *component framework* formado por um conjunto integrado de *component frameworks* de comunicação, coordenação e cooperação.

Este novo ambiente tem como proposta inicial para o seu projeto, a utilização de um *design pattern* (*Component-Based Design Pattern*) refletido pela tecnologia de componentes [Bachman00] apresentado na Figura 1.

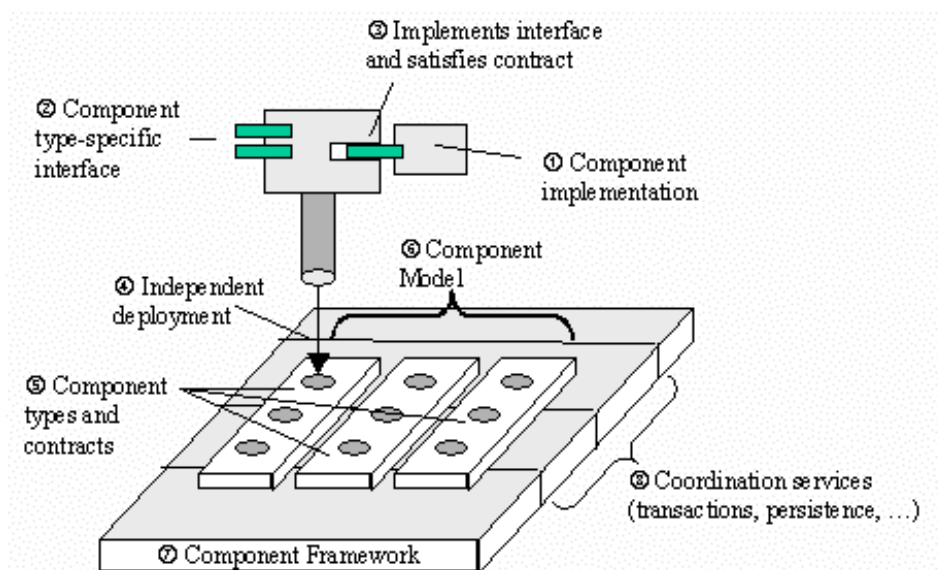


Figura 1 – Component-Based Design Pattern [Bachman00]

Neste *design pattern*, um componente (1) é uma implementação de Software que pode ser executada em um dispositivo lógico ou físico. Um componente implementa uma ou mais interfaces que são “impostas” a ele (2). Isso implica que o componente satisfaça algumas obrigações, que mais tarde serão descritas com um contrato (3). Essas obrigações contratuais garantem que componentes desenvolvidos independentemente possam interagir ou não, e que possam ser conectados em ambientes em tempo de execução ou em tempo de projeto (4). Um sistema baseado em componentes é composto de um pequeno número de *component types*, cada um com seu papel especializado no sistema (5), descrito por uma interface (2). Um *component model* (6) é o conjunto de *component types*, suas interfaces, e uma especificação de padrões de interação permitidos entre os *component types*. Um *component framework* (7) fornece uma variedade de serviços de *runtime* (8) para apoiar e reforçar o *component model*.

Algumas vezes, os *component frameworks* são como sistemas operacionais com propósitos especiais, apesar deles operarem em um nível muito mais alto de abstração [Bachman00].

5. Considerações finais

Como foi exposto anteriormente, o AulaNet foi desenvolvido buscando o aprimoramento do aprendizado em grupo, e seus princípios podem ser reutilizados num software para trabalho em grupo. Esse *groupware* seria o ambiente eLabora, que teria um propósito mais geral do que o AulaNet. Com algumas adaptações, principalmente na nomenclatura, é possível imaginar que um curso no AulaNet seria correspondente a um projeto no eLabora, uma turma a uma equipe, uma aula a uma etapa de trabalho, etc. Da mesma forma que podemos “generalizar”, podemos perceber que se tivermos o eLabora implementado, o caminho inverso também é possível, ou seja, poderemos instanciar o AulaNet ou diversos outros *groupware* a partir do eLabora.

A arquitetura do AulaNet está baseada numa tecnologia proprietária, que promove a disseminação do código. Além disso, foi desenvolvido por prototipação, levando a diversas características não planejadas durante o projeto. Algumas adaptações foram necessárias na sua arquitetura e modelagem e, com o passar do tempo, surgiu a necessidade do seu re-projeto, que será feito no eLabora.

O projeto do eLabora pode ser influenciado por tecnologias como o desenvolvimento baseado em componentes, uma vez que a extensibilidade do ambiente [Bachman00] ocorre e é estimulada. O reuso de componentes pode ser justificado pela necessidade de implementar outros *groupware* com abordagens semelhantes, como o AulaNet. A colaboração de equipes externas, através do desenvolvimento de componentes para o ambiente, será estimulada e favorecerá a utilização de tecnologias de ponta [Szyperski99]. Por fim, o código ficará mais modularizado e a quantidade de tempo gasto em sua manutenção será reduzida [Bachman00], possibilitando assim a pesquisa de novos conceitos.

Com equipes de outras instituições e empresas podendo desenvolver novos componentes para o eLabora, que poderão ser acoplados tanto como novos serviços nos projetos/cursos quanto novos mecanismos de administração, a equipe AulaNet poderá focalizar seus esforços no *component model*, no eLabora framework e na integração dos diversos componentes. Além disso, instituições que forem usar o eLabora terão subsídios para adequá-lo e incrementá-lo de acordo com suas necessidades. A qualidade do ambiente aumentará, visto que equipes especializadas poderão atuar nos diferentes componentes e módulos do ambiente.

Até abril de 2001 foram distribuídos mais de 4100 instaladores do ambiente AulaNet. Ao propor um novo ambiente de trabalho que seja similar ao ambiente de aprendizagem das pessoas, pretende-se incentivar a convergência de seus ambientes de trabalho e de aprendizagem. Esse novo ambiente pode atender as necessidades de aprendizagem continuada e auxiliar na capacitação dos trabalhadores modernos.

6. Agradecimentos

O projeto AulaNet é parcialmente financiado pela Fundação Padre Leonel Franca, pelo Ministério da Ciência e Tecnologia através de seu Programa de Núcleos de Excelência (PRONEX) bolsa nº 76.97.1029.00 (3366) e também através de bolsas individuais do Conselho Nacional de Pesquisa: Carlos José Pereira de Lucena nº 300031/92-0, Hugo Fuks nº 524557/96-9, Leonardo Magela Cunha nº 131877/2000-3. Marco Aurélio Gerosa recebeu

bolsa individual do Conselho de Aperfeiçoamento do Ensino Superior do Ministério da Educação.

7. Referências

- [Bachman00] Bachman, F. et alli; 2000; Volume II: Technical Concepts of Component-Based Software Engineering, Internal Research and Development; Carnegie Mellon University; <<http://www.sei.cmu.edu/publications/documents/00.reports/00tr008/00tr008title.html>> (Consulta em 16/04/2001).
- [Blois99] Blois, M., Choren, R., Laufer, C., Ferraz, F., & Fuks, H.; 1999; Desenvolvendo Aplicativos para a Web com o Scriba; Anais do XXVI SEMISH - Seminário Integrado de Software e Hardware, Sociedade Brasileira de Computação (SBC), pp 119-133, Rio de Janeiro.;
- [Blois00] Blois, M, Fuks, H., Lucena, C.J.P.; 2000; The E-Society Groupware Approach; SSGRR'2000 - Computer & eBusiness Conference, 2000; L' Aquila - Roma, Itália.
- [Coleman95] Coleman, D., & Khanna, R.; 1995; Groupware: Technology and applications; EUA; Prentice Hall, Inc.
- [Cousineau98] Cousineau, G. & Mauny, M. The Functional Approach to Programming. Cambridge University Press, English edition, December 1998. ISBN: 0521576814.
- [Fuks99] Fuks, H., Laufer, C., Choren, R., & Blois, M.; 1999; Communication, Coordination and Cooperation in Distance Education. In Proceedings of V AMCIS'99, Association for Information Systems (AIS). Milwaukee, USA. pp. 130-132.
- [Goodman98] Goodman, D. Dynamic Html: The Definitive Reference. USA: O'Reilly & Associates, 1998. ISBN: 1565924940.
- [Goodman01] Goodman, D. & Eich, B. Javascript Bible. USA: Hungry Minds, Inc, 2001. ISBN: 0764533428.
- [Harasim97] Harasim, L., Hiltz, S. R., Teles, L., & Turoff, M.; 1997; Learning networks: A field guide to teaching and online learning (3rd ed.); EUA; MIT Press.
- [JSP01] JSP Technology; 2001; <<http://www.java.sun.com/products/jsp>> (Consulta em 16/04/2001).
- [Lucena99] Lucena, C. J. P., Fuks, H., Milidiú, Laufer, C., Blois, M., Choren, R., Torres, V. & Daflon, L.; 1999; AulaNet: Helping Teachers to Do Their Homework; Multimedia Computer Techniques in Engineering Education Workshop; Graz, Austria.
- [Pressman00] Pressman, R. Software Engineering: A Practitioner's Approach. 5th Edition. McGraw-Hill Higher Education, 2000. ISBN: 0073655783.
- [Siyan97] SIYAN, K. S. & WEAVER, J. L. Inside Java. New Riders Publishing, 1997.
- [Szyperski99] Szyperski, C. Component Software – Beyond Object-Oriented Programming. New York: ACM Press, 1999. ISBN 0201178885.
- [Torres01] Torres, V., Lucena, C. J. P. & Fuks, H. ContentNet: A Framework for the Interoperability of Educational Content Using Standard IMS. Computers & Education Journal. (submitted)