

## **Um Ambiente de Desenvolvimento Distribuído de *Software-DiSEN***

**Elisa H. M. Huzita, Tania F. C. Tait, Thelma E. Colanzi, Marcos A. Quinaia\***

Depto de Informática – Universidade Estadual de Maringá  
Av. Colombo, 5790 - 87020-900 Maringá - PR - Brasil

\*Depto de Ciência da Computação – Universidade Estadual do Centro-Oeste  
Rua Presidente Zacarias, 875 - 85015-430 Guarapuava – PR – Brasil

[emhuzita,tait,thelma@din.uem.br](mailto:emhuzita,tait,thelma@din.uem.br), [quinaia@unicentro.br](mailto:quinaia@unicentro.br)\*

**Abstract** *The distributed software development has growing as a strategy adopted in organizations. Advantages can be obtained as better use of resources. On the other hand, problems appear due to the geographical dispersion and cultural differences. Therefore, the adoption of this strategy needs instruments to make it possible. In that context, DiSEN - Distributed Software Engineering Environment is presented, whose architecture is multilayers (dynamic, application and infrastructure). This environment offers resources to communication, persistence and cooperation to support the work of dispersed teams geographically.*

**Resumo** *O desenvolvimento distribuído de software tem se tornado, progressivamente, uma estratégia adotada em empresas de diversos setores. Se por um lado pode-se obter vantagem com melhor aproveitamento de recursos, por outro surgem problemas devido à dispersão física e diferenças culturais. Portanto, a adoção desta estratégia necessita de instrumentos para viabilizá-la. Nesse contexto, é apresentado o ambiente DiSEN (Distributed Software Engineering Environment), cuja arquitetura obedece o estilo em camadas (dinâmica, aplicação e infra-estrutura). Este oferece recursos para comunicação, persistência e cooperação para apoiar o trabalho de equipes dispersas geograficamente.*

### **1. Introdução**

Em busca de vantagem competitiva, diversas organizações optaram por distribuir o processo de desenvolvimento de *software* dentro de seu país, ou em outros países. Lidar com equipes constituídas por pessoas de diferentes localidades, culturas distintas e diferentes expectativas para trabalhar em projetos distribuídos pode, muitas vezes, gerar dificuldades (Herbsleb, 2001). Dentre estas dificuldades tem-se a necessidade de uma infra-estrutura adequada para comunicação, persistência e cooperação.

Para Carmel (1999), as principais características que diferenciam o desenvolvimento distribuído de *software* (DDS) do desenvolvimento co-localizado são: a dispersão geográfica, a dispersão temporal e as diferenças culturais. Portanto, os desafios apresentados pela distribuição da equipe envolvida no processo de desenvolvimento de *software* são significativos. Assim, torna-se necessário a utilização de ambientes de desenvolvimento que ofereçam o suporte adequado para esta nova

realidade. Segundo Damian (2006), embora já tenha sido formado um corpo de conhecimento sobre desenvolvimento global, ainda existe um significativo entendimento a ser alcançado, métodos e técnicas a serem desenvolvidas e práticas a serem amadurecidas.

Neste artigo é apresentado o ambiente DiSEN com as soluções implementadas no que tange a comunicação, persistência e cooperação, como parte do projeto de pesquisa realizado na Universidade Estadual de Maringá com financiamento do CNPq. Para a sua apresentação, este artigo se encontra estruturado conforme segue: a seção 2 trata a distribuição no desenvolvimento de *software*; os trabalhos relacionados são abordados na seção 3; o ambiente DiSEN é apresentado na seção 4 contendo uma visão da arquitetura e dos elementos de apoio à infra-estrutura de comunicação, persistência e cooperação; a avaliação é apresentada na seção 5 e, finalmente, na seção 6 são realizadas as considerações finais.

## **2. A Distribuição no Desenvolvimento de *Software***

Um Ambiente de Desenvolvimento de *Software* (ADS) deve se preocupar com: o apoio às atividades individuais e ao trabalho em grupo, o gerenciamento do projeto, o aumento da qualidade geral dos produtos e o aumento da produtividade, permitindo ao engenheiro de *software* acompanhar o projeto e medir, por meio de informações obtidas ao longo do desenvolvimento, a evolução dos trabalhos (Travassos, 1994). Os ambientes de desenvolvimento de *software* que provêm suporte a processos distribuídos devem coordenar usuários de diversas equipes, notificando uma equipe sobre as atividades realizadas pelas outras e permitindo a troca de artefatos entre as equipes (Ben-Shaul e Kaiser, 1998).

Na literatura corrente, o conceito de gerência distribuída do processo aparece intimamente relacionado ao conceito de distribuição do desenvolvimento (Damian, 2004). Processos distribuídos representam, geralmente, projetos com desenvolvimento distribuído. Segundo Gumm (2005), pode-se distribuir: equipes, habilidades, indivíduos, conhecimento, interesses, poder, organizações, informações, tomada de decisão, processo e artefatos.

No contexto dessa pesquisa, considera-se que o desenvolvimento distribuído de *software* ocorre por meio de atividades de projeto executadas por membros de equipes que podem estar em locais geograficamente distintos. Além disto, os gerenciadores, que são responsáveis por prover o suporte necessário, podem estar distribuídos, levando ao ambiente distribuído de desenvolvimento. Com isto, procura-se tirar o máximo proveito da distribuição almejando um produto de qualidade com custos reduzidos.

## **3. Trabalhos relacionados**

Devido ao crescente número de empresas que adotam o desenvolvimento de *software* de maneira distribuída, é possível encontrar vários projetos de pesquisa tais como: GENESIS (*Generalized Environment for proCESs management in cooperatIve Software engineering*) (Aversano *et al.*, 2003); Odyssey Share (Werner *et al.*, 2003); Gossip (Babak, 2000) e MILOS (*Minimally Invasive Longterm Organizational Support*) (Maurer e Martel, 2002).

Analisando as propostas de ambientes acima listados, podem-se enumerar algumas características necessárias em ambientes de desenvolvimento distribuído. Em

## WDDS 2007

### *I Workshop de Desenvolvimento Distribuído de Software*

Steinmacher *et al.* (2005) as forças centrípetas para a formação de equipes distribuídas, apresentadas por Carmel (1999), foram mapeadas para características desejáveis em ambientes de desenvolvimento distribuído de *software* (ADDS). Essas características são: definição de uma arquitetura de produtos; suporte à metodologia de desenvolvimento; suporte à persistência de dados; suporte à gerência de recursos, de artefatos, de processos e de projetos; infra-estrutura de comunicação e tecnologia de colaboração.

De acordo com Steinmacher *et al.*(2005), dentre os ambientes listados, Odissey Share é um dos que apresenta propostas para apoiar as características desejáveis, além de oferecer uma grande gama de mecanismos de colaboração nesse ambiente. No entanto, o Odissey Share não trata de forma completa a especificação do mecanismo de persistência de dados. O projeto GENESIS não possui definições relacionadas à arquitetura de produtos, além de não possuir definições muito claras do mecanismo de suporte à persistência, sendo tal suporte feito apenas por um banco de dados relacional.

O Gossip apresenta todos os seus mecanismos de suporte às características desejáveis baseados em um mecanismo central de *awareness*. Estes mecanismos possuem definições gerais, algumas vezes sendo muito vagas. Não foram encontradas definições com relação às características de suporte à Arquitetura de Produtos, Metodologia de Desenvolvimento e Gerência de Recursos. Por último, o projeto MILOS não apresenta preocupação com o suporte à Arquitetura de Produtos e Gerenciamento de Artefatos, e apenas define seu suporte à Persistência como sendo um banco de dados Orientado a Objetos. O MILOS se destaca pelo suporte à Metodologia de Desenvolvimento, sendo totalmente voltada para processos ágeis (XP, Scrum).

#### **4. Ambiente DiSEN**

Para a construção do ambiente partiu-se primeiramente da elaboração de uma arquitetura para o mesmo (Pascutti, 2002). Na seqüência, foram abordados aspectos relativos ao gerenciamento de projeto em DDS e a infra-estrutura adequada para comunicação, persistência e cooperação. Com isto buscou-se atender às características apresentadas na seção anterior (Steinmacher, *et al.* 2005).

##### **4.1 Arquitetura do DiSEN**

Como pode ser visualizada na Figura 01, a arquitetura é dividida em camadas: dinâmica, aplicação e infra-estrutura. A camada Dinâmica permite a manutenção dos componentes de *software* e serviços de forma dinâmica, em tempo de execução. A camada de Aplicação oferece o apoio necessário ao gerenciador de objetos, à MDSODI (Metodologia para Desenvolvimento de *Software* DIstribuído), ao gerenciador de *workspace* e ao gerenciador de agentes de *software*.

O Gerenciador de Objetos é responsável pelo controle e gerenciamento do ciclo de vida dos artefatos (tais como: diagramas, modelos, códigos fonte). O Gerenciador de *Workspace* é responsável pelo controle e gerenciamento da edição cooperativa de documentos e itens de *software*. O Gerenciador de Agentes é responsável pela criação, registro, localização, migração e destruição de agentes. O repositório é o responsável pelo armazenamento de dados, modelo e conhecimento.

A camada de infra-estrutura apóia as tarefas de persistência, nomeação e concorrência e contém, também, o canal de comunicação. Este último é responsável

pela comunicação entre os elementos da arquitetura e constituído pelo *middleware* e pelo *middle-agent*. O *middleware* é o responsável quando a comunicação ocorrer unicamente por meio de objetos e, quando ela envolver agentes, a comunicação é gerenciada pelo *middle-agent*.

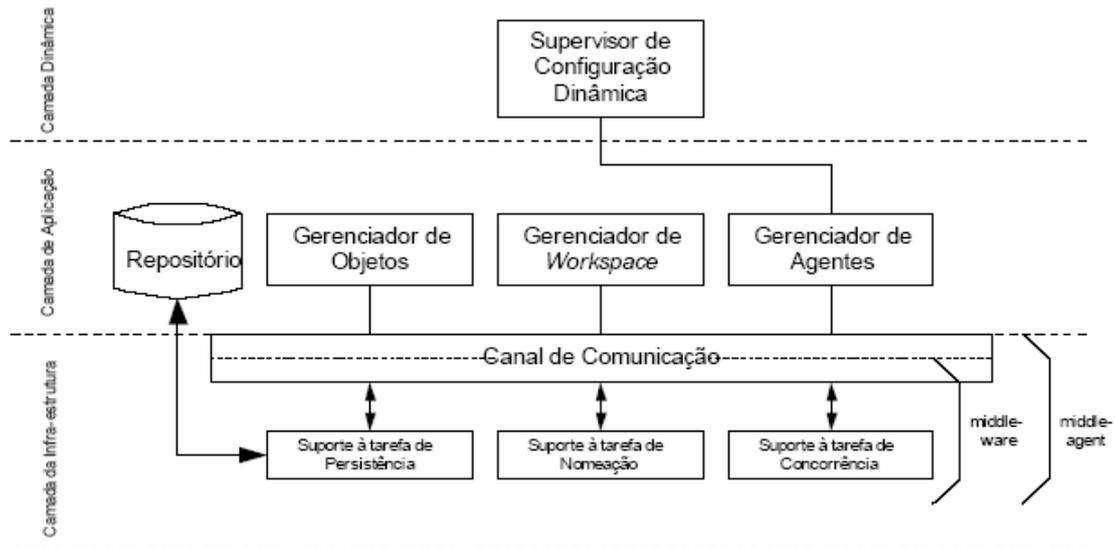


Figura 01. Arquitetura do DiSEN (Pascutti, 2002)

### Uma Instanciação de Projeto no DiSEN

Uma ilustração do ambiente DiSEN é apresentada na Figura 02. Assim, podem-se visualizar as interfaces para o cadastramento de projetos com as fases, as atividades e os participantes.

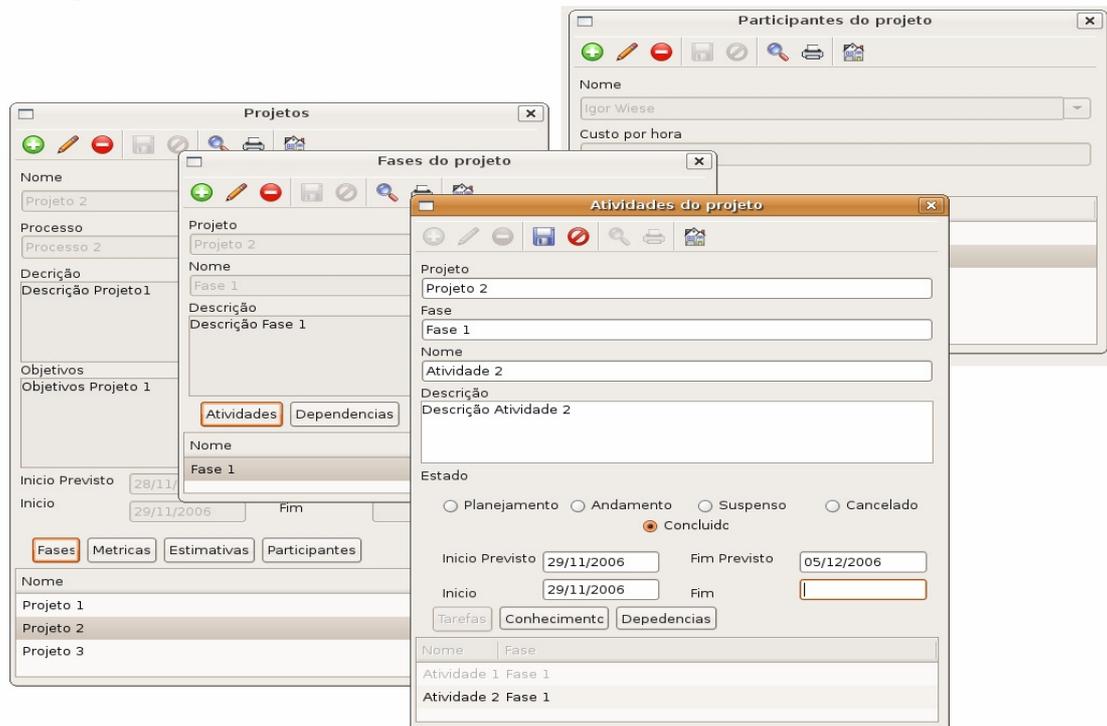


Figura 02. Interfaces para Gerenciamento de Projetos no DiSEN

Para iniciar a utilização do DiSEN, o usuário deve efetuar sua entrada no sistema e este será validado de acordo com o seu perfil. Para a construção da interface gráfica de usuário, em tempo de execução, foi desenvolvido um *framework* que facilita a validação e o povoamento dos campos dos formulários manipulados na área de trabalho.

#### **4.2 Infra-estrutura de apoio ao ambiente DiSEN**

A comunicação, o gerenciamento de atividades e de artefatos são elementos que devem estar presentes quando se quer estabelecer a colaboração no desenvolvimento de projetos. A seguir, são apresentadas as soluções adotadas na construção do ambiente DiSEN no que se refere à infra-estrutura necessária à cooperação, persistência e comunicação.

##### ***Apoio à Cooperação***

As características de um ADDS como: a formação das equipes virtuais e o necessário uso de ferramentas de apoio para o desenvolvimento de *software* foram consideradas para a composição do modelo de gerenciamento de projetos denominado *Distributed Software Engineering Environment-Project Management Model* (DiSEN-PMM) (Enami, 2006). Para esse modelo, a organização é tratada nos níveis estratégico, tático e operacional configurados nos gerentes: geral; local e de projeto juntamente com os engenheiros de *software* respectivamente. Ainda, para a interação entre os participantes bem como o gerenciamento das atividades sendo executadas, foi definido um modelo de cooperação denominado SPC (Sincronização, Percepção e Cooperação) (Pozza, 2006).

A especificação das atividades e participantes distribuídos por locais geograficamente dispersos, exige um controle mais rigoroso por parte do gerente de projetos de *software* que é apoiada pela ferramenta DIMANAGER (Pedras, 2003) a qual faz uso também de um mecanismo, que foi implementado utilizando agentes de *software*, e que considera o conhecimento, a habilidade e a disponibilidade das pessoas em diferentes locais para a alocação de recursos humanos (Huzita *et al.*, 2005).

Para o apoio ao desenvolvimento, tem-se a ferramenta Requisite (Batista, 2003), que auxilia na especificação de requisitos, por meio da criação de cenários e casos de uso. A ferramenta GeCA (Steinmacher *et al.*, 2006), por sua vez, permite obter os dados a partir de: (i) bancos de dados já existentes; (ii) códigos Java compilados; (iii) pacotes Java compilados; e (iv) diagramas de classe em UML, gerando uma estrutura de classes e pacotes capazes de gerar códigos de saída (SQL, XML, JavaBeans, formulários) e diagrama de classes. O DiSEN inclui, também, chat como um mecanismo de comunicação síncrona entre os participantes.

Para tratar da interoperabilidade entre as ferramentas, o DiSEN faz uso do IMART (*Interoperability Model of ARTifacts*) que proporciona os recursos para manipular as informações contidas nos artefatos por meio de um meta modelo. Esses recursos são importantes para permitir o trabalho cooperativo e a reutilização dos artefatos, bem como a integração de ferramentas no ambiente de desenvolvimento (Wiese e Huzita, 2006).

Para apoiar a reutilização de artefatos de *software*, o DiSEN possui um gerenciador de padrões de *software*. Com este gerenciador, pode-se reutilizar padrões de *software*, bem como cadastrar padrões e linguagens de padrões.

### ***Apoio à Persistência***

O apoio à persistência se encarrega de persistir as informações dos objetos de negócio e meta-dados envolvidos no desenvolvimento de *software*. O padrão de projeto *Data Access Object* (DAO) foi utilizado por permitir criar uma camada na aplicação separando as rotinas de acesso a banco de dados das rotinas de negócio de tal modo que a gravação dos objetos de negócio seja transparente.

A fim de atender às necessidades do DiSEN, foi utilizado um mecanismo de persistência que permite criar uma única conexão com vários bancos de dados; executar operações de escrita e leitura e verificar como se comporta em relação a: (i) vários bancos de dados homogêneo/heterogêneos no mesmo *host*, (ii) vários bancos de dados homogêneo/heterogêneos em *hosts* distintos – mesma LAN (*local area network*) e (iii) vários bancos de dados homogêneo/heterogêneos em *hosts* diferentes – WAN (*world area network*).

### ***Apoio à Comunicação***

O ambiente inclui os modelos de comunicação: cliente-servidor, entre servidores, servidor-cliente e entre clientes. A possibilidade de apoiar a comunicação servidor-cliente distingue o *middleware* do ambiente dos demais protocolos como o RMI (*Remote Method Invocation*), por exemplo, que não permite que o servidor envie uma informação ao cliente caso o mesmo não a solicite.

Optou-se por construir um *middleware* próprio, pois o DiSEN demanda por eventos de *Computer Supported Cooperative Work* (CSCW), tais como reuniões virtuais, interação entre desenvolvedores para a geração de artefatos, que necessita de um suporte adequado. Para o transporte e estabelecimento da comunicação entre os nós é utilizado *Socket* por permitir comunicação *full-duplex*. Desta maneira, os servidores podem ativar a comunicação com os clientes, enviar mensagens e instanciar objetos remotos. As mensagens trocadas pelo *middleware* são objetos serializáveis compactados para garantir um melhor aproveitamento da infra-estrutura física da rede.

## **5. Avaliação do ambiente DiSEN**

A avaliação é composta de duas fases. A primeira fase foi realizada no Laboratório de Engenharia de *Software* (LES/UEM) onde os recursos computacionais do projeto estão instalados e interligados. As funcionalidades do ambiente foram avaliadas pelos integrantes do grupo, os quais tiveram participação em partes distintas da construção do ambiente. Também, o modelo de gerenciamento de projetos e a ferramenta DIMANAGER foram validados por empresas desenvolvedoras de *software* que atestaram a sua viabilidade de utilização (Pedras, 2003; Enami, 2006).

A segunda fase encontra-se em formatação para envolver empresas, incluindo-se o APL (Arranjo Produtivo Local) de *software* da região, que avaliarão o uso do DiSEN no desenvolvimento de *software*. Algumas dessas empresas já manifestaram interesse em adotar o desenvolvimento distribuído em seus processos de trabalho.

## **6. Considerações finais**

Para a implementação do DiSEN, a partir de sua arquitetura, são utilizados: a linguagem Java, a construção de *frameworks* e a aplicação de padrões de projeto. Estas técnicas

## WDDS 2007

### *I Workshop de Desenvolvimento Distribuído de Software*

foram adotadas por/para permitir o reuso, o que facilita a manutenção, quer seja quando novos requisitos são adicionados, ou mesmo para dar continuidade à sua construção.

Fatores internos ao grupo de desenvolvimento, como dispersão geográfica, falta de coordenação e comunicação ineficiente, podem afetar a cooperação entre equipes remotas. Assim, o suporte oferecido pela cooperação, persistência e comunicação viabilizadas pela infra-estrutura do DiSEN contribuem para atender aos fatores que afetam o DDS. Dessa forma, o IMART, presente neste ambiente, apóia a integração de ferramentas, o trabalho cooperativo e a reutilização dos artefatos.

As tendências atuais apontam para a utilização da abordagem de aspectos e DDS. Assim, uma versão da ferramenta Requisite, incluindo aspectos, foi prototipada em (Silva, 2007); diretrizes para uma linguagem de descrição de arquitetura orientada a aspectos e critérios para estabelecer interações entre aspectos também estão sendo definidos. Encontram-se em desenvolvimento: as estratégias de riscos em DDS; uma infra-estrutura para realizar reuniões entre as equipes do DiSEN e um mecanismo para resolução de conflitos decorrentes de diferentes versões de um mesmo artefato.

Como trabalhos futuros estão previstos: uma ferramenta de apoio à estimativa de custos em DDS; exploração de *context awareness* em desenvolvimento distribuído e estabelecimento de estratégias de validação e verificação, incluindo-se testes.

Assim, visando oferecer o suporte adequado ao desenvolvimento distribuído de *software*, a equipe responsável pela construção do DiSEN adotou os elementos de apoio à comunicação, persistência e cooperação como suas linhas mestras.

**Agradecimentos:** ao CNPq pelo apoio financeiro (processo no. 506511/2004-9).

### **Referências**

- Aversano, L.; Cimitile, A.; Lucia, A.; Stefanucci, S.; Vilann, M. L., (2003) “Workflow Management in the GENESIS Environment”. RCST, University of Sannio.
- Babak, F. A. (2000) “Gossip: An Awareness Engine for Increasing Product Awareness in Distributed Development Projects”. In: International Conference on Advance Information Systems Engineering, Stockholm, 2000. Proceedings... p. 264-278.
- Batista, S. M. (2003) “Uma Ferramenta de Apoio à Definição de Requisitos da MDSODI no Contexto do Ambiente DiSEN”. Programa de Pós-Graduação em Informática da UFPr/UEM. Curitiba.
- Ben-Shaul, I.; Kaiser, G. (1998) “Federating Process-Centered Environments: the Oz Experience”. In *Automated Software Engineering*, 5, 1998, Proceedings...The Netherlands:Kluwer Academic Publishers, p. 97-132.
- Carmel, E. (1999) “Global *Software* Teams – Collaborating Across Borders and Time-Zones”. Prentice Hall.
- Damian, D. (2004) “Global *Software* Development”. In *Workshop de Desenvolvimento Distribuído de Software*. 2004. Porto Alegre. Anais...Porto Alegre: PUC-RS.
- Damian, D.; Moitra, D. (2006) “Global *Software* Development: How far have we come”. *IEEE Computer*. 23(5), p.17-19.
- Enami, L. N. M. (2006) “Um Modelo de Gerenciamento de Projetos para um Ambiente de Desenvolvimento Distribuído de *Software*”. Dissertação (Mestrado) Programa de Pós-Graduação em Ciência da Computação. UEM-DIN. Maringá/PR,Brasil.

**WDDS 2007**  
*I Workshop de Desenvolvimento Distribuído de Software*

- Gumm, D. C. (2005) “The Phenomenon of Distribution in *Software* Development Projects: A Taxonomy Proposal”. European Conference on Information Systems (EMCIS 05), Information Institute.
- Herbsleb, J.D.; Moitra, D. (2001) “Global *Software* Development”. *IEEE Software*, 18(2), March/April, p. 16-20.
- Huzita, E. H. M. ; Tait, T. F. C. ; Lima, F. de . (2005) “Um Mecanismo de Suporte ao Gerenciamento de Recursos Humanos no Desenvolvimento Distribuído de *Software*”. In: In CACIC 2005. Concórdia. Workshop de Ingeniería de *Software* y Base de Datos (WISBD).
- Maurer, F.; Martel, S. (2002) “Process Support for Distributed Extreme Programming Teams”. In: International Conference on *Software* Engineering, Workshop on Global *Software* Development. Orlando.
- Pascutti, M.C.D. (2002) “Uma Proposta de Arquitetura de um ambiente de Desenvolvimento de *Software* Distribuído Baseado em Agentes”. Dissertação (Mestrado). Universidade Federal do Rio Grande do Sul – Instituto de Informática. Porto Alegre/RS, Brasil.
- Pedras, M.E.V. (2003) “Uma Ferramenta de Apoio ao Gerenciamento do Desenvolvimento de *Software* Distribuído”. Dissertação (Mestrado). Universidade Federal do Paraná/Universidade Estadual de Maringá. Maringá-PR, Brasil.
- Pozza, R.S.; Huzita, E.H.M. (2006) “Modelo de Cooperação para um Ambiente Distribuído de Desenvolvimento de *Software*”. In: XXXII Congresso Latino Americano de Informática – CLEI, 2006, Santiago (Chile). Anais... (em cd).
- Silva, L.P. (2007) “A Engenharia de Requisitos Orientada a Aspectos: a Abordagem DAORE”. Dissertação (Mestrado). Programa de Pós-Graduação em Ciência da Computação. Universidade Estadual de Maringá-DIN. Maringá/PR, Brasil.
- Steinmacher, I. F.; Wiese, I.S.; Pozza,R.S.; Huzita, E.H.M. Amorim, E.F; Pascutti, M.C. D. (2005) “Uma Proposta de Arquitetura para Ambientes de Desenvolvimento Distribuído de *Software*”. In CACIC 2005. Concórdia – Entre Rios. Workshop de Ingeniería de *Software* y Base de Datos (WISBD).
- Steinmacher, I. F.; Schiavoni, F. L; Amorim, E. F.; Wiese, I.S.; Huzita, E. H. M. (2006) “GeCA: Uma Ferramenta de Engenharia Reversa e Geração Automática de Código”. Simpósio Brasileiro de Sistemas de Informação (SBSI) 2006 Proceedings (em cd). Curitiba.
- Travassos, G. H. (1994) “O Modelo de Integração de Ferramentas da Estação TABA”. Tese (Doutorado). COPPE/UFRJ, Rio de Janeiro.
- Werner, C.; Mangan, M.; Murta, L.; Pinheiro, R.; Mattoso, M.; Braga, R.; Borges, M. (2003) “OdysseyShare: an Environment for Collaborative Component-Based Development”. In: IEEE International Conference on Information Reuse and Integration, 2003, Las Vegas. 2003. Proceedings... v. 1. p. 61-68.
- Wiese, I.S.; Huzita, E.H.M. (2006) “IMART: An Interoperability Model for Artifacts of Distributed *Software* Development Environments”. In: IEEE International Conference on Global *Software* Engineering” 2006. Florianópolis. v.1, p. 255-256.