

# Evoluindo para uma Arquitetura de Groupware Baseada em Componentes: o Estudo de Caso do Learningware AulaNet

Hugo Fuks<sup>1</sup>, Marco Aurélio Gerosa<sup>1</sup>, Mariano Gomes Pimentel<sup>1</sup>, Alberto Barbosa Raposo<sup>2</sup>, Luís Henrique Raja Gabaglia Mitchell<sup>1</sup> & Carlos José Pereira de Lucena<sup>1</sup>

<sup>1</sup> Lab. de Engenharia de Software (LES), Departamento de Informática

<sup>2</sup> Lab. de Tecnologia 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

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

**Abstract.** *Groupware must be flexible in order to satisfy the different characteristics and the continuous evolution of the workgroups and processes where it is used. One way of achieving such flexibility is to use component-based development techniques. This article proposes a new J2EE-based architecture to be used as a basis for building different types of groupware. It ensues in the new AulaNet architecture that will guide the construction of a groupware generator. The current AulaNet architecture, the difficulties identified in maintaining it and some of the reasons that led to the proposal of a new architecture are also presented.*

**Resumo.** *Groupware deve ser flexível para atender às diversas características dos grupos e processos de trabalho onde são utilizados. Uma forma de atingir esta flexibilidade é utilizar técnicas de desenvolvimento baseado em componentes. Neste artigo é proposta uma nova arquitetura baseada em J2EE para o ambiente AulaNet, um ambiente de ensino-aprendizagem na Web. Pretende-se utilizar esta arquitetura como base para a construção de groupware de diferentes propósitos. A arquitetura atual do ambiente e seus problemas e características que levaram a necessidade de uma nova arquitetura também são abordados.*

**Palavras chaves:** *Arquitetura de software baseada em componentes, groupware, componentes para Web*

## 1. Introdução

Uma crescente parte do trabalho das empresas e instituições deixou de ser feita individualmente, com uma pessoa trabalhando sozinha até completar as tarefas. O trabalho é cada vez mais realizado de forma colaborativa. Esta tendência se deve parcialmente ao aumento de complexidade das tarefas, que passaram a requerer habilidades multidisciplinares, e aos novos padrões de trabalho, que envolvem diversos setores da empresa, ou até mesmo outras empresas, trabalhando conjuntamente nas diversas fases de elaboração de um produto ou no desenvolvimento de um projeto.

Desenvolver ambientes que dêem suporte a este trabalho colaborativo é uma tarefa complexa, que requer domínio de diversas áreas de conhecimento, como informática, psicologia, pedagogia, sociologia, cognição, etc. Estes ambientes são bastante susceptíveis a falhas, visto que os processos de trabalho entre os indivíduos são muito específicos e evoluem com o tempo e a adoção de um groupware requer mudanças de atitudes e maneiras de trabalhar [Grudin, 1989]. Os desenvolvedores de aplicações colaborativas necessitam de ferramentas que os auxiliem a desenvolver aplicações flexíveis o suficiente para serem adaptadas às características do grupo e à evolução dos processos de trabalho em questão, pois

não há como, a princípio, antever todas as demandas para um groupware. Porém, mesmo groupware diferentes compartilham diversas características. O Laboratório de Engenharia de Software (LES) da Universidade Católica do Rio de Janeiro (PUC-Rio) objetiva construir um gerador de groupware baseado em *frameworks* que possibilite aos desenvolvedores compor ambientes específicos para cada caso.

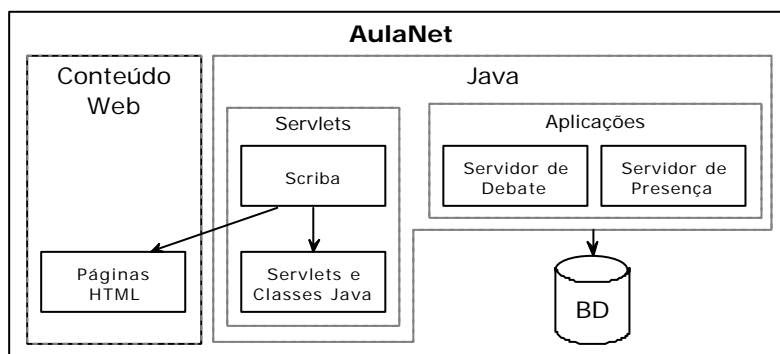
O LES já desenvolve, desde junho de 1997, um groupware gratuito<sup>1</sup> voltado para educação denominado AulaNet. Para chegar a uma arquitetura que atinja a flexibilidade requerida pelo gerador de groupware, começou-se pela remodelagem do ambiente AulaNet. Com isto, pretende-se desenvolver este gerador por prototipação, começando pela nova arquitetura do ambiente AulaNet até futuramente poder-se instanciar o próprio AulaNet.

Neste artigo, apresenta-se a nova arquitetura do ambiente AulaNet. Ela foi projetada visando flexibilidade e escalabilidade. Optou-se por utilizar técnicas de desenvolvimento baseado em componentes, que podem ser executados de forma distribuída. Como suporte a esta arquitetura, será utilizado um servidor de aplicação J2EE. Na Seção 2 deste documento a arquitetura atual do ambiente e seus problemas são descritos, enquanto na Seção 3 a nova arquitetura é explicada.

## 2. O Ambiente AulaNet

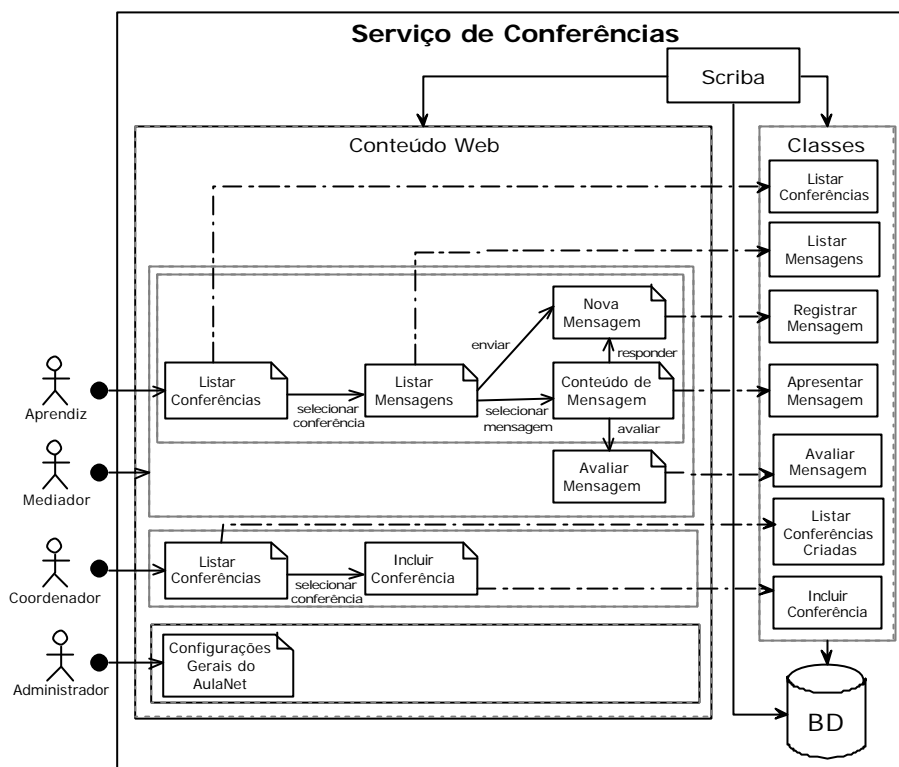
O AulaNet é um ambiente baseado numa abordagem groupware para o ensino-aprendizagem na Web. Os serviços do AulaNet são classificados em serviços de comunicação, coordenação e cooperação. O docente pode selecionar os serviços que deseja para compor seu curso.

A arquitetura do AulaNet 2.0, mostrada na Figura 1, funciona num modelo cliente-servidor na Web. Esta arquitetura foi desenvolvida tendo como base a tecnologia Scriba [Blois, 1999], que é um servlet para intermediar toda a comunicação do cliente com o servidor. O Scriba oferece uma linguagem própria a ser embutida nos arquivos HTML para possibilitar, entre outras facilidades, chamadas a classes implementadas em Java, acesso a banco de dados e definição de variáveis para armazenamento temporário de dados. Além do servidor Web, no servidor AulaNet também rodam aplicações Java que implementam os serviços de comunicação síncrona (que não fazem uso do protocolo HTTP), acessados pelos clientes através de applets executados no navegador. Os outros serviços do AulaNet são implementados por páginas HTML com código Scriba embutido, classes Java e uma base de dados cujo acesso se faz através de uma ponte JDBC-ODBC. A Figura 2 esquematiza como um serviço (Conferências) foi projetado na arquitetura do AulaNet 2.0.



**Figura 1. Arquitetura do servidor AulaNet 2.0**

<sup>1</sup> <http://www.eduweb.com.br>, <http://www.les.inf.puc-rio.br/groupware>



**Figura 2. Projeto do serviço de Conferências na arquitetura do AulaNet 2.0**

Quando o Scriba foi desenvolvido, a Sun Microsystems ainda não havia distribuído a tecnologia JSP que tem funcionalidades semelhantes às do Scriba. A utilização de uma tecnologia proprietária como o Scriba dificulta a integração de novos membros à equipe de desenvolvimento e dificulta o desenvolvimento de novos produtos. Um outro problema na arquitetura atual do AulaNet é a disseminação de código em arquivos HTML e classes Java. Os arquivos HTML com comandos Scriba recorrem a classes implementadas em Java para funções mais complexas. Esta disseminação de código entre as diferentes linguagens e arquivos dificulta a manutenção do ambiente.

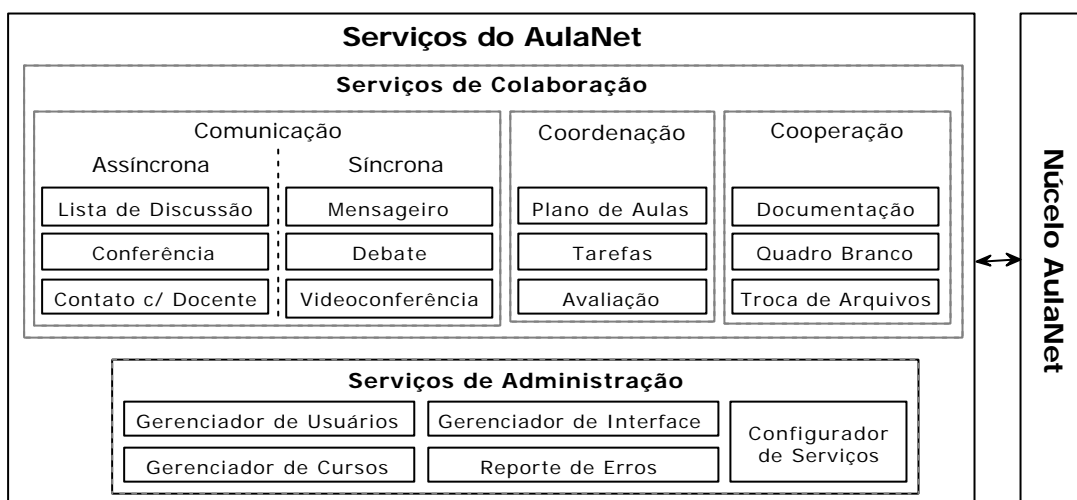
As classes Java implementam as funções específicas de cada página. Desta forma, apesar de terem sido desenvolvidas utilizando uma linguagem orientada a objetos, estas classes se comportam com uma biblioteca de funções, o que se assemelha ao paradigma de programação funcional [Cousineau, 1998]. O uso da tecnologia Scriba, a baixa modularidade (disseminação de código) e o uso do paradigma funcional são considerados os problemas mais graves da arquitetura e implementação do AulaNet 2.0. Além disto, o AulaNet vem sendo desenvolvido por prototipação e seus desenvolvedores, alunos de doutorado, mestrado e graduação da PUC-Rio, além de mantê-lo, usam-no em suas teses, dissertações e monografias, implementando e testando os conceitos de seus trabalhos. Com isto o AulaNet cresceu e suas funcionalidades foram implementadas na medida da necessidade, tornando necessária agora uma reestruturação do código.

### 3. A Arquitetura do AulaNet 3.0

Requisitos de groupware dificilmente são claros o suficiente para possibilitar uma especificação precisa antecipada do comportamento do sistema. É difícil prever como um determinado grupo colabora [Gutwin & Greenberg, 2000]. Os sistemas colaborativos exigem então o uso de metodologias que comportem a natureza evolucionária de um groupware,

eliminando a necessidade de um conjunto completo de requisitos. O uso de componentes de software facilita a adoção de requisitos que se alteram [Szyperski, 1999].

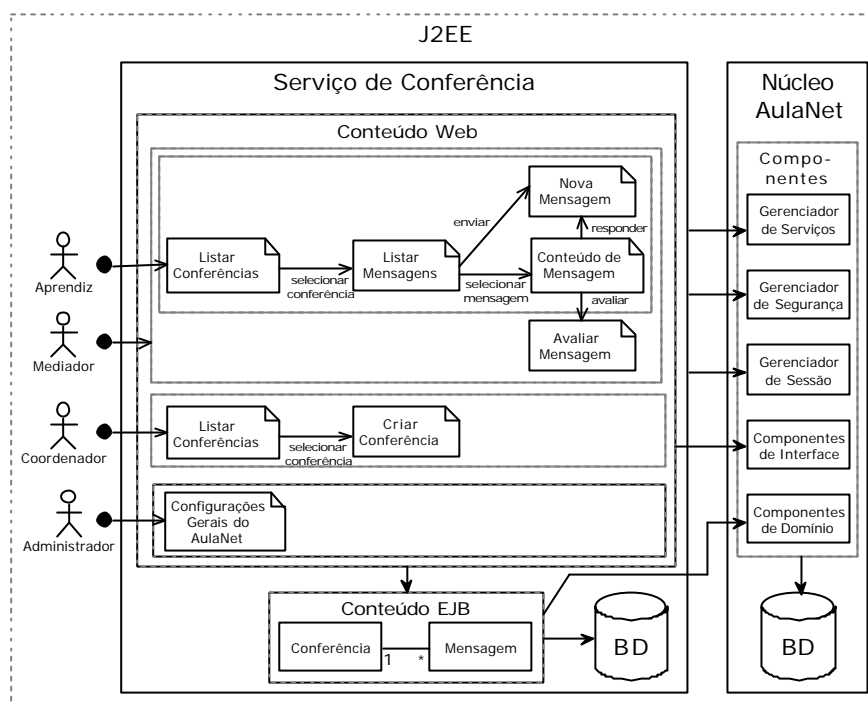
Na nova arquitetura, o AulaNet é constituído de serviços que se conectam a um núcleo principal, conforme esquematizado na Figura 3. Os serviços exemplificados nesta figura já estão presentes no AulaNet 2.0, porém sua implementação é misturada com todo o código. No AulaNet 3.0 eles serão componentes que podem ser alterados, substituídos ou removidos do ambiente. Também poderão ser incluídos os novos serviços que forem sendo desenvolvidos. Estes serviços também serão desenvolvidos internamente através de componentes.



**Figura 3. AulaNet 3.0: Arquitetura baseada em serviços e componentes**

Dada a necessidade de instalações e configurações mínimas nas estações clientes, nesta nova arquitetura foi mantida a estratégia de cliente magro (*thin client*), onde a maior parte do processamento é feita no servidor. Os clientes usarão um navegador Web para interagir com o AulaNet e seus serviços. Eventualmente alguns serviços mais complexos, como os serviços de comunicação síncrona, necessitarão de componentes executando no cliente.

A Figura 4 exemplifica como um serviço (Conferências) é projetado para esta nova arquitetura. O núcleo AulaNet contém componentes para lidar com funcionalidades gerais a todos os serviços. O *Gerenciador de Serviços* gerencia o acoplamento e configuração dos serviços do ambiente; cuida de questões como instalação de serviços, controle de versões, localização remota, etc. O *Gerenciador de Segurança* cuida da identificação, autenticação e controle de acesso (permissões) dos usuários. O *Gerenciador de Sessão* cuida da persistência de dados entre as chamadas de um mesmo usuário. Os *Componentes de Interface* devem ser utilizados pelos serviços para montar a interface com o usuário. Estes componentes implementam os padrões de interface do AulaNet e gerenciam os textos a serem apresentados num determinado idioma e que podem ser modificados pelo administrador do AulaNet na instituição. Os *Componentes do Domínio* contém objetos que representam o modelo de dados comum entre os serviços, contendo objetos como aprendiz, docente, curso, turma, matrícula, etc. Mantendo os componentes de interface e do domínio separados, pode-se utilizar a mesma lógica da aplicação para interfaces e domínios diferentes, o que é especialmente útil para o gerador de groupware.



**Figura 4. Projeto do serviço Conferências na arquitetura do AulaNet 3.0**

Com a nova arquitetura orientada a serviços, componentes e objetos, disponibilizando um núcleo comum aos diversos serviços do AulaNet, pode-se adicionar, remover e substituir componentes por versões mais robustas ou que sejam mais apropriadas ao hardware, ao sistema operacional ou a produtos legados com os quais o ambiente tenha que operar. Além disto, os serviços do AulaNet podem estar distribuídos em várias máquinas visando melhorar a performance ou facilitar o acesso ao recurso. O desenvolvimento baseado em componentes possibilita alta modularidade [Clements, 1995], o que facilita equipes externas ao projeto AulaNet, como pesquisadores de outras instituições, desenvolverem e adquirem novos serviços e incorporá-los ao ambiente.

Nesta arquitetura, o J2EE auxilia na integração dos componentes, provendo recursos para a distribuição dos mesmos em mais de um servidor AulaNet, melhorando com isto a escalabilidade do ambiente. A conectividade remota é feita através de enterprise beans, cujos métodos são chamados como se o componente estivesse na mesma máquina virtual. Para localizar os beans, utiliza-se a tecnologia JNDI (*Java Naming and Directory Interface*). Além disto, o J2EE também provê recursos próprios de tratamento de sessão e segurança. O J2EE cuida da autenticação do usuário, da restrição de acesso a recursos por papel e da transmissão de variáveis de sessão através de cookies e urls codificadas.

#### 4. Conclusão

Como os processos de trabalho entre os indivíduos são muito específicos e evoluem com o tempo, a tecnologia de groupware deve prover flexibilidade suficiente para ser adaptada às necessidades de cada grupo e à evolução dos processos de trabalho. O groupware deve prover a possibilidade do grupo montar seu contexto de trabalho, selecionando e configurando um conjunto de ferramentas colaborativas específicas para suas necessidades. A utilização de técnicas de desenvolvimento baseado em componentes é uma forma de facilitar o desenvolvimento de groupware para que este seja mais flexível e modularizado.

O desenvolvimento baseado em componentes facilita a prototipação e o modelo espiral de desenvolvimento de software, o que vai ao encontro das características da equipe de

desenvolvimento do AulaNet, que é composta de alunos desenvolvendo seus trabalhos de curso. Além disto, possibilita que equipes de outras instituições e empresas, em posse das especificações das interfaces do núcleo do AulaNet, possam desenvolver novos componentes a serem acoplados. Com isto, a equipe AulaNet poderá focalizar seus esforços na arquitetura e na integração dos componentes e instituições que forem usar o AulaNet terão subsídios para adequá-lo e incrementá-lo de acordo com suas necessidades. Apesar de suas vantagens, o desenvolvimento baseado em componentes demanda mais esforço inicial no projeto e implementação para montar a infra-estrutura do sistema em questão e construir uma biblioteca robusta de componentes reutilizáveis.

Como grande parte dos princípios adotados no AulaNet valem para outros ambientes de trabalho em grupo, as características do ambiente de aprendizagem AulaNet podem ser aproveitadas na elaboração de um gerador, que poderá ser utilizado para instanciar groupware específicos para cada situação, como por exemplo, medicina, engenharia, projeto de software e, obviamente, aprendizagem, bastando para tanto alterar a configuração dos componentes e a nomenclatura utilizada.

Neste artigo, foi apresentada a nova arquitetura do ambiente AulaNet baseada em componentes e serviços. Foi também apresentada a arquitetura atual do AulaNet, as dificuldades identificadas na sua manutenção e aperfeiçoamento e alguns motivos que levaram a propor uma nova arquitetura para o ambiente. A transição para esta nova arquitetura está sendo feita de forma gradual, transformando os serviços antigos em novos serviços compatíveis com os novos paradigmas requeridos: orientação a objetos, construção de componentes, conceito de serviços, etc. Desta forma, o novo código pode coexistir com o antigo, podendo ser testado dentro do ambiente completo, e o caminho em direção da distribuição através do J2EE vai sendo trilhado gradativamente.

## 5. Referências

- 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). Rio de Janeiro. pp. 119-133, 1999.
- Clements, P.C. (1995), "From Subroutines to Subsystems: Component Based Software Development," *American Programmer*, vol. 8, No. 11, November 1995.
- Cousineau, G. & Mauny, M. (1998) *The Functional Approach to Programming*, Cambridge University Press, English edition, December 1998. ISBN: 0521576814
- Grudin, J. (1989) Why groupware applications fail: Problems in design and evaluation. *Office: Technology and People* 4, (3), 245-264.
- Gutwin, C., Greenberg, S. (2000) The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. *IEEE 9th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2000, pp. 98-103.
- Lucena, C.J.P. & Fuks, H. (2000) *Professores e Aprendizes na Web: A Educação na Era da Internet*. ISBN 85-88011-01-8 Editora Clube do Futuro, Rio de Janeiro, 2000.
- Szyperski, C. (1999) *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley, 1999.