



Barreto, C.G., Fuks, H. & Lucena, C.J.P. (2005) "Agregando Frameworks em uma Arquitetura Baseada em Componentes: Um Estudo de Caso no Ambiente AulaNet", Anais do 5º Workshop de Desenvolvimento Baseado em Componentes - WDBC 2005, 07-09 de Novembro, Juiz de Fora-MG, ISBN 85-88279-47-9, pp. 25-32.  
Disponível em <http://www.les.inf.puc-rio.br/groupware>



# **Agregando Frameworks em uma Arquitetura Baseada em Componentes: Um Estudo de Caso no Ambiente AulaNet**

**Celso Gomes Barreto, Hugo Fuks & Carlos José Pereira de Lucena**

Lab. de Engenharia de Software (LES), 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

{celsogbj, hugo, lucena}@inf.puc-rio.br

***Abstract.** Software development is propitious to the component based development technique. In this paper it is presented how a groupware can get benefits when aggregating frameworks, taking as case the AulaNet learning environment and the frameworks Spring, Hibernate and Java Server Faces.*

***Resumo.** O desenvolvimento de software é propício à aplicação de técnica de desenvolvimento baseado em componentes. Neste artigo é apresentado como um groupware pode se beneficiar da agregação de frameworks tomando como estudo de caso no ambiente de aprendizado AulaNet e os frameworks Spring, Hibernate e Java Sever Faces.*

## **1. Introdução**

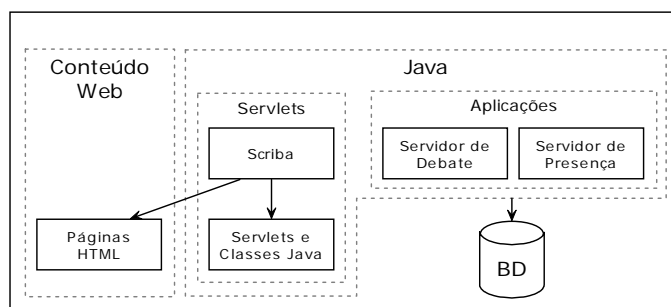
O desenvolvimento de softwares que dão suporte ao trabalho colaborativo, denominados groupware, é uma tarefa complexa que requer domínio em diversas áreas de conhecimento, como informática, pedagogia, psicologia, sociologia, etc. Uma arquitetura componentizada permite que componentes sejam selecionados para montar um groupware voltado aos interesses específicos de um grupo [Gerosa et al., 2004].

Segundo [Fayadd, 1997], frameworks de classes podem ser classificados como de infra-estrutura, integração ou de domínio de aplicação. Há ainda frameworks de componentes que regulamentam como componentes plugados no framework podem interagir [Szyperski, 1999]. Neste artigo será apresentado como uma arquitetura baseada em componentes pode se beneficiar do uso de frameworks de aplicação tomando como estudo de caso o ambiente AulaNet.

## **2. O Ambiente AulaNet**

O AulaNet é um ambiente de ensino-aprendizagem na Web baseado no modelo de colaboração 3C que vem sendo desenvolvido desde junho de 1997 pelo Laboratório de Engenharia de Software (<http://www.les.inf.puc-rio.br>) da PUC-Rio e distribuído gratuitamente pela EduWeb (<http://www.eduweb.com.br>), que também realiza modificações no AulaNet sob demanda de seus clientes. Os serviços do AulaNet são classificados em serviços de comunicação, coordenação e cooperação, seguindo o modelo 3C [Fuks et al., 2005].

A arquitetura do AulaNet 2.0, como pode ser visto na figura 1, é baseada em um modelo cliente-servidor e usa a tecnologia Scriba [Fuks et al., 2004].



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

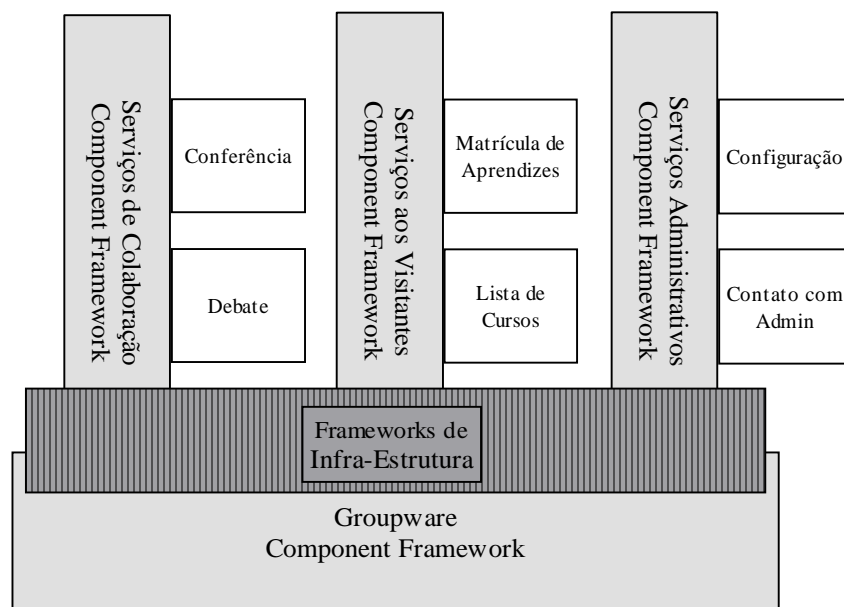
A tecnologia Scriba é composta por uma linguagem de script que é embutida em arquivos HTML e um Servlet que interpreta os arquivos HTML. O Scriba permite, dentre outras coisas, chamadas a classes Java, acesso a banco de dados, definições de variáveis.

Vale ressaltar que quando o AulaNet começou a ser desenvolvido em 1997 a tecnologia Java Server Pages não existia. Esta, que tem funcionalidades semelhantes à do Scriba, surge apenas em meados de 1999. O uso de tecnologias proprietárias dificulta a integração com outras equipes de desenvolvimento, como a EduWeb, que precisa treinar cada novo funcionário nesta tecnologia. Este e outros problemas são endereçados na versão 3.0 do AulaNet da qual é falado a seguir.

### **3. A Arquitetura do AulaNet 3.0**

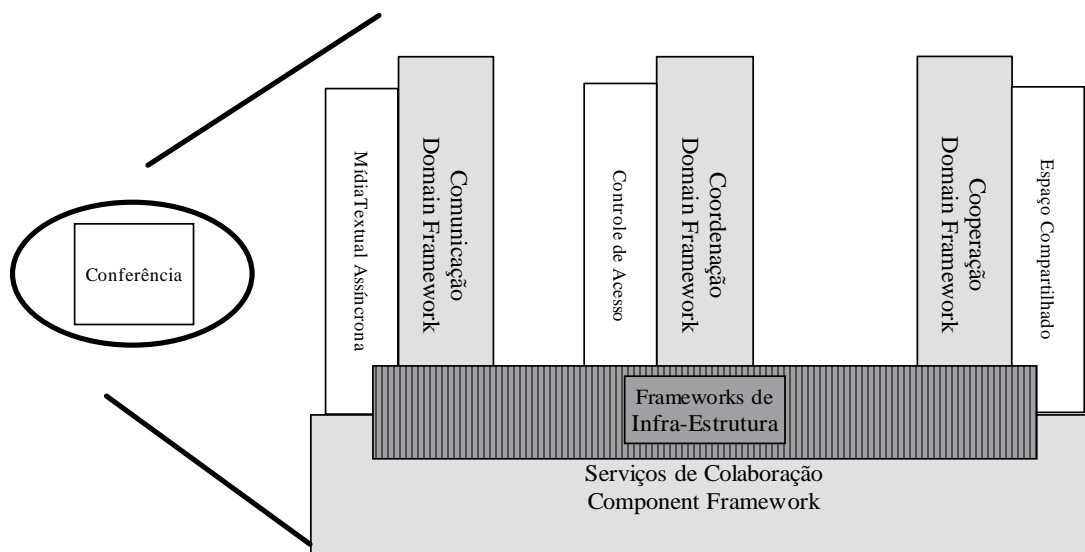
O AulaNet deve ser flexível para ser configurável pelos diversos grupos de alunos e professores que o usam. Cada curso pode ter sua própria abordagem pedagógica, nível de profundidade de conteúdo, etc. É impossível prever todas as necessidades dos usuários e, mesmo que fosse possível prever, o ambiente se tornaria difícil de usar e configurar. Este cenário de requisitos instáveis é propício para a abordagem de desenvolvimento baseado em componentes que provê maior flexibilidade para atender estes requisitos [Szyperski, 1999].

Os serviços do AulaNet podem ser vistos como componentes que são plugados ou desplugados conforme a necessidade do grupo (figura 2). Um framework de componentes denominado Groupware Component Framework fornece um conjunto de interfaces que devem ser implementadas pelos componentes para que estes possam ser plugados no framework e interagir com outros componentes. O Groupware Component Framework possui ainda três extensões: Component Framework de serviços administrativos, usado para gerar componentes de administração do sistema como, por exemplo, a configuração do ambiente e o contato com o administrador; Component Framework de serviços de visitantes usado para gerar componentes acessíveis a visitantes não cadastrados como o a matrícula de aprendizes e a listagem de cursos do ambiente; Component Framework de serviços de colaboração, usado para gerar os componentes de colaboração como Conferência e Debate [Fuks et al., 2005].



**Figura 2: Arquitetura do AulaNet 3.0 (por motivos de clareza, apenas alguns serviços foram representados).**

Cada serviço também é desenvolvido usando uma abordagem baseada em componentes, com pode ser visto na figura 3.



**Figura 3: O serviço conferência e seus componentes (por motivos de clareza, apenas alguns componentes foram representados).**

Os componentes de colaboração são plugados no Component Framework de Serviços de Colaboração e são implementados usando os frameworks de domínio de aplicação: Comunicação Domain Framework, Coordenação Domain Framework e Cooperação Domain Framework específicos para cada um dos três aspectos do modelo 3C. Estes frameworks refletem a experiência adquirida pelo Groupware@LES ao longo do tempo que o AulaNet vem sendo desenvolvido.

A utilização de frameworks de infra-estrutura possibilita que tanto os frameworks de componentes quanto os frameworks de domínio de aplicação usados na arquitetura do AulaNet 3.0 sejam especializados na sua área de domínio, deixando aspectos como gerência de eventos de interação com usuário, por exemplo, para outros frameworks, denominados frameworks de infra-estrutura.

#### 4. Agregando Frameworks de Aplicação a Arquitetura do AulaNet 3.0

O uso de frameworks contribui na composição de soluções eficazes, pois dão suporte a abstrações de mais alto nível para criar componentes e estruturas reusáveis [Caminada, 2003]. Nas próximas subseções são apresentados três frameworks de aplicação usados no AulaNet 3.0: Java Server Faces (<http://java.sun.com/j2ee/javaserverfaces/>), Spring (<http://www.springframework.org>) e Hibernate (<http://www.hibernate.org/>).

##### 4.1. O Framework de Infra-Estrutura Java Server Faces

Java Server Faces (JSF) é uma tecnologia desenvolvida pela Sun Microsystems que compreende um framework para representar componentes de interface com o usuário e gerenciar o seu estado. A tecnologia JSF oferece um conjunto pré-definido de componentes para atender as necessidades mais usuais, como botões e caixas de textos. Ela também implementa o padrão de projeto Model View Controller que irá garantir a separação entre o modelo e visão, considerada uma boa prática de design de software, e entre visão e controlador, considerada importante em se tratando de interfaces web [Fowler, 2002].

Apesar do conjunto padrão de componentes do JSF cobrir diversos casos recorrentes na composição de interfaces web, este pode ser estendido acrescentando novos componentes. Existem projetos como o MyFaces (<http://myfaces.apache.org/>) que fornecem gratuitamente componentes adicionais como por exemplo um componente que permite que o usuário selecione uma data em um calendário (figura 4) ou um componente que apresenta um mini-editor de textos HTML estilo *What You See Is What You Get* (figura 5) que já foi solicitado por diversos usuários do AulaNet.



Figura 4: Calendário.

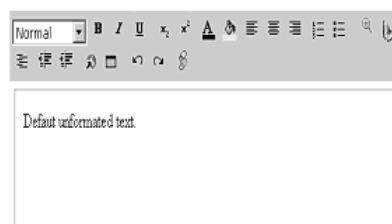


Figura 5: Editor HTML.

Por ser um padrão estabelecido por um Java Community Process, processo colaborativo onde as especificações da tecnologia Java são criadas por membros da própria comunidade, cujo grupo de especialistas é composto por várias empresas que desenvolvem ferramentas de desenvolvimento como, por exemplo, Borland Software Corporation, IBM, Oracle e BEA Systems, é de se esperar que cada vez mais JSF seja incorporado a ferramentas de produtividade. De fato, já existem alguns ambientes integrados de desenvolvimento com suporte a desenvolvimento com Java Sever Faces como, por exemplo, o Borland JBuilder e o Sun Java Studio Creator mostrado na figura

6. Estes ambientes possibilitam que o desenvolvedor escolha o componente mais adequado à sua necessidade em um catálogo.

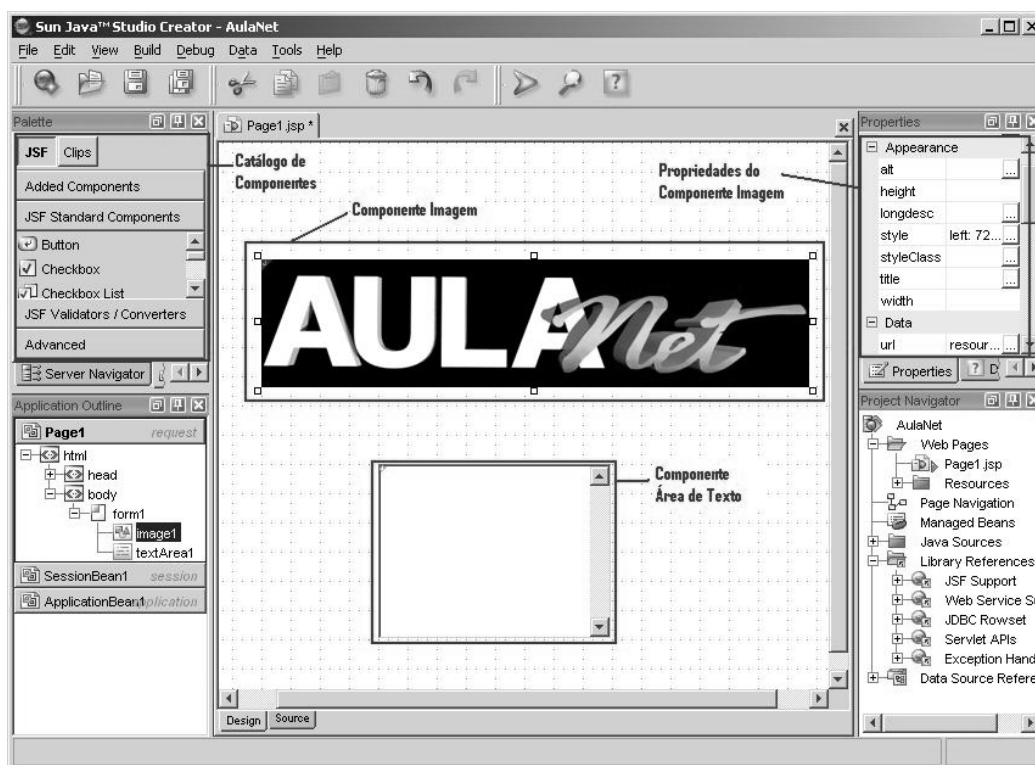


Figura 6: Sun Java Studio Creator.

O Groupware Component Framework e o Component Framework de Serviços de Colaboração estendem o JSF fornecendo novos componentes de interface de groupware como, por exemplo, a lista de participantes de um chat.

#### 4.2. O Framework de Infra-Estrutura Hibernate

Toda aplicação colaborativa tem que se preocupar com a persistência de seus dados que pode ser implementada de várias formas. Um das características da arquitetura do AulaNet 2.0 é que diversas classes realizavam consulta ao banco de dados o que trazia problemas de manutenibilidade, já que uma mudança na base de dados se propaga para um número imprevisível de classes. Para solucionar este problema é definido um conjunto de componentes que implementam o padrão de projeto Data Access Object. Estes componentes possuem uma estratégia que define como o acesso ao banco de dados é realizado [Johnson, 2004]. Optou-se por usar o framework Hibernate como estratégia.

Hibernate é o framework responsável por mapear objetos em tabelas de um banco de dados relacional, estreitando o *gap* conceitual que existe entre os dois. Ele também fornece uma linguagem própria de seleção, o Hibernate Query Language (HQL) similar ao SQL, mas que possibilita a seleção de objetos ao invés de linhas. O mapeamento objeto-tabela é feito através de arquivos XML. A configuração das características do banco de dados no Hibernate pode ser feita de diversas formas como, por exemplo, através da própria interface do framework ou através de arquivos de

configuração XML. As operações de criação, atualização, remoção e seleção de objetos são feitas através da API do Hibernate que, consultando seus arquivos de mapeamento e configuração, envia os comandos corretos ao banco de dados configurado. Desta forma, a aplicação fica portátil entre bancos de dados.

Apesar do Hibernate estreitar o *gap* conceitual entre objetos e tabelas, uma implementação que use ele está sujeita a erros de programação como, por exemplo, o esquecimento de sessões abertas. Para evitar estes erros, optou-se por usar o framework Spring.

### 4.3. O Framework de Infra-Estrutura Spring

Spring é um framework dividido em módulos que podem ser usados separadamente ou em conjunto conforme a necessidade do projeto. O módulo de mapeamento objeto-relacional integra-se ao Hibernate, controlando a abertura e o fechamento de sessões, evitando o aparecimento de erros como o descrito na subseção anterior. Outros módulos usados no AulaNet 3.0 são o de *Dependency Injection* e o de programação orientada a aspectos.

O termo *Dependency Injection* foi cunhado em [Fowler, 2004] como uma forma mais específica de inversão de controle. Considere o seguinte exemplo demonstrado na figura 7 com a classe Conference usando um componente de envio de e-mails.

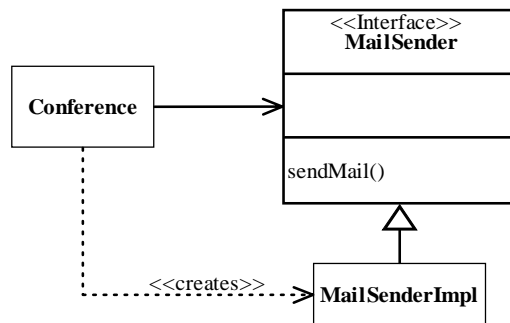


Figura 7: Relações de Dependência Sem *Dependency Injection*.

A classe Conference referencia a interface do componente MailSender, mas também está acoplada a implementação pois precisa obter uma referência para usá-la. Com *Dependency Injection* um outro objeto, o Assembler, é responsável por instanciar a implementação adequada do componente e configurar a classe Conference (figura 8).

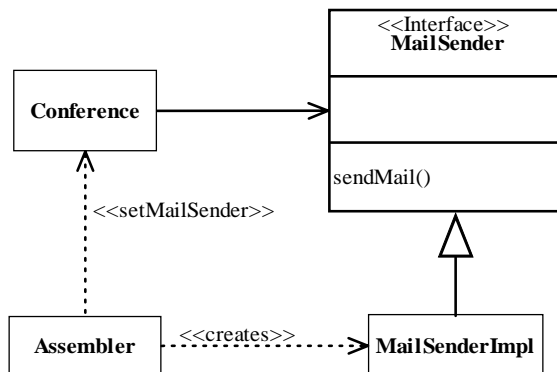


Figura 8: Relações de Dependência com *Dependency Injection*.

Desta forma consegue-se um baixo acoplamento entre o componente MailSender e a classe que o usa.

O módulo de *Dependency Injection* do framework Spring entra no lugar do Assembler, gerenciando a criação dos componentes e a configurações de dependências que são descritos em um arquivo em formato XML. Dessa forma consegue-se um acoplamento mais frouxo entre os componentes usados no AulaNet 3.0.

Finalmente, o módulo de programação orientada a aspectos do Spring compreende um framework de geração de aspectos bem integrado com o módulo de *Dependency Injection* e um conjunto de aspectos comuns em diversas aplicações. A programação orientada a aspectos é um paradigma que propõe uma forma de tratar interesses transversais, os aspectos, que costumam se espalhar pelo código em outros paradigmas como o da orientação a objetos. É importante ressaltar que a programação orientada a aspectos atua de forma complementar a orientação a objetos.

Dois exemplos de aspectos usados no AulaNet 3.0 são *logging* de performance a cada operação, de forma que os esforços de otimização possam ser direcionados para as operações críticas do sistema e controle de transações, que permitem que POJOs (Plain Old Java Objects) utilizem um modelo de transações declarativas similar ao modelo dos Enterprise Java Beans.

## 5. Conclusão

A construção de component frameworks que especificam as interfaces com as quais os componentes plugados no framework podem interagir fornece uma base sobre a qual componentes de groupware podem ser desenvolvidos. Paralelamente, frameworks de domínio de aplicação que implementam aspectos de groupware podem ser usados na implementação dos componentes.

A incorporação de outros frameworks de infra-estrutura como Hibernate, JSF e Spring retirou a responsabilidade dos frameworks de domínio de aplicação e de componentes de tratar aspectos fora do seu domínio, como persistência de dados, por exemplo. Dentre os benefícios desta abordagem, o uso de frameworks e padrões abertos possibilitam a integração com outras equipes de desenvolvedores sem que seja preciso um treinamento específico para uma tecnologia proprietária. Além disso, será reusado código que já foi testado em outras aplicações.

Neste artigo, foi discutido como ambientes de groupware podem se beneficiar do desenvolvimento baseado em componentes. Para o AulaNet, propôs-se uma arquitetura baseada em componentes utilizando tanto frameworks específicos para groupware quanto frameworks que tratam de outros aspectos que são comuns a outros domínios. O projeto AulaNet sabe por experiência própria o impacto do uso de soluções caseiras pois, como dito antes, a tecnologia Scriba apesar de necessária em 1997 quando o AulaNet começou a ser desenvolvido tornou-se desnecessária em 1999 com o advento do JSP. Para o desenvolvimento do AulaNet 3.0 optou-se por usar soluções bem conhecidas como frameworks, padrões de projetos sempre que possível.

## 6. Agradecimentos

O Projeto AulaNet é financiado 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 Lucena nº 300031/92-0 e Hugo Fuks nº 303055/02-2 e da CAPES: Celso Gomes Barreto.

## 7. Referências

- Caminada, N., Filho, A.H., Godoy, R. A.& Staa, A. V. (2003): “Um modelo para desenvolvimento e evolução de componentes utilizando frameworks e design patterns” WDBC 2003 - III Workshop de Desenvolvimento Baseado em Componentes.
- Fayad, M., Schmidt, D. C. (1997): Object-oriented application frameworks, Communications of the ACM, v.40 n.10, p.32-38, Oct. 1997
- Fowler, M. (2002): Patterns of Enterprise Application Architecture. Reading: Addison-Wesley, 2002
- Fowler, M. (2004): Inversion of Control Containers and the Dependency Injection pattern: Artigo disponível em <http://martinfowler.com/articles/injection.html>.
- Fuks, H., Gerosa, M.A., Pimentel, M.G., Raposo, A.B., Mitchell, L.H.R.G. & Lucena, C.J.P. (2003): “Evoluindo para uma Arquitetura de Groupware Baseada em Componentes: o Estudo de Caso do Learningware AulaNet”, WDBC 2003 - III Workshop de Desenvolvimento Baseado em Componentes, Anais Eletrônicos, São Carlos-SP, 10 a 12 de setembro de 2003.
- Fuks, H., Raposo, A.B., Gerosa, M.A. & Lucena, C.J.P. (2005): “Applying the 3C Model to Groupware Development”, International Journal of Cooperative Information Systems (IJCIS), v.14, n.2-3, Jun-Sep 2005, World Scientific, ISSN 0218-8430, p.299-328.
- Gerosa, M.A., Barreto, C.G, Raposo, A.B., Fuks, H. & Lucena, C.J.P. (2004): “O Uso de uma Arquitetura Baseada em Componentes para Incrementar um Serviço do Ambiente AulaNet”, Anais do 4º Workshop de Desenvolvimento Baseado em Componentes - WDBC 2004, 15-17 de Setembro, João Pessoa-PB, ISBN 85-7669-001-2, pp. 55-60.
- Johnson, R. (2004): Expert One-on-One J2EE Development without EJB. Reading: Wiley Publishing Inc., 2002
- Szyperski, C. (1999): Component Software: Beyond Object-Oriented Programming. Addison-Wesley.