

OdysseyShare: Desenvolvimento Colaborativo de Componentes

**Cláudia Werner¹, Marcos R. S. Borges², Marta Mattoso¹, Regina Braga³,
Fernanda Campos³, Marco Mangan^{1,4}, Vaninha Vieira¹**

¹Programa de Engenharia de Sistemas - COPPE/UFRJ

²Departamento de Ciência da Computação – IM&NCE/UFRJ

³Universidade Federal de Juiz de Fora

⁴Faculdade de Informática – PUCRS

odyssey@cos.ufrj.br - <http://www.cos.ufrj.br/~odyssey/share>

Abstract: *Component-based software development is a collaborative intensive activity. Knowledge necessary to identify, specify, develop, and reuse a component is usually dispersed among stakeholders. This means that stakeholders need to collaborate to share their knowledge assets. This paper presents the OdysseyShare Project, whose aim is to explore the collaborative aspects of software component development.*

Resumo: *O desenvolvimento de componentes de software é uma atividade colaborativa. O conhecimento necessário para a identificação, especificação, construção e reutilização do componente encontra-se distribuído entre os integrantes da equipe de desenvolvimento. Por esse motivo, os integrantes precisam colaborar para compartilhar seu conhecimento. Este artigo apresenta o projeto OdysseyShare, cujo objetivo é explorar aspectos de colaboração do desenvolvimento de componentes.*

1. Introdução

O Desenvolvimento Baseado em Componentes (DBC) [Szyperki, 1998] está cada vez mais ganhando adeptos e é uma das propostas promissoras para a melhoria da produção de software nos próximos anos. A construção de software através de componentes leva a uma maior reutilização e conseqüente alta na qualidade e produtividade.

Grande parte dos produtos de software disponíveis no mercado ainda utiliza uma abordagem de desenvolvimento em módulos cujo relacionamento não é evidente e nem bem delimitado. Por outro lado, o DBC tem como objetivo a definição de componentes interoperáveis, com interfaces bem definidas, evidenciando os tipos de relacionamentos permitidos por estes componentes. Desta forma, a complexidade no desenvolvimento é reduzida, assim como os custos, através da reutilização de componentes exaustivamente testados.

Segundo Krueger (1992), uma técnica de reutilização de software deve reduzir a distância cognitiva entre o conceito inicial de um sistema e sua implementação final. Sendo assim, no contexto de um processo de DBC, a aplicação da reutilização nas fases iniciais do desenvolvimento deve ter como resultado a reutilização de componentes na implementação da aplicação, ou seja, deve haver uma relação direta entre os conceitos de um dado domínio de aplicação considerados reutilizáveis nas fases iniciais do processo de desenvolvimento e os componentes codificados que serão utilizados na implementação da aplicação. Logo,

consideramos que o DBC, para ser efetivo, deve estar apoiado num processo de Engenharia de Domínio (ED) que o permita criar componentes de alta qualidade, atendendo os requisitos de um dado domínio de aplicação.

A ED é o processo de busca, análise, manipulação e formalização das informações do domínio relevantes (i.e. modelo de domínio) para a implementação de um programa de reutilização, que promova o desenvolvimento de aplicações do domínio a um custo reduzido. A ED tem como característica a utilização de uma abordagem baseada em modelos de domínio para facilitar a instituição da reutilização e sua aplicação em cadeia no ciclo de desenvolvimento de software. Além disso, a ED deve permitir a criação e posterior evolução desses modelos em um processo intuitivo, interativo e iterativo. Neste sentido, as informações do domínio coletadas durante o processo devem ser organizadas, manipuladas e validadas pelos engenheiros do domínio durante todo o processo, para assegurar resultados mais confiáveis. Devem ainda ser armazenadas adequadamente, sendo seu acesso garantido para a construção de aplicações no domínio.

As atividades deste processo exigem interação e cooperação entre os diversos profissionais envolvidos. Desta forma, a ED é desempenhada conjuntamente por profissionais com funções distintas e que possuem diferentes *backgrounds*, visões, formas de trabalho e interesses, o que torna a interação social bastante rica neste processo.

No entanto, várias dificuldades existem em relação à interação humana, como por exemplo, a pouca disponibilidade de tempo e sua possível distribuição geográfica [Machado et al., 1999]. Esses fatores diminuem a produtividade do processo. Portanto, é importante estimular novas formas de trabalho, que não só amenizem os problemas de interação, mas aumentem a qualidade do processo através de um enfoque em aspectos de colaboração. O reconhecimento de que processos de desenvolvimento de software são extremamente colaborativos não é uma novidade. Sua natureza colaborativa tem motivado a proposta e construção de uma série de ferramentas e ambientes de apoio à colaboração [Begole, Rosson e Shaffer, 2001] [Farshchian, 2001] [Grundy, Mugridge e Hoskings, 2000] [Gutwin e Greenberg, 1999].

Neste projeto, estamos preocupados com a proposta e construção de um ambiente cooperativo para o desenvolvimento de componentes, denominado *OdysseyShare* [Werner et al., 2002], que combine as técnicas utilizadas no contexto do DBC, Trabalho Cooperativo e Banco de Dados. Este projeto tem como objetivo explorar os aspectos colaborativos, através da construção de mecanismos para o suporte à interação em grupo em um ambiente para DBC pré-existente denominado *Odyssey* [Werner et al., 2000].

O projeto teve sua primeira fase iniciada em 2001, sendo um projeto de pesquisa integrado, com financiamento CNPq, e desenvolvido por uma equipe interdisciplinar, formada através da colaboração entre as áreas de engenharia de software e banco de dados (COPPE/UFRJ), de sistemas de informação, mais especificamente, trabalho cooperativo (IM-NCE/UFRJ) e qualidade de software (Núcleo de Qualidade Softex/UFJF).

A primeira etapa envolveu as seguintes vertentes de pesquisa tecnológica: (i) técnicas de trabalho cooperativo/*groupware*, (ii) máquina de processo baseada em agentes, (iii) definição de uma arquitetura de componentes baseada em modelos de análise, (iv) técnicas de gerência e acesso a componentes reutilizáveis heterogêneos e distribuídos, (v) agentes inteligentes para apoio à navegação dinâmica na Internet e (vi) integração de banco de dados com serviços de ontologia. Os resultados desta etapa foram apresentados em publicações sobre o ambiente [Werner et al., 2002] [Murta, 2002] [Souza et al., 2001] [Mangan et al., 2002] e um estudo de caso realizado no domínio agropecuário [Campos e Braga, 2003].

A segunda etapa teve início neste ano, com a integração da equipe de sistemas de informação da PUCRS. As vertentes de pesquisa nesta etapa são: (i) continuidade dos trabalhos relacionados a técnicas de trabalho cooperativo/*groupware*, (ii) gerência de configuração, (iii) suporte à percepção em bases de dados compartilhadas, (iv) agentes cooperativos para ambientes de DBC e (v) modelagem baseada em componentes. O estudo de caso no domínio agropecuário prossegue e um novo estudo de caso no domínio de educação baseada na *Web* será realizado.

Neste artigo, apresentamos, inicialmente, uma visão geral sobre o ambiente (Seção 2). Em seguida, o suporte à colaboração proposto é descrito (Seção 3). Este texto inclui ainda um relato do andamento do estudo de caso no domínio agropecuário (Seção 4). Para concluir este texto, apresentamos algumas considerações finais (Seção 5).

2. Uma visão geral sobre o ambiente

O uso de técnicas de reutilização nas fases iniciais do desenvolvimento (ex.: análise, projeto) de aplicações baseado em componentes deve facilitar a reutilização de componentes em fases mais avançadas do desenvolvimento (ex. implementação). Logo, um método de DBC deve se basear em informações específicas a um certo domínio (i.e. conceitos, processos e atividades que são particulares a uma organização ou área de negócios) para a especificação de componentes [Szyperki, 1998]. No *Odyssey*, o processo de engenharia de domínio trata dessa aplicação das técnicas de DBC *para* a reutilização (*Odyssey-Engenharia de Domínio, Odyssey-ED*).

Durante o processo de desenvolvimento de aplicações no domínio, estes componentes são refinados e adaptados ao contexto de um projeto específico. Estas atividades são suportadas por um conjunto de ferramentas para definição, armazenamento, recuperação, navegação e reutilização dos modelos. No *Odyssey*, o processo de engenharia de aplicação trata da aplicação de técnicas de DBC no desenvolvimento *com* reutilização (*Odyssey-Engenharia de Aplicação, Odyssey-EA*).

Os principais usuários do ambiente são o *engenheiro do domínio* e o *especialista do domínio*, que especificam e evoluem os conceitos do domínio, e o *engenheiro de software*, que obtém esse conhecimento sobre o domínio e o aplica na especificação de sua aplicação. A especificação de um domínio pode depender de informação que está além do conhecimento de um engenheiro de software. Por outro lado, especialistas do domínio normalmente não possuem conhecimento sobre técnicas de desenvolvimento, mas possuem uma visão detalhada sobre domínios como saúde, telecomunicações, finanças ou leis.

Sendo assim, um ambiente de desenvolvimento colaborativo de software, com o objetivo de apoiar DBC baseado em modelos de domínio, deve tratar de aspectos relacionados à coordenação, cooperação, comunicação e, especialmente, ao estabelecimento de uma comunidade de especialistas, engenheiros de domínio e engenheiros de software.

A proposta do *OdysseyShare* é similar a de outros ambientes como Serendipity [Grundy, Mugridge e Hosking, 2000], Gossip [Farshchian, 2001] e MILOS [Maurer e Martel, 2002]. Esses ambientes são motivados pelas necessidades de equipes de desenvolvimento distribuídas, ou virtuais, característicos da globalização do desenvolvimento de software. Entre os requisitos desses ambientes, encontram-se: (a) apoio à encenação de processos em equipes distribuídas, (b) incentivo à comunicação e à socialização e (c) apoio à atividade individual e em pequenas equipes. A concepção do *OdysseyShare* apresenta ênfase no suporte a cenários de colaboração frequentes no DBC. Desta forma, provemos processos, atividades, papéis e artefatos adequados aos objetivos da comunidade de desenvolvedores.

Neste contexto, inicialmente, os processos de desenvolvimento da modelagem de domínio e da aplicação propriamente dita são planejados, utilizando-se uma máquina de processos [Murta, 2002]. Esta máquina permite a tomada de decisões gerenciais e o controle do fluxo de trabalho das equipes de engenheiros de software envolvidos nas atividades cooperativas de Engenharia de Domínio e de Aplicação suportadas pelo ambiente. Ao estabelecer estes processos definem-se, entre outros aspectos, a seqüência de atividades, as ferramentas a serem utilizadas, os papéis dos desenvolvedores e os artefatos consumidos e produzidos. Estes artefatos são recuperados em repositórios de componentes na Internet, através de mediadores e ontologias capazes de integrar repositórios distribuídos [Maurer e Martel, 2002]. Finalmente, de posse dos artefatos comuns de trabalho, os membros das diferentes equipes podem interagir através da edição colaborativa ou manter-se informado sobre o trabalho de outros participantes através de um serviço de percepção assíncrono. Neste sentido, um conjunto de componentes de percepção oferece apoio ao trabalho da equipe, em interações remotas, síncronas e assíncronas, em todas as ferramentas do ambiente, conforme detalhado na próxima seção.

3. Suporte à Colaboração no Ambiente

Assumindo que boa parte do tempo do desenvolvimento de software é gasta na edição de documentos e que, com freqüência, a complexidade de um software extrapola as capacidades de um único indivíduo, sendo necessária a participação de diversos desenvolvedores na equipe, o apoio à edição colaborativa de documentos torna-se um requisito fundamental para o suporte à colaboração no desenvolvimento de software.

A edição de um documento por diversas pessoas é uma atividade comum, mas a edição colaborativa com o uso de editores colaborativos não é realizada com freqüência. Em grande parte, a razão para isso é que as facilidades naturais encontradas na edição de uma folha de papel sobre uma mesa, onde dois ou mais participantes expõem suas idéias e rascunham um documento, não se encontram presentes durante a edição através de uma ferramenta colaborativa. Portanto, o problema da percepção é introduzido pela necessidade de um melhor apoio à edição colaborativa.

Em uma sessão de edição colaborativa em tempo real, define-se como informação de percepção (*awareness information*) a informação que aumenta a percepção de cada participante em relação às ações dos outros usuários. Diversos mecanismos de colaboração têm sido propostos na literatura com o objetivo de aumentar a quantidade e a qualidade da informação de percepção disponível aos participantes [Gutwin e Greenberg, 1999]. Tais mecanismos exploram algum tipo de representação gráfica que permite transferir informações a respeito da atividade dos demais participantes para cada participante. A informação de percepção completa a comunicação explícita oferecida por um canal de voz ou bate-papo disponível durante a edição do documento. Dois exemplos desses mecanismos são a visão em radar e o teleapontador (Figura 1).

A visão de radar (*radar view*) é um serviço de percepção comum em editores gráficos e textuais (ex.: *Acrobat Reader*, *Aladdin GhostView*, *Microsoft PowerPoint*) que auxilia a determinar a posição corrente de edição em um documento grande. No contexto de *groupware*, a visão de radar exhibe informações sobre o posicionamento tanto do usuário local como dos demais usuários do grupo. O teleapontador (*telepointer*) apresenta a posição do cursor do mouse, ou de outro dispositivo de indicação, de cada usuário no grupo. Teleapontadores são populares em teleconferências, mas são vistos raramente fora do contexto de *groupware*.

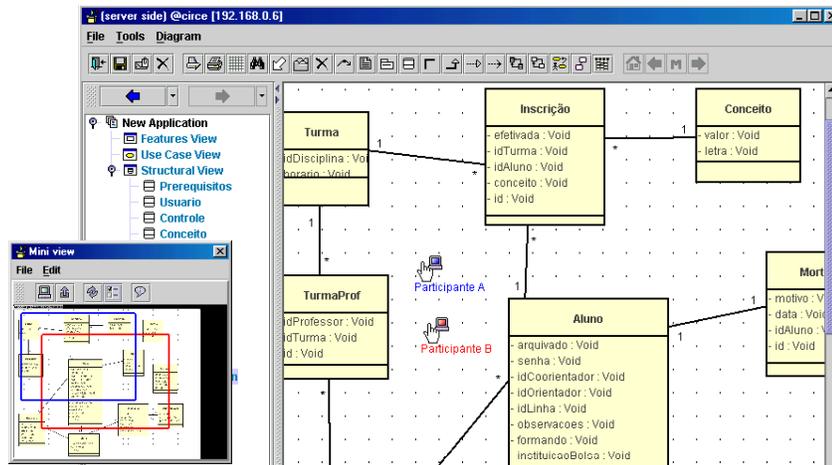


Figura 1 - Teleapontadores (centro) e visão de radar (esquerda) num diagrama UML.

Apesar dos efeitos positivos da utilização desses mecanismos sobre o desempenho dos participantes, sua aplicação fica restrita a aplicações desenvolvidas por grupos de pesquisa em CSCW (Begole, Rosson e Shaffer, 2001). A questão da adoção destes e outros mecanismos disponíveis, porém pouco utilizados na prática, é atualmente um dos maiores desafios de CSCW. A proposta de componentização de mecanismos de percepção parte da premissa de que é possível reutilizar um mesmo mecanismo em diversas aplicações, adotando técnicas de DBC [Szyperki, 1998] apropriadas. Cada componente realiza uma separação de interesses (*concerns*), uma segmentação vertical que realiza completamente o mecanismo, de forma que a aplicação integrada sofra um mínimo de alterações.

Durante a edição, duas ou mais instâncias de execução de uma aplicação formam uma única sessão colaborativa através do compartilhamento de eventos de interface. Os eventos são obtidos através do sistema de janelas (*windowing system*). Essa técnica de implementação é conhecida como colaboração transparente (Begole, Rosson e Shaffer, 2001). Com isso, os componentes podem ser utilizados em qualquer aplicação que utilize o sistema de janelas.

O ambiente *OdysseyShare* conta, ainda, com um serviço de percepção para interações assíncronas (*Ariane*). Esse serviço parte do pressuposto de que nesse modo de interação a cooperação ocorre indiretamente através da manipulação de artefatos compartilhados. Para garantir a durabilidade desses artefatos utiliza-se, em geral, persistência em bancos de dados e através do conhecimento sobre o que mudou no banco de dados é possível saber o que ocorreu aos artefatos compartilhados em interações passadas. *Ariane* objetiva, assim, ampliar a oferta de percepção no ambiente *OdysseyShare* através do monitoramento das mudanças em seus artefatos persistentes.

O processo de percepção utilizado é dividido em três etapas: produção, consumo e análise de eventos. A produção ocorre com a captura das ações realizadas sobre o banco de dados (através do acoplamento de um sensor ao mecanismo de persistência da aplicação) a geração e o armazenamento dos eventos correspondentes. Os tipos de eventos tratados pertencem a quatro classes: conexão, transação, modificação e consulta, abrangendo as ações que uma aplicação pode executar sobre um banco de dados. O consumo dos eventos é realizado por componentes visuais (*widgets*) que apresentam informações de percepção ao usuário.

A análise desses eventos visa prover informações de percepção aos usuários com funções gerenciais, interessados em consultas mais sofisticadas, agregadas e resumidas sobre as interações do grupo. Uma forma de apoiar esse tipo de consulta é através de ferramentas

OLAP (*On-Line Analytical Processing*), as quais possuem capacidades analíticas bastante poderosas e permitem que comparações e cálculos sejam executados sobre várias dimensões dos dados [Kimball e Mertz, 2000]. Para que os eventos produzidos sejam utilizados nessas ferramentas, eles devem passar por um processo de transformação, sendo modelados de forma dimensional e armazenados em um repositório separado, comumente chamado de *cube multidimensional*. A modelagem dimensional e o processo de transformação utilizados foram adaptados de uma proposta para criação de *data webhouses* [Kimball e Mertz, 2000].

A proposta dos serviços de edição e percepção cumpre então dois objetivos: (a) oferecer informações de percepção para permitir a coordenação de atividades no ambiente *OdysseyShare* e (b) apresentar uma alternativa para a adoção de mecanismos de CSCW através da criação de componentes de percepção que podem ser integrados no projeto de uma nova aplicação ou na implementação de uma aplicação já existente.

A implementação atual do suporte à percepção foi realizada na plataforma Java [Kramer, 1996], usando os serviços do sistema de janelas (*Java Foundation Classes*), as facilidades para programação distribuída da invocação remota de métodos (*Remote Method Invocation - RMI*) e o modelo de propagação de eventos (*JavaBeans* e *Abstract Window Toolkit*). Técnicas de programação orientada a aspectos (AOP) [Kiczales, 1997] e de persistência transparente [JDO, 2003] foram empregadas para aumentar o potencial de reutilização da implementação.

4. Estudo de Casos no Domínio Agropecuário

A demanda por tecnologia no setor agropecuário brasileiro é motivada pela busca de redução dos custos, ganho de produtividade e redução de riscos das atividades. O setor agropecuário é extenso e há várias linhas de atuação onde aplicações complexas são desenvolvidas. Algumas das áreas onde a produção de software se destaca são a automação de processos e o monitoramento de rebanhos e da produção na pecuária leiteira.

O desenvolvimento de componentes de software para o setor agropecuário é um dos estudos de casos do projeto de pesquisa *OdysseyShare*. A seleção de sub-domínios (ex. controle leiteiro) e componentes é realizada com base no conhecimento de especialistas do domínio pecuário de órgãos públicos (Embrapa Gado de Leite) e de empresas do setor. Os especialistas auxiliam na validação dos componentes desenvolvidos no trabalho, na detecção das carências do setor e no suporte para o desenvolvimento dos componentes. Parte do conhecimento adquirido no estudo de caso está descrito em uma ontologia (organização de termos do vocabulário) do setor de software agropecuário [Campos e Braga, 2003].

Neste estudo de caso, as etapas iniciais do *Odyssey-ED* foram aplicadas para a especificação do domínio. Assim, analisamos a documentação e sistemas existentes referentes ao domínio agropecuário e especificamos um modelo de domínio para este setor, denominado modelo de *features*. Um modelo de *features* descreve as similaridades e diferenças entre aplicações por meio de uma classificação hierárquica das suas características (*features*) (Figura 2). A partir deste modelo, foram desenvolvidos componentes para o sub-domínio do controle leiteiro.

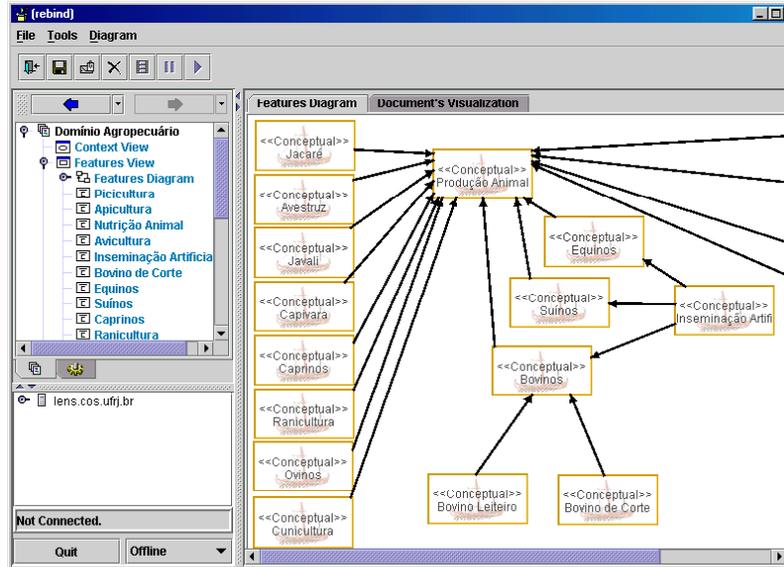


Figura 2 – Modelo de *Features* do domínio agropecuário no ambiente *Odyssey*

O próximo passo do estudo de caso será o desenvolvimento de aplicações (*Odyssey-EA*) para avaliar a aplicabilidade dos componentes desenvolvidos. Nesta etapa, pretende-se utilizar as facilidades de colaboração providas pelo ambiente *OdysseyShare* para promover a colaboração dos especialistas e dos desenvolvedores de aplicação, os quais encontram-se dispersos por diversas empresas produtoras de *software*.

5. Conclusões

O ambiente *OdysseyShare* combina diversas ferramentas para o suporte ao desenvolvimento cooperativo de componentes. Neste artigo, apresentamos em linhas gerais as principais vertentes de pesquisa envolvidas no projeto, com especial ênfase no apoio a edição colaborativa e percepção síncrona e assíncrona. Dois mecanismos colaborativos (telepontador e visão de radar) que minimizam o problema de coordenação das ações individuais de participantes durante a edição de documentos. Participantes que não compartilham o mesmo documento contam com um serviço de percepção assíncrono que oferece informações de percepção sobre as mudanças ocorridas nos repositórios compartilhados. Esses serviços são desenvolvidos na forma de componentes de software, de maneira que outras aplicações similares ao *Odyssey* possam também adotá-los. Apresentamos ainda o atual estágio do estudo de caso no domínio agropecuário que permite a avaliação e aprimoramento contínuos da infra-estrutura proposta por meio da aplicação desses serviços em um processo de desenvolvimento de componentes concreto.

Agradecimentos. Ao CNPq e à CAPES, pelo apoio financeiro, e a todos aqueles que participam deste projeto de pesquisa, por seu trabalho e dedicação.

Referências bibliográficas

- Begole, J.; Rosson, M.B.; Shaffer, C.A. “Flexible Collaboration Transparency: Supporting Worker Independence in Replicated Application-Sharing Systems”, *ACM Trans. on Computer-Human Interaction*, v. 6, n. 2, 1999, p. 95-132.
- Campos, F.; Braga, R., “Desenvolvimento baseado em Componentes: um repositório para o setor agropecuário”, *Simpósio Brasileiro de Informática Aplicada a Agropecuária*, Porto Seguro, Setembro, 2003 (aceito para publicação).

- Farshchian, B. A. "Integrating Geographically Distributed Development Teams Through Increased Product Awareness", *Information Systems Journal*, v. 26, n. 3, Maio 2001, p. 123-141.
- Grundy, J.C.; Mugridge, W.B.; Hosking, J.G. "Constructing Component-based Software Engineering Environments: issues and experiences", *Information & Software Technology*, v.42, n.2, 2000, p. 103-114.
- Gutwin, C; Greenberg, S. "The effects of workspace awareness support on the usability of real-time distributed groupware", *ACM Transactions on Computer-Human Interaction*, v. 6, n.3, 1999, p. 243-281.
- JDO Central, "JDO". In: <http://www.jdocentral.com>, acesso em 06/2003.
- Kiczales, G. et al., "Aspect Oriented Programming". In: *ECOOP'97*, v.1241 LNCS/Springer-Verlag, 1997. p. 220-42.
- Kimball, R., Merz, R., 2000, *The Data WebHouse Toolkit*, New York, USA, John Wiley & Sons, Inc.
- Kramer, D. The Java Platform White Paper. 1996. <ftp://ftp.javasoft.com/docs/papers/JavaPlatform.pdf>.
- Krueger, C. Software Reuse, *ACM Computing Surveys*, vol. 24, no 2, junho de 1992, p. 131-183.
- Machado, M.; Santos, F.; Werner, C.; Borges, M.; "Uma Infra-estrutura de apoio à Aquisição Cooperativa de Conhecimento em Engenharia de Domínio", In: SBES, Florianópolis, outubro 1999, p. 79-94.
- Mangan, M.; Araujo, R.; Kalinowski, M.; Borges, M.; Werner, C. "Towards the evaluation of awareness information support applied to peer reviews of software engineer", In: *Int. Conference on CSCW in Design*, Rio de Janeiro, 2002.
- Maurer, F.; Martel, S. "Process Support for Distributed Extreme Programming Teams", In: *International Conference on Software Engineering, International Workshop on Global Software Development*, Orlando, Florida, USA, 2002.
- Murta, L.G.P.; "Charon: Uma Máquina de Processos Extensível Baseada em Agentes Inteligentes", *Dissertação de M.Sc., COPPE, UFRJ, Rio de Janeiro, 2002.*
- Souza, R. et al. "Software Components Reuse Through Web Search and Retrieval", In: *International Workshop on Information Integration on the Web Technologies and Applications*, Rio de Janeiro, Abr. 2001, p. 12-18.
- Szyperski, C. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley, New York, 1998.
- Werner, C. et al. "Infra-estrutura Odyssey: estágio atual", In: SBES, Ferramentas, João Pessoa, Out. 2000, p. 366-369.
- Werner, C.; Mangan, M.; Murta, L.; et al. "OdysseyShare: Um Ambiente para o Desenvolvimento Cooperativo de Componentes", In: SBES, Gramado, outubro 2002, p. 444-449.