

# Infraestrutura de Apoio ao Reuso e Gerenciamento de Famílias de Frameworks Transversais

Projeto submetido ao Edital MCT/CNPq 09/2010 - PDI

**Prof. Dr. Valter Vieira de Camargo – valter@dc.ufscar.br (Coordenador)**

**Profa. Dra. Rosângela Aparecida Delloso Penteado (pesquisador)**

*Departamento de Computação da Universidade Federal de São Carlos (UFSCar)*

*Caixa Postal 676 – 13565-905 – São Carlos - Brasil*

**Prof. Dr. Ricardo Argenton Ramos (pesquisador)**

*Colegiado em Engenharia de Computação da Universidade Federal do Vale do São Francisco – UNIVASF Juazeiro, Bahia, Brasil.*

## **Alunos diretamente envolvidos:**

Thiago Gottardi (mestrado)

Rhafik Daud Gonzáles (IC-cnpq)

Carlos Eduardo Tibério (IC-cnpq)

## **Resumo**

Framework Transversal é um tipo especial de framework orientado a aspectos que encapsula apenas um interesse transversal, como persistência, concorrência e regras de negócio. Cada um desses interesses possui subinteresses que podem agir como características (*features*) de uma família de FTs, permitindo que diferentes membros de uma família possam ser gerados por meio da composição de diferentes características. Este projeto tem por objetivo a construção de uma infraestrutura integrada para desenvolvimento de software que contenha ferramentas e técnicas de apoio ao reuso e ao gerenciamento de Famílias de Frameworks Transversais (FTs). Essa infraestrutura utilizará famílias de FTs que podem ser reusadas durante o desenvolvimento de novas aplicações e/ou evolução de aplicações existentes, de forma a reduzir o tempo de desenvolvimento e aumentar a qualidade das aplicações. Outro diferencial é que todo o trabalho deve ser feito apenas utilizando modelos, elevando o nível de abstração que deve ser empregado. Ganhos de produtividade poderão ser obtidos em decorrência do reuso de projeto e código das famílias de FTs já previamente testadas. A infraestrutura será desenvolvida de forma a apoiar as fases/disciplinas/atividades de um processo chamado ProFT/PU, que é um processo de desenvolvimento de software orientado a aspectos iterativo e incremental que já foi especificado. Um conjunto de ferramentas de apoio e de métodos será desenvolvido para auxiliar na construção de membros, na instanciação dos membros e no acoplamento dos FTs, bem como em seu gerenciamento durante e após o processo. Com os resultados obtidos neste projeto é possível aumentar a eficiência de desenvolvedores de software e também a qualidade das aplicações desenvolvidas em termos de modularidade, reuso e manutenibilidade.

## 1 – INTRODUÇÃO

---

A Programação Orientada a Aspectos (POA) (Kiczales *et al.*, 1997) é usada desde o final da década de 90 para modularizar de forma mais adequada os “interesses transversais” de um sistema, que até então ficavam misturados e espalhados pelo código dos interesses-base sem que fosse possível organizá-los em módulos independentes. O entrelaçamento e espalhamento de código acabavam culminando em problemas de modularização e consequentemente, manutenção e reuso. Na terminologia da POA, os “interesses-base” referem-se à funcionalidade principal do sistema e os “transversais” referem-se a restrições globais e a requisitos não-funcionais, como por exemplo, persistência, distribuição, autenticação, controle de acesso, criptografia e concorrência (Kiczales *et al.*, 2001). Com as novas abstrações fornecidas pela POA é possível implementar separadamente os interesses-base e os interesses transversais, o que até então era difícil somente com a programação orientada a objetos. Assim, é possível projetar um sistema de forma assimétrica, isto é, com duas partes distintas, uma “parte base” e uma “parte transversal” que devem ser posteriormente combinadas para a geração da aplicação final.

Com o surgimento da POA a maior parte das pesquisas conduzidas no contexto da orientação a objetos tiveram que ser revistas sob a óptica das abstrações dessa nova técnica de programação. Um exemplo são os chamados “Frameworks Orientados a Aspectos” (FOAs) como uma nova proposta para a reutilização de interesses transversais ou como uma forma alternativa de modularizá-los (Lazanha *et al.*, 2010; Oliveira *et al.*, 2010; Kulesza *et al.*, 2006; Coelho *et al.*, 2008; Camargo e Masiero, 2005a; Zanon *et al.*, 2009; Couto *et al.*, 2005; Hanenberg *et al.*, 2004; Huang *et al.*, 2004; Shah e Hill, 2004; Constantinides e Elrad, 2001; Pinto *et al.*, 2002; Rashid e Chitchyan, 2003; Soares *et al.*, 2006; Vanhaute *et al.*, 2001; Huang *et al.*, 2004; Mortensen e Ghosh, 2006a; Mortensen e Ghosh, 2006b; Soares, 2004).

Framework Orientado a Aspectos (FOA) refere-se a qualquer framework que possua aspectos em sua arquitetura, entretanto, neste projeto de pesquisa, o enfoque é em um tipo particular de Framework Orientado a Aspectos denominado aqui de “Framework Transversal” (FT) (Camargo e Masiero, 2005). Um FT é um framework que encapsula apenas um interesse transversal, como persistência, segurança, concorrência ou alguma regra de negócio, e é o tipo mais comum de FOA existente.

Além dos frameworks, outra linha de pesquisa influenciada pela POA foram as Linhas de Produtos ou Famílias de Produtos de Software (Clements e Northrop, 2001). Uma família de produtos possui uma base comum que é reusada por todas as versões do produto, enquanto as variações específicas para cada membro da família são devidamente gerenciadas, organizadas em módulos e compostas para gerar as versões específicas (Clements e Northrop, 2001). Vários autores têm investigado como características (*features*) podem ser implementadas com os conceitos da POA para proporcionar mais flexibilidade em sua composição e melhorar a qualidade do sistema final (Fuentes *et al.*, 2009; Apel *et al.*, 2006; Mezini e Osterman, 2004; Alves *et al.*, 2007; Kulesza *et al.*, 2007).

Outro conceito importante neste projeto é o de Desenvolvimento Dirigido a Modelos (MDD). No MDD, modelos não são apenas artefatos que servem para ilustrar ou comunicar ideias. Ao contrário, são efetivamente utilizados como entrada para geradores de código que produzem aplicações completas. MDD é a combinação de programação generativa, linguagens específicas de domínio e transformações de software. Seu objetivo é reduzir a distância semântica entre o problema e a implementação/solução, por meio de modelos de alto nível que protegem os desenvolvedores das complexidades da plataforma de implementação (France e Rumpe, 2007). No MDD, modelos são usados para se expressar conceitos do domínio de forma mais efetiva, enquanto transformações geram automaticamente os artefatos que refletem a solução expressa nos modelos (Schmidt, 2006; Pastor e Molina, 2007).

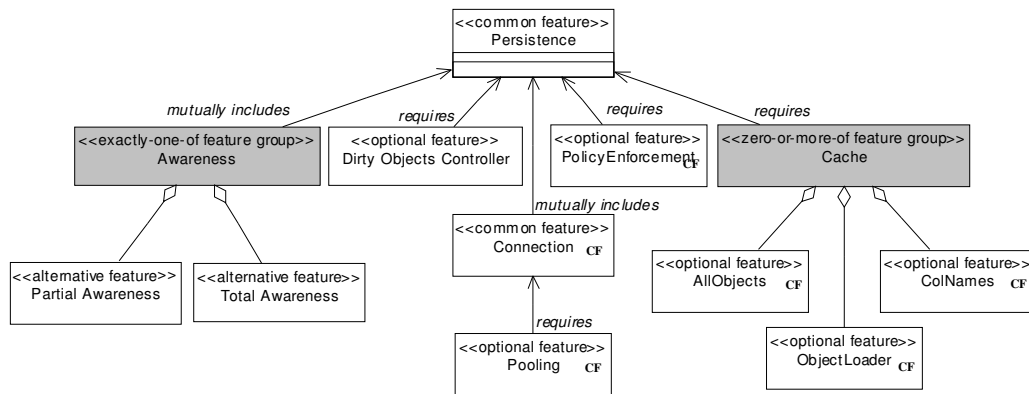
## **1.1 - Experiência na Área de Pesquisa do Projeto**

O proponente deste projeto de pesquisa desenvolveu 12 (doze) FTs agrupados em três famílias – uma de persistência, uma de segurança e uma de regras de negócio (Camargo e Masiero, 2008b; Camargo *et al.*, 2003; Camargo, 2006, Ramos *et al.*, 2004, De Grandi *et al.*, 2008). A família de FTs de persistência, por exemplo, permite gerar cento e noventa e dois (192) FTs distintos por meio da combinação de suas características.

Projetar FTs como famílias de produtos (ou linha de produtos) tem a vantagem de cada membro da família possuir apenas as características realmente necessárias à aplicação que está sendo desenvolvida (Camargo e Masiero, 2008b). Isso é o oposto do que ocorre com os frameworks tradicionais (Fayad e Johnson, 2000), em que todo o framework faz parte da arquitetura final da aplicação, mesmo se apenas um subconjunto de suas variabilidades é necessário. Dessa forma, pode-se evitar que grande parte da aplicação final

desenvolvida com base no FT contenha “código morto”, o que pode levar a problemas de manutenção, evolução e reúso.

Na Figura 1 é mostrado um diagrama de características da família de FTs de Persistência (Camargo e Masiero, 2008b) seguindo a notação proposta por Gomaa (2004). Cada característica representada por uma caixa vazia do diagrama é um pacote (*package*) de software que contém classes e aspectos de um determinado interesse ou subinteresse. As que estão marcadas com a sigla FT são genéricas, isto é, são Frameworks Transversais, e aquelas que não possuem essa marcação são características que só podem ser reusadas dentro da própria família. As características que estão representadas por caixas cinzentas são grupos e não possuem representação como software, são apenas conceituais e representam um agrupamento de características alternativas – estereótipo <<exactly-one-of-feature group>> – ou opcionais – estereótipo <<zero-or-more-of-feature group>>. A etiqueta *mutually includes* significa que a escolha de uma determinada característica inclui automaticamente a característica apontada pela seta, e a etiqueta *requires* indica que a escolha da característica exige que a característica apontada pela seta já esteja presente no projeto.



**Figura 1 – Diagrama de Características da Linha de Produtos de FTs de Persistência**

O tema de pesquisa “Famílias de FTs” possui diversas vertentes que são pesquisadas pelo proponente deste projeto. Por exemplo, um detalhe importante é fornecer meios eficazes para que as variabilidades dos FTs e seus mecanismos de composição abstratos sejam adequadamente projetados com padrões (Camargo e Masiero, 2008a, Fujii *et al.*, 2008) e fiquem evidentes em seus modelos de análise e projeto (Uetanabara *et al.*, 2010; Uetanabara *et al.*, 2009a; Uetanabara *et al.*, 2009b; Gottardi *et al.*, 2010), pois isso pode auxiliar nos processos de reúso e manutenção do framework e das aplicações desenvolvidas com o apoio desses FTs (Camargo e Masiero, 2007). A adequada adoção de uma arquitetura de referência

também tem influência direta na qualidade do FT e das aplicações geradas com o seu apoio (Lazanha *et al.*, 2010).

Outra vertente de pesquisa é com relação ao controle de versões das aplicações desenvolvidas com base em FTs. Como os FTs tornam-se mais efetivos quando há um conjunto deles disponível para reuso, pode ocorrer de uma aplicação ter sido desenvolvida com o apoio de vários FTs. Quando esses frameworks são utilizados em conjunto é interessante que exista um controle de versões específico (Arimoto *et al.*, 2007; Arimoto *et al.*, 2008), pois podem existir várias versões deles acoplados a uma aplicação e qualquer modificação em uma versão pode afetar os outros FTs.

A evolução de sistemas legados existentes é outra vertente de pesquisa que merece destaque. A reestruturação de sistemas existentes para utilizarem FTs exige que determinados interesses transversais sejam identificados nessas aplicações e substituídos por FTs. Isso deve ser apoiado por ferramentas e técnicas adequadas (Parreira *et al.*, 2010a, Parreira *et al.*, 2010b, Costa *et al.*, 2009a; Costa *et al.*, 2009b)

A modularização de frameworks de aplicação com FTs também tem relação com o presente projeto. Frameworks de aplicação são softwares complexos que possuem vários interesses transversais em sua arquitetura (Fayad e Johnson, 2000). A adequada modularização desses frameworks com FTs resulta em melhores níveis de manutenção para o framework e para as aplicações que são desenvolvidas com o seu apoio (Oliveira *et al.*, 2010; Zanon *et al.*, 2009).

A partir do aparato tecnológico desenvolvido e citado nos parágrafos anteriores, também houve preocupação em se criar um processo que facilitasse a utilização de FTs. Assim foi especificado o processo ProFT/PU, que é uma adaptação do Processo Unificado (Larman, 2004) e que possui atividades específicas para o tratamento de aspectos e FTs ao longo do processo de desenvolvimento (Camargo e Masiero, 2008c). Esse processo foi delineado para mostrar como a presença de FTs influencia algumas atividades de um processo de desenvolvimento. Enquanto alguns autores apresentam processos que se concentraram apenas na parte transversal da aplicação (Clarke e Baniassad, 2005; Jacobson e Ng, 2004; Baniassad *et al.*, 2006; Araújo e Moreira, 2003; Araújo *et al.*, 2002), ProFT/PU mostra como conduzir o desenvolvimento da parte base da aplicação e da parte aspectual e como essas partes se inter-relacionam à medida que o processo progride.

Um estudo anterior também mostrou que FTs são efetivos quanto à produtividade de desenvolvedores de software. Um estudo comparativo foi conduzido no sentido de comparar a produtividade de desenvolvedores que utilizam um framework orientado a aspectos e um framework orientado a objetos de persistência (Camargo *et al.*, 2006) e os resultados foram satisfatórios a favor da tecnologia de aspectos.

Novas pesquisas também estão sendo conduzidas no sentido de avaliar como a presença de FTs influencia um processo de desenvolvimento dirigido a modelos (Camargo *et al.*, 2009).

## **1.2 - Enquadramento do Problema nos Grandes Desafios da Computação**

Na seção anterior mostrou-se que existem trabalhos na literatura que visam a dar apoio apenas a partes de um processo de desenvolvimento. Entretanto, não foi encontrada uma infraestrutura que forneça várias visões em diferentes níveis de abstração de um processo completo de reuso e gerenciamento de famílias de FTs. Alguns desafios que são encontrados quando se pretende apoiar um processo como esse são:

1. **Desenvolvimento Dirigido a Modelos.** Até o momento, não é de conhecimento do autor deste projeto um ambiente ou ferramenta que dê apoio ao reuso e gerenciamento de FTs com base em modelos. Ferramentas que fornecem apoio apenas em nível de código não aproveitam todo o potencial de reuso, prejudicam a produtividade e podem levar a problemas de manutenção.
2. **Armazenamento e Mineração de Famílias de FTs.** Embora vários autores já tenham trabalhado com FTs (Couto *et al.*, 2005; Hanenberg *et al.*, 2004; Huang *et al.*, 2004; Shah e Hill, 2004; Constantinides e Elrad, 2001; Pinto *et al.*, 2002; Rashid e Chitchyan, 2003; Soares *et al.*, 2006; Vanhaute *et al.*, 2001; Huang *et al.*, 2004; Mortensen e Ghosh, 2006a; Mortensen e Ghosh, 2006b; Soares, 2004), não é encontrado na literatura o projeto de um repositório de FTs e técnicas para busca de um determinado FT que atenda aos requisitos de uma aplicação. É importante que exista uma forma de averiguar se as variabilidades/características disponíveis na família de FTs atendem aos requisitos da aplicação que deve ser desenvolvida. Para que famílias de FTs sejam adequadamente usadas durante um processo de desenvolvimento, é importante um apoio computacional que facilite essa tarefa.

3. **Gerenciamento de Processo de Reuso.** O Acoplamento de um FT a um código-base pode não ser uma tarefa simples dependendo da quantidade de características e variabilidades disponível. Além disso, é interessante que o engenheiro de aplicação consiga fazer todo o processo utilizando uma ferramenta bem como obtenha uma análise do impacto que o acoplamento ou desacoplamento do FT pode ocasionar. Couto e outros (2005) propuseram uma ferramenta que dá apoio ao processo de reuso de FTs, entretanto é bastante simples e não foi projetado para apoiar famílias de FTs; apenas para o interesse transversal de persistência.
4. **Gerenciamento de Pontos Afetados.** O acoplamento de um ou mais FTs em um código-base deve se feito com cautela, visto que podem afetar os mesmos pontos de junção. Além disso, modificações indiscriminadas podem alterar pontos em que os FTs estavam atuando, fazendo com que eles parem de operar. Há algumas pesquisas relacionadas com a interferência de aspectos, mas nenhum deles trata de um apoio computacional visual para o gerenciamento dessas interferências.
5. **Visualização Arquitetural.** Depois que vários FTs encontram-se acoplados a um código-base é imprescindível ter uma visão mais genérica da arquitetura do sistema e da arquitetura do próprio FT (Lazanha *et al.*, 2010). Alguns autores já propuseram notações arquiteturais e linguagens de descrição arquitetural (*ADLs – Architectural Description Languages*) em trabalhos anteriores (Krechetov *et al.*, 2006; Batista *et al.*, 2006; Chavez *et al.*, 2009), mas o apoio ferramental a essas notações ainda é escasso.
6. **Controle das versões.** Embora uma ferramenta para controle de versões para FTs já tenha sido desenvolvida (Arimoto *et al.*, 2008) em nosso grupo de pesquisa, ela não está integrada com os FTs. Assim, as manutenções que são feitas nos FTs e que levam a novas versões devem ser manualmente registradas na ferramenta.
7. **Sincronização dos Modelos.** Para que a documentação do software apóie de forma adequada todo o processo é interessante que constantes manutenções no código sejam refletidas nos modelos de projeto e análise. Assim, qualquer acoplamento ou desacoplamento de FT deve ser refletido em seus modelos automaticamente. Embora a sincronização de modelos UML com código orientado a objetos já seja feita coerentemente por vários ambientes de desenvolvimento, como por exemplo, Eclipse (Eclipse, 2010) e NetBeans (NetBeans, 2010), o mesmo não ocorre no contexto da POA.

Dessa forma, o objetivo neste projeto é o desenvolvimento de uma infraestrutura de apoio ao desenvolvimento de software que se utiliza de Famílias de FTs. Essas famílias poderão ser reusadas de forma controlada durante o desenvolvimento de forma a aumentar a qualidade das aplicações desenvolvidas. Além disso, toda manipulação na infraestrutura deverá ser feita por intermédio de modelos, assim pretende-se elevar o nível de abstração em que o engenheiro de software trabalha.

Dentre os seis “Grandes Desafios de Pesquisa para a Computação no Brasil”, identificados pela SBC (Sociedade Brasileira de Computação), o sexto é o “Desenvolvimento Tecnológico de Qualidade: Sistemas Disponíveis, Corretos, Seguros, Escaláveis, Persistentes e Ubíquos (Carvalho *et al.*, 2010). No relatório apresentado pela SBC, esse desafio está relacionado a seis tópicos de pesquisa, entretanto, este projeto está diretamente relacionado aos seguintes tópicos:

- **Desenvolvimento e avaliação de modelos e ferramentas de modelagem de sistemas de software com base teórica sólida.** A interface de desenvolvimento fornecida pela infraestrutura proposta deverá ser baseada em modelos. Esses modelos deverão ser validados quanto a sua correteza semântica e estar em um nível de abstração adequado para permitir que o desenvolvedor trabalhe em um nível de abstração mais alto do que o código fonte. O proponente deste projeto possui alguns trabalhos em desenvolvimento na área de modelos (Camargo *et al.*, 2009, Uetanabara *et al.*, 2010, Lazanha *et al.*, 2010; Gottardi *et al.*, 2010).
- **Desenvolvimento de ferramentas de apoio ao processo de implementação e evolução de software.** A infraestrutura proposta pode ser vista como um ambiente ou um conjunto de ferramentas que permitirão tanto o desenvolvimento quanto a evolução de um software existente. A parte de evolução de software existente será apoiada por ferramentas que permitem analisar automaticamente o código de sistemas existentes e gerar modelos anotados. O proponente deste projeto possui trabalhos em desenvolvimento nesta linha (Parreira *et al.*, 2010a, Parreira *et al.*, 2010b, Costa *et al.*, 2009a; Costa *et al.*, 2009b).

A construção da infraestrutura proposta atende diretamente a esses dois tópicos citados pela SBC.



## 2 - OBJETIVOS E METAS A SEREM ALCANÇADOS

---

Como já comentado, o objetivo é a criação de uma infraestrutura integrada que forneça apoio ferramental e teórico para a **reutilização** de famílias de FTs durante o desenvolvimento de aplicações e também ao **gerenciamento** de FTs que já se encontram acoplados (reutilizados) em uma aplicação que está em operação. Essa infraestrutura deverá integrar ferramentas, artefatos e técnicas que fornecem apoio às fases de elaboração e construção do processo ProFT/PU (Camargo e Masiero, 2008c). Outro ponto importante desse infraestrutura é que as interfaces de manipulação com os FTs serão dirigidas a modelos, assim, todas as tarefas de desenvolvimento e manutenção serão feitas por meio de modelos (France e Rumpe, 2007; Schmidt, 2006; Pastor e Molina, 2007).

O objetivo mais amplo deste projeto de pesquisa é aumentar a qualidade das aplicações que são desenvolvidas em termos de modularidade, reuso, manutenibilidade e também a eficiência de desenvolvedores de software. A eficiência será alcançada mediante a utilização de técnicas de reuso de análise, projeto e código. A qualidade das aplicações finais também deverá ser melhorada com o reuso, uma vez que os artefatos/técnicas envolvidas já foram previamente testados e aprovados.

Outro objetivo é fazer com que o departamento de computação da UFSCar se torne referência nacional na área de DSOA (Desenvolvimento de Software Orientado a Aspectos). Também pretende-se fortalecer o colegiado em engenharia da computação da UNIFASF por meio de reuniões, workshops e até mini-cursos realizados em ambas as universidades.

## 3 – METODOLOGIA EMPREGADA PARA CONDUÇÃO DO PROJETO

---

A estratégia adotada para desenvolvimento deste projeto é dividi-lo em subprojetos que poderão ser alocados a alunos de iniciação científica, mestrado e treinamento técnico. Como a ideia da infraestrutura é apoiar determinadas fases, disciplinas e atividades do ProFT/PU, esse serve de arcabouço conceitual para a condução do projeto.

Na Tabela 1 encontram-se propostas de subprojetos que poderão compor este projeto. Ressalta-se que isso é uma proposta e que esses subprojetos podem ser modificados em consequência de fatores como número de alunos, investigações mais detalhadas de trabalhos da área, lançamentos tecnológicos mais recentes etc.

Cada SP# é um subprojeto de mestrado (**M**), iniciação científica (**IC**) ou treinamento Técnico (**DT**). Cada uma das colunas da Tabela 1 refere-se a uma fase do ProFT/PU e os projetos encontram-se explicitados de acordo com a fase que eles mais apóiam. Quando um projeto apóia/envolve mais de uma fase, a coluna é mais larga, englobando essas fases. Os números dos subprojetos não representam ordem cronológica de desenvolvimento.

No subprojeto 1 (SP1 – Armazenamento e Mineração de Famílias de FTs), o objetivo é o desenvolvimento de um repositório para armazenamento de famílias de FTs. Deverá ser elaborado um metamodelo adequado ao armazenamento e que facilite a recuperação dos FTs, deve ser genérico o suficiente para descrever diferentes FTs, como segurança, persistência e concorrência. Com isso outros FTs, desenvolvidos por outros grupos de pesquisa, também podem ser adicionados nesse repositório, como por exemplo Distribuição e Gerenciamento de Transações (Soares *et al.*, 2006), Concorrência (Constantinides e Elrad, 2001) e Segurança (Sha e Hill, 2004; Vanhaute *et al.*, 2001). Para viabilizar essa integração deverá ser desenvolvido um “protocolo” que os FTs devem seguir para que possam ser armazenados e publicados nesse repositório. Esse protocolo deverá estipular interfaces que os FTs deverão se adequar para que após o armazenamento também possam ser minerados (buscados) com facilidade. O desenvolvimento de arquiteturas de referência para FTs (Lazanha *et al.*, 2010) também pode auxiliar nesse processo, pois pode fornecer uma base comum para se projetar FTs. Este subprojeto deve apoiar principalmente a fase de construção do ProFT/PU, como é mostrado na Tabela 1.

**Tabela 1 – Subprojetos que Compõem a Infraestrutura Integrada Proposta**

Fases do ProFT/PU		
Elaboração	Construção	Manutenção
SP4) Visualização Arquitetural (DT3)	SP1) Armazenamento e Mineração de Famílias de FTs (M)	SP5) Controle de Versões (IC) SP7)Manutenção com apoio de Modelos Anotados (M)
SP2) Gerenciamento do Processo de Reuso (Instanciação e acoplamento) (2 bolsas DT3) SP3) Gerenciamento de Atuação de FTs (IC) SP6) Sincronização de Modelos (IC) SP8) Checagem da Semântica de Modelos (M) SP10) Gerenciamento de FTs com Modelos (M)		
SP9) Desenvolvimento da Arquitetura da Infraestrutura (2 bolsas DT3)		

No subprojeto 2 (SP2 – Gerenciamento do Processo de Reúso), o objetivo é a criação de um apoio ferramental que auxilie no processo de reuso das famílias de FTs. Essa ferramenta deverá ser integrada com o repositório desenvolvido no SP1 e deverá permitir que primeiramente as características de uma determinada família sejam recuperadas a partir dos requisitos da aplicação que está sendo desenvolvida (Camargo e Masiero, 2008b). Nesse momento, a ferramenta deverá guiar o desenvolvedor no processo de composição das características para montar um FT apenas com as características exigidas. Nesse processo a ferramenta também deve indicar claramente as características que são alternativas, opcionais e obrigatórias. Depois disso, a ferramenta também deve facilitar o processo de reuso (instanciação e composição) do FT com o código-base. Essa ferramenta também deve informar possíveis conflitos entre os pontos de atuação dos FTs. A parte de controle de interferência entre aspectos existente no SP2 será realizada pelo subprojeto 3 de Iniciação Científica (SP3 – Gerenciamento de Atuação de FTs).

No subprojeto 4 (SP4 – Visualização Arquitetural) a ideia é elaborar ou adaptar alguma linguagem de descrição arquitetural e construir uma ferramenta para visualizar graficamente a arquitetura do sistema com os FTs que se encontram acoplados a ele (Lazanha *et al.*, 2010).

No subprojeto 5 de Iniciação Científica (SP5 – Controle de Versões) o objetivo é evoluir e integrar a ferramenta de controle de versões de FTs já desenvolvida em nosso grupo (Arimoto *et al.*, 2008) com as ferramentas desenvolvidas nos projetos 1 2 e 3. Essa ferramenta deve gerenciar adequadamente as várias configurações permitidas e não permitidas entre versões de FTs. Há varias incompatibilidades entre versões dos FTs e isso deve ser gerenciado de forma automática.

No subprojeto 6 (SP6 – Sincronização de Modelos) o objetivo é a criação de um mecanismo para gerenciar a sincronização dos modelos de projeto com o código-fonte dos FTs. Neste projeto pretende-se utilizar e adaptar um perfil UML para projeto de FTs já desenvolvido em um trabalho anterior (Uetanabara *et al.*, 2010; Uetanabara *et al.*, 2009a; Uetanabara *et al.*, 2009b).

No subprojeto 7 (SP7) o objetivo é evoluir e aprimorar uma ferramenta que está em desenvolvimento (Parreira *et al.*, 2010a, 2010b) e que dá suporte à recuperação de modelos de sistemas legados. A ideia principal é que os modelos recuperados possam servir de principal artefato para manutenções e evoluções que possam ser feitas no sistema e que

técnicas de transformação de modelos e refatorações possam ser aplicadas para a geração do novo código.

O SP8 tem como objetivo aprimorar técnicas de avaliação de perfis UML e implementá-las dentro da infraestrutura de forma que os modelos manipulados pelo engenheiro de software tenham garantia de correção. Dessa forma, os modelos criados e modificados poderão ser checados quando à sua consistência em relação a um metamodelo previamente definido.

No SP9, o objetivo é a implementação da arquitetura (parte técnica) que dá apoio a toda infraestrutura. Serão investigados prós e contras do desenvolvimento de um novo ambiente completamente novo comparado com o desenvolvimento de plug-ins para ambientes já existentes, como por exemplo o Eclipse.

No SP10 o objetivo é estudar técnicas e ferramentas para o adequado gerenciamento de FTs com modelos. Serão analisadas várias propostas de notação para POA existentes na literatura, desde perfis UML a notações específicas de domínio.

Na Figura 2 é mostrado um diagrama de componentes que representa os módulos que integram a infraestrutura proposta. Os módulos brancos são denominados de “módulos de apoio” e o módulo cinza é o “módulo central”. A ideia é projetar esses módulos com interfaces providas e requeridas. Interfaces providas são operações com assinaturas claras que determinado componente expõe para que outros módulos possam usar suas funcionalidades. Interfaces requeridas são formatos pré-estabelecidos de interfaces que determinado componente exige para que ele possa se utilizar de serviços disponibilizados por outros. Não há uma correspondência direta entre cada subprojeto mostrado na Tabela 1 e cada componente. Alguns projetos podem contribuir com o desenvolvimento de um ou mais componentes e vice versa.

O módulo central deve ser projetado apenas com interfaces requeridas enquanto que os outros módulos de apoio apenas com interfaces providas. Dessa forma, o módulo central torna-se apenas consumidor das funcionalidades disponibilizadas pelo outros módulos e independente da implementação desses módulos. Além disso, essa estratégia permite que esses módulos de apoio possam ser substituídos por implementações feitas por outros grupos de pesquisa, ou mesmo mantidos<sup>1</sup>, sem que alterações sejam propagadas para o módulo

---

<sup>1</sup> “Mantido” aqui está sendo usado no sentido de “efetuar tarefas de manutenção”

central. Outro ponto importante é que o adequado projeto dessas interfaces permite que os módulos sejam projetados e construídos independentemente uns dos outros para que ao final do projeto a integração possa ser feita. Os retângulos tracejados ao redor das interfaces representam adaptadores que devem ser construídos para adaptar a interface de módulos de outros grupos ao formato exigido pelo módulo central.

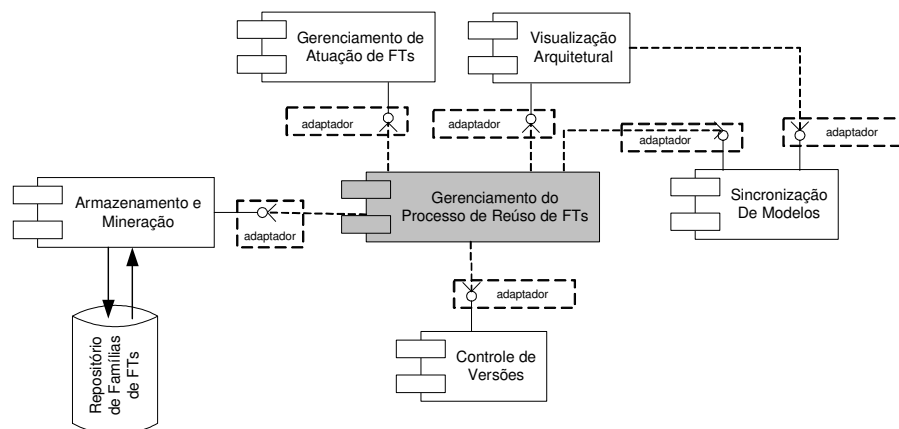


Figura 2 – Infraestrutura de Apoio ao Processo de Reuso e Gerenciamento de Famílias de FTs

## 4 – PRINCIPAIS CONTRIBUIÇÕES CIENTÍFICAS OU TECNOLÓGICAS

As principais contribuições científicas esperadas deste projeto são:

1. **Um ferramental integrado, baseado em modelos, de apoio ao reuso e gerenciamento de famílias de FTs:** Possivelmente poderá ser desenvolvido como vários *plugins* do ambiente Eclipse. Para a interoperabilidade das ferramentas deverá ser desenvolvido um metamodelo em XMI.
2. **Um metamodelo para representação de famílias de FTs.** Para que seja possível realizar o processo de mineração pelas funcionalidades das famílias de FTs, será necessário a criação de um metamodelo.
3. **Teoria/notação e ferramental para gerenciamento de conflito entre FTs.** Deverá ser desenvolvida uma técnica e uma notação para controlar conflitos entre FTs. Também será desenvolvida uma ferramenta que permita gerenciar esses conflitos.
4. **Aperfeiçoamento da Ferramenta de Controle de Versões.** A ferramenta de controle de versões de FTs desenvolvida até o momento (Arimoto *et al.*, 2008) não

opera para famílias de FTs, apenas para FTs individuais. Assim, será necessário evoluir essa ferramenta.

5. **Aperfeiçoamento do perfil UML-AOF.** O perfil UML-AOF (Uetanabara *et al.*, 2010) foi desenvolvido para representar algumas características básicas de FTs como idiomas (Hanenberg e Schimidmier, 2003), padrões (Camargo e Masiero, 2008a) e pontos de extensão (Fontoura *et al.*, 2002). Entretanto, a representação de famílias de FTs possui outras características que também precisam ser contempladas por esse perfil, como por exemplo, características alternativas, opcionais e obrigatórias, características mutuamente exclusivas e relacionamentos com outras famílias.
6. **Empacotamento de Experimentos.** Diversos experimentos deverão ser conduzidos durante o projeto. Pretende-se empacotar esses experimentos para que eles possam ser replicados por outros grupos de pesquisadores (Amaral e Travassos, 2002).
7. **Técnicas de Mineração de Funcionalidades em Famílias de FTs.**
8. **Artigos Científicos.** Publicação de artigos em conferências e *journals* qualificados da área.

## 5 – AVALIAÇÃO

---

Como o projeto proposto é constituído por subprojetos, pretende-se avaliar cada um dos subprojetos à medida que eles são finalizados. Essas avaliações intermediárias serão conduzidas de acordo com a natureza do subprojeto. Por exemplo, o subprojeto 2 (SP2) será avaliado no sentido de verificar se as funcionalidades desejadas são realizadas adequadamente e se as interfaces são intuitivas de forma que realizem o gerenciamento pretendido. O mesmo poderá ser feito para o subprojeto 3 (SP3). Já para o subprojeto 4 (Visualização Arquitetural) o processo de avaliação constitui da comparação da notação usada com alguma já proposta na literatura, como por exemplo a de Krechetov e outros (2006) ou a de Batista e outros (2006). Nesse tipo de avaliação é interessante usar as estratégias de experimentação proposta por Wohlin e outros (2000), que já foram usadas em experimentos anteriores (Camargo *et al.*, 2006, Uetanabara *et al.*, 2010).

Ao final deste projeto, pretende-se conduzir experimentos controlados (Basili, 1993, Wohlin *et al.*, 2000) para averiguar a produtividade de engenheiros de software que utilizam a infraestrutura proposta. Também pretende-se averiguar com experimentos controlados e

métricas, a qualidade do software (Sant'Anna, 2004) desenvolvido com o apoio da infraestrutura em termos de manutenibilidade, reusabilidade e evolutibilidade.

O projeto será disseminado por meio de um portal Web e toda contribuição tecnológica desenvolvida será disponibilizada com código aberto em portais como o SourceForge<sup>2</sup> sob licença GPL (*General Public License*).

## 6 – ORÇAMENTO DETALHADO

Para viabilizar o desenvolvimento deste projeto, diversos recursos já disponíveis nas universidades dos participantes serão utilizados. Em termos de recursos humanos, serão engajados neste projeto, além dos professores participantes, alunos de mestrado e iniciação científica – alguns já se encontram engajados em alguns subprojetos. Em termos de infraestrutura, as duas universidades contam com toda infraestrutura necessária para a condução do projeto.

Quanto aos recursos financeiros, no entanto, serão necessários os relacionados à compra de computadores; diárias para possibilitar reuniões do projeto, já que os participantes são provenientes de diferentes localidades; diárias para apresentação de trabalhos em congressos, bem como ajuda de custo para passagens aérea e terrestre e despesas com inscrições em congressos. Na Tabela abaixo apresenta-se uma estimativa total de gastos deste projeto e, em seguida, uma descrição detalhada de cada tipo de despesa.

**Tabela 2 – Estimativa Total de Despesas**

Descrição	Valor
<b>CUSTEIO</b>	
Material de consumo, componentes, peças de reposição	R\$ 500,00
Diárias mensais para reuniões do projeto	R\$ 3.756,60
Passagens para apresentação de trabalhos em congressos e reuniões nas duas Universidades (UFSCar e Unifast)	R\$ 6.000,00
<b>CAPITAL</b>	
Equipamentos (computadores que serão distribuídos entre as duas universidades participantes do projeto)	R\$ 26.744,00
<b>BOLSAS</b>	
5 BolsaS DTI-3	<b>62.753,40</b>
<b>TOTAL GERAL</b>	<b>R\$ 99253,4</b>

<sup>2</sup> <http://sourceforge.net/>

## 6.1 - CUSTEIO

### 6.1.1 – Material de Consumo, componentes e peças de reposição.

Está sendo solicitado R\$ 500,00 para materiais de consumo, componentes e peças de reposição, como por exemplo, *toners* de impressoras.

### 6.1.2 – Passagens e diárias para apresentação de trabalhos em congressos

Como o projeto envolve uma universidade do nordeste, haverá reuniões nas duas universidades envolvidas (UFSCar e Unifasf). Estima-se que ocorram quatro reuniões durante o período de vigência do projeto. Assim, solicita-se 20 diárias (4 reuniões/workshops de 5 dias) e R\$ 5.500,00 em passagens aéreas para deslocamento entre São Carlos e Petrolina e para participação em congressos qualificados da área.

## 6.2 - CAPITAL

### 6.2.1. Equipamentos

Seis (6) Microcomputadores (4 desktops e dois notebooks) e duas impressoras. 2 desktops, um notebook e uma impressora serão alocados na Universidade Federal de Petrolina e 2 desktops, um notebook e uma impressora para a Universidade Federal de São Carlos. A configuração e o preço dos computadores desktop abaixo foram obtidos no site da Dell: (<http://www1.la.dell.com/content/products/productdetails.aspx/alienware-aurora?c=br&l=pt&s=dhs&cs=brdhs1>). A configuração e o preço dos notebooks e impressoras abaixo foram obtidos no site das lojas americanas: (<http://www.americanas.com.br/AcomProd/590/3149078>)

O detalhamento da configuração dos equipamentos encontra-se na tabela abaixo.

Configuração	Preço Unitário	Valor Item
Processador Intel® Core™ i5-750 (2.66 GHz, 8 MB L3 cache) - BRH8801 Windows® 7 Home Premium Original 64-bit em Português Memória 6GB DDR3 1066MHz (2x2GB + 2x1GB) Disco Rígido SATA de 500GB (7200RPM) 3.0Gb/s c/ 16MB cachê Monitor Dell ST2010 HD de 20 Polegadas Widescreen Gravador de DVD/CD Dual Layer (Unidade DVD+/- RW 16x) Placa de Vídeo Nvidia GeForce GTS240 (1024MB GDDR3) Alto-Falantes Externos Dell AX210 2.0 1.2W Microsoft Office Starter 2010 McAfee SecurityCenter - 3 anos Fax Modem 56K PCI	R\$ 3.812,00	4 unidades = R\$ 15.248,00



<b>Notebook</b>	<b>Preço Unitário</b>	<b>Valor Item</b>
Netbook Sony VAIO P110JB/D c/ Intel® Atom Z530 1.60GHz 2GB 128GB SSD Webcam 8" Windows 7 Basic – Sony Processador: Processador Intel® Atom Z530 (1.60GHz) Memória: 2GB DDR2 HD: 128GB SSD Tela: 8"(1600 x 768) Widescreen com tecnologia LED Tela: 8"(1600 x 768) Widescreen com tecnologia LED Áudio: Intel(R) High Definition Audio Vídeo: Intel(R) Graphics Media Accelerator 500 Webcam: 640x480 – VGA Rede: 10Base-T/100Base-TX/1000Base-T Wireless: Foxconn T77H126.00 - IEEE 802.11/802.11b/802.11g/Draft 802.11n (20/40) Bluetooth: Standard Ver. 2.1+EDR Sistema Operacional: Windows® 7 Home Basic original Conexões: USB (2 portas), VGA (1 saída) e DC In Teclado: Qwerty em Português Mouse: Touch pad eletrostático	R\$ 3.999,00	2 unidades = R\$ 7.998,00
<b>IMPRESSORAS</b>	<b>Preço Unitário</b>	<b>Valor Item</b>
MULTIFUNCIONAL HP LASERJET MONO Modelo M1120	R\$ 749,00	R\$ 1.498,00
<b>Valor Total de Equipamentos</b>		R\$ 24.744,00

### 6.2.2 – Material Permanente de Escritorio

É previsto um gasto de R\$ 2.000,00 com mobiliário para adequada acomodação dos computadores.

### 6.3 - Bolsas

Na Tabela abaixo encontram-se as bolsas que estão sendo solicitadas. São cinco bolsas DTI-3; sendo três para a UFSCar e duas para a Unifasf. Essas bolsas serão atribuídas a alunos com boa formação acadêmica pois o objetivo é que esses alunos auxiliem no desenvolvimento da parte técnica do ambiente proposto. Além dessas bolsas, o proponente pretende incluir outros alunos de iniciação científica e de mestrado para a adequada condução do projeto.

<b>Tipo da Bolsa</b>	<b>Nro bolsas</b>	<b>Duração</b>	<b>Valor</b>
DTI-3	5	12 meses	62.753,40

## 7 – CRONOGRAMA FÍSICO FINANCEIRO

Como mostrado na Seção 4, há 10 subprojetos que compõem o projeto de infraestrutura proposta. O cronograma mostrado na Tabela 3 distribui os subprojetos de acordo com o tempo disponível de 36 meses.

Cada **xT** representa um determinado trimestre do ano. As letras IC, M e DT3 representam o tipo da bolsa que será atribuída para condução de cada projeto. Note-se que alguns projetos envolvem duas bolsas de DT3 de 12 meses ao invés de uma de 24 meses. Isso foi feito para assegurar um melhor controle do aluno que estiver com essa bolsa. Se o desempenho desse estiver adequado, a segunda bolsa é atribuída ao mesmo aluno, caso contrário troca-se o aluno e também a bolsa. Além disso, bolsas de 12 meses permitem uma flexibilidade maior caso alguma coisa precisa ser alterada no cronograma.

### Tabela 3 – Cronograma Físico-Financeiro

[illegible]

Na parte inferior da tabela encontra-se a previsão de compra dos computadores, impressoras e mobiliário e também as reuniões que serão realizadas nas duas universidades. Na parte das reuniões há indicação de quantos dias cada uma levará. Note-se que a soma do número de dias corresponde ao número de diárias que foi solicitado (20 diárias).

## **8 – IDENTIFICAÇÃO DOS DEMAIS PESQUISADORES**

---

A **Profa. Dra. Rosângela Penteado** possui graduação em Licenciatura Em Matemática pela Universidade Estadual Paulista Júlio de Mesquita Filho (1976), mestrado em Ciências da Computação pela Universidade de São Paulo (1984) e doutorado em Física Computacional pela Universidade de São Paulo (1996). Atualmente é professora associada da Universidade Federal de São Carlos, Revisora - INFOCOMP Journal of Computer Science. Tem experiência na área de Ciência da Computação, com ênfase em Engenharia de Software, atuando principalmente em pesquisas que envolvem: reengenharia de software, padrões de projeto, orientação a objetos, orientação a aspectos, geradores de aplicação, manutenção e qualidade de software.

**Lattes:** <http://lattes.cnpq.br/4061946080136286>

O **Prof. Dr. Ricardo Argenton Ramos** possui graduação em Processamento de Dados pela Faculdade de Tecnologia (1999), especialização em Ciência da Computação pela Universidade Federal de São Carlos (2002), mestrado em Ciência da Computação pela Universidade Federal de São Carlos (2004) e doutorado em Ciência da Computação pela Universidade Federal de Pernambuco (2009). Atualmente é professor Adjunto I da Universidade Federal do Vale do São Francisco em Juazeiro (UNIFASF). Tem experiência na área de Ciência da Computação, com ênfase em Engenharia de Software, atuando principalmente nos seguintes temas: programação orientada a aspectos, reengenharia e engenharia reversa de softwares, documento de requisitos, casos de uso, qualidade e métricas.

**Lattes:** <http://lattes.cnpq.br/6190953685221120>

## 9 - REFERÊNCIAS BIBLIOGRÁFICAS

---

- Alves, V., Matos J. P., Osandy A., Cole N. L., Vasconcelos, A., Borba, P., Ramalho, G. Extracting And Evolving Code In Product Lines With Aspect-Oriented Programming. *Transactions On Aspect-Oriented Software Development*, V. Iv, P. 117-142, 2007.
- Amaral, E.A.G., Travassos, G.H. Em Busca De Uma Abordagem Para Empacotamento De Experimentos Em Engenharia De Software. In: *Anais Da 2ª JIISIC – Jornada Ibero-Americana de Engenharia de Software e Engenharia de Conhecimento*, 2002, Salvador, 2002.
- Apel, S., Leich, T., Saake, G. Aspectual Mixin Layers: Aspects And Features In Concert. In: *Proceedings Of ICSE (International Conference On Software Engineering)*, 2006.
- Araújo, J., Moreira, A. M. D. An Aspectual Use-Case Driven Approach. In: *Jisbd*, Pp. 463–468, 2003.
- Araújo, J., Moreira, A., Brito, I., Rashid, A. Aspect-Oriented Requirements With Uml. In: *Proceedings Of The Workshop On Aspect-Oriented Modelling With Uml* (held in conjunction with the International Conference on Unified Modelling Language UML 2002), 2002.
- Arimoto M.M., Cagnin, M.I., Camargo, V.V. Version Control In Crosscutting Framework Based Development. In *ACM Annual Symposium On Applied Computing (ACM-SAC)*, Fortaleza, Brasil, 2008.
- Arimoto, M.M., Camargo, V.V., Cagnin, M.I. Uma Ferramenta De Controle De Versões De Frameworks. In: *Congresso Latino Americano De Informática (CleI'07)*, 2007.
- Baniassad, E., Clements, P.C., Moreira, A., Araújo, J., Rashid, A., Tekinerdogan, B. *Discovering Early Aspects*. *IEEE Software*, Pp. 61-70, 2006.
- Basili, V. The Empirical Paradigm In Software Engineering, Experimental Software Engineering Issue: Critical Assessment And Future Directions. *Spring Verlag*, #706, *Lecture Notes In Computer Science*, University Of Maryland, 1993.
- Batista, T., Chavez, C., Garcia, A., Sant'Anna, C. N., Kulesza, U., Lucena, C. J. P. (2006) **Aspectual Connectors**: Supporting the Seamless Integration of Aspects and ADLs. In: *20º Simpósio Brasileiro de Engenharia de Software (SBES)*, 2006, Florianópolis. *Anais do XX Simpósio Brasileiro de Engenharia de Software (SBES)*, p. 17-32.
- Camargo, V. V.; Ramos, R. A.; Penteado, R. A. D.; Masiero, P. C. (2003). Projeto Baseado Em Aspectos Do Padrão Camada De Persistência. In: *XVII Simpósio Brasileiro De Engenharia De Software (SBES'2003)*, Manaus-AM. P. 114-129.
- Camargo, V.V. Frameworks Transversais: Definições, Classificações e Utilização em um Processo de Desenvolvimento. Tese, Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, 2006.
- Camargo, V.V., Masiero, P.C. (2005) “Frameworks Orientados A Aspectos”. In: *Anais Do 19º Simpósio Brasileiro De Engenharia De Software (SBES'2005)*, Uberlândia-MG, Brasil, Outubro.
- Camargo, V.V., Masiero, P.C. A Pattern To Design Crosscutting Framework Families. In *ACM Annual Symposium On Applied Computing (ACM-SAC)*, Fortaleza, Brasil, 2008a.
- Camargo, V.V., Masiero, P.C. Proft/Pu – Um Processo De Desenvolvimento De Software Orientado A Aspectos Apoiado Por Frameworks Transversais. Tutorial Do *XXII Simpósio Brasileiro De Engenharia De Software (2008c)*, Campinas, SP, Brasil.

- Camargo, V.V., Masiero, P.C., An Approach To Design Crosscutting Framework Families. In: Aspects, Componentes And Patterns For Infrastructure Software (ACP4IS) Workshop, Bruxelas, Bélgica, 2008b.
- Camargo, V.V., Ramos, R.A., Pentead, R.A.D., Pastor, O.L. Extensão do Método Orientado a Objetos com Frameworks Transversais. Projeto Universal CNPq, edital MCT/CNPq 14/2009.
- Camargo, V.V.; Masiero, P.C. Projeto de Frameworks Transversais Utilizando A Abordagem Tema E O Perfil UML-F. Revista De Informática Teórica E Aplicada, V. 14, P. 85-102, 2007.
- Carvalho, A.P.L.F et al. Grandes Desafios da Pesquisa em Computação no Brasil - 2006 – 2010. Relatório sobre o Seminário Realizado em 8 e 9 de maio de 2006. Disponível em: [www.sbc.org.br](http://www.sbc.org.br). Consultado em 29/07/2010.
- Chavez, C. V. F. ; Garcia, A. ; Batista, T. V. ; Oliveira, M. ; Santanna, C. ; Rashid, A. . Composing Architectural Aspects Based On Style Semantics. In: 8th International Conference On Aspect-Oriented Software Development (AOSD), 2009, Charlottesville. Proceedings Of The 8th International Conference On Aspect-Oriented Software Development (AOSD). New York : ACM, 2009. P. 111-122.
- Clarke, S; Baniassad, E. Aspect Oriented Analysis And Design – The Theme Approach. Editora Addison-Wesley, Primeira Edição, 2005.
- Clements, P., Northrop, L. (2002) Software Product Lines: Practices And Patterns, Addison Wesley, Boston.
- Constantinides C. A.; Elrad T. Composing Concerns with a Framework Approach. In: *International Workshop on Distributed Dynamic Multiservice Architectures in conjunction with the 21st International Conference on Distributed Computing Systems (ICDCS)*. Phoenix, Arizona, USA, Abril, 2001.
- Costa, H.X; Parreira, P.A.; Camargo, V.V.; Pentead, R.A.D. Recovering Class Models Stereotyped with Crosscutting Concerns. In: Session Tool of WCRE (Working Conference on Reverse Engineering), 2009, Lille, França. WCRE - Working Conference on Reverse Engineering, 2009b.
- Costa, H.X; Parreira, P.A.; Camargo, V.V.; Pentead, R.A.D. Recuperação de Modelos de Classes Anotados com Índícios de Interesses Transversais. In: III LA-WASP (Latin American Workshop on Aspect-Oriented Software Development), 2009, Fortaleza. SBES - Simpósio Brasileiro de Engenharia de Software, 2009a
- De Grandi, M., Camargo, V.V., Spotto, E.S. Uma Abordagem Para Tratamento De Regras De Negócio Nas Fases Iniciais Do Desenvolvimento. In: XI Workshop Iberoamericano De Ambientes De Software E Engenharia De Requisitos (IDEAS), Recife, Pernambuco, 2008.
- Eclipse. w[www.Eclipse.Org](http://www.Eclipse.Org). Último Acesso Em 29/07/2010
- Evermann, Joerge. 2007. A Meta-Level Specification And Profile For Aspectj In UML. Victoria University Wellington, Wellington, New Zealand. AOSD 2007.
- Fayad, M. E.; Johnson, R. E. (Eds) (2000). *Domain-Specific Application Frameworks: Frameworks Experience By Industry*, John Wiley & Sons.
- France, B., Rumpe B. Model-driven Development of complex systems: A research roadmap. In 29<sup>th</sup> International Conference on Software Engineering 2007 – Future of Software Engineering, pages 37-54, Minneapolis, MN, USA, 2007. IEEE Computer Society.
- Fuentes, F., Nebrera, C., Sánchez, P. Feature-Oriented Model Driven Software Product Lines: The TENTE Approach. In Proceedings of Caise 2009.
- Fujii, C.S. ; CHIARAMONTE, R. B. ; CAMARGO, Valter Vieira De . Arquitetura Genérica Para Projeto E Implementação De Criptografia Com Programação Orientada A Aspectos.

- In: CLEI - Conferência Latino Americana De Informática, 2008, Santa Fe - Argentina. Conferência Latino Americana De Informática (CLEI), 2008. V. 1. P. 519-528.
- Gomaa, Hassan. (2004) Designing Software Product Lines With UML – From Use Case To Pattern-Based Software Architectures. Addison Wesley, 1a. Edição.
- Gottardi, T. ; Ramos, R. A.; Penteadó, R.; Pastor, O. L. ; Camargo, V.V. Diretrizes para Avaliação e Aperfeiçoamento de Perfis UML para Sistemas Orientados a Aspectos. In: XXXVI Conferência Latino-Americana de Informática, 2010, Assunção. XXXVI Conferência Latino-Americana de Informática, 2010.
- Hanenberg, S., Schmidmeier, A. Aspectj Idioms For Aspect-Oriented Software Construction. In *Proceedings Of 8th European Conference On Pattern Languages Of Programs (Eurolop)*, Irsee, Germany, 25th–29th June, 2003.
- Huang, M., Wang, C., Zhang, L. Towards a Reusable and Generic Aspect Library. In: *Proceedings of AOSDSEC'04 (AOSD Technology for Application-Level Security). Workshop of the Aspect Oriented Software Development Conference*, Lancaster, UK, March, 23, 2004.
- Jacobson, I., Ng, P. Aspect-Oriented Software Development With Use Cases. Addison-Wesley, 2004.
- Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., Griswold, G. Getting Started With Aspectj, Communications Of The Acm, Vol 44, No. 10, Pp.59-65, 2001.
- Kiczales, G.; Lamping, J.; Mendhekar, A.; Maeda, C.; Lopes, C.; Loingtier, J.; Irving, J. Aspect Oriented Programming. In: *Proceedings Of ECOOP*. Pp. 220-242, 1997.
- Krechetov, I., Tekinerdogan, B., Garcia, A., Chavez, C., Kulesza, U. Towards An Integrated Aspect-Oriented Modeling Approach For Software Architecture Design. In: *Aspect-Oriented Modelling Workshop (AOM'06), In Conjunction With Aspect-Oriented Software Development Conference (AOSD'06)*, Bonn, Alemanha, 2006.
- Kulesza, U., Alves, V., Garcia, A., Costa Neto, A., Cirilo, E., Lucena, C., Borba, P. Mapping Features To Aspects: A Model-Based Generative Approach. In: 10th Internacional Workshop On Early Aspects, 6th ACM International Conference On Aspect-Oriented Software Development (AOSD 2007), 2007, Vancouver. Early Aspects 2007 Workshop, Lecture Notes In Computer Science. Berlin Heidelberg : Springer-Verlag, 2007. V. LNCS. P. 155-174.
- Larman, G. Utilizando Uml E Padrões – Uma Introdução A Análise E Ao Projeto Orientados A Objetos E Ao Processo Unificado. 2ª. Edição, Bookman, 2004.
- Lazanha, R.; Oliveira, A.; Penteadó, R.A.D.; Ramos, R.A.; Pastor, O.L.; Camargo, V.V. Uma Arquitetura de Referência Baseada em Papéis Para Frameworks Transversais de Persistência: Uma Análise Quantitativa. In: XXXVI Clei – Conferência Latino-Americana de Informática, 2010, Assunção, Paraguay. (Aceito para publicação)
- Mezini, M., Ostermann, K. Variability Management With Feature Oriented Programming And Aspects. In: *SIGSOFT/FSE*, ACM, 2004.
- Mortensen, M., Ghosh, S. Creating Pluggable and Reusable Non-functional Aspects in AspectC++. In: *Proceedings of the 5<sup>th</sup> Workshop on Aspects, Components and Patterns for Infrastructure Software (ACP4IS'06). Workshop of the Aspect-Oriented Software Development Conference*, Bonn, Alemanha, 2006a.
- Mortensen, M., Ghosh, S. Using Aspects With Object-Oriented Frameworks. In: *Proceedings of the Aspect-Oriented Software Development Conference – industry track*, Bonn, Alemanha, 2006b.
- Netbeans. <http://www.netbeans.org/>. Último Acesso Em 29/07/2010.

- Oliveira, A.L.; Pentead, R.; Camargo, V.V. Manutenção de Frameworks Orientados a Objetos com Orientação a Aspectos. In: WMSWM, 2010, Belém. Workshop de Manutenção de Software Moderna, 2010.
- Parreira, P.A.; Costa, H.X.; Camargo, V.V.; Pentead, R.A.D. Uma Abordagem Iterativa para Identificação de Interesses Transversais com o Apoio de Modelos de Classes Anotados. In: IV Latin American Workshop on Aspect-Oriented Software Development, 2010, Salvador. IV Latin American Workshop on Aspect-Oriented Software Development, 2010
- Parreira, P.A.; Costa, H.X.; Camargo, V.V.; Pentead, R.A.D. Uma Abordagem Iterativa para Identificação de Interesses Transversais com o Apoio de Modelos de Classes Anotados. In: IV Latin American Workshop on Aspect-Oriented Software Development, 2010, Salvador. IV Latin American Workshop on Aspect-Oriented Software Development, 2010
- Pastor, O., Molina, J.C. Model-Driven Architecture in Practice – A Software Production Environment Based on Conceptual Modeling. Springer Verlag, 2007.
- Ramos, R.A., Camargo, V.V., Pentead, R.A.D. Reuso Da Implementação Orientada A Aspectos Do Padrão De Projeto Camada De Persistência. In: 4<sup>th</sup> Latin American Conference On Pattern Languages Of Programming (Sugarloaf Plop), Fortaleza, 2004.
- Rausch, A., Rumpe, B., Hoogendoorn, L. Aspect-Oriented Framework Modeling. In: The 4th AOSD Modeling With UML Workshop. 2004.
- Sant'anna, C. N. : Manutenibilidade e Reusabilidade de Software Orientado a Aspectos: Um Framework de Avaliação. Dissertação de Mestrado, Departamento de informática, PUC-Rio Rio de Janeiro, 200Schmidt, D.C. Guest editor's instruction: Model Driven Engineering. IEEE Computer, 39 (2): 25-31, 2006.
- Shah, V., Hill, V. An Aspect-Oriented Security Framework: Lessons Learned. In: Proceedings of AOSDSEC'04 (AOSD Technology for Application-Level Security). Workshop of the Aspect Oriented Software Development Conference, Lancaster, UK, March, 23, 2004.
- Soares, S., Laureano, E., BORBA, P. Distribution and Persistence as Aspects. In: Software: Practice and Experience, volume 33, issue 7, pp 711-759, 2006.
- Stein, D.; Hanenberg, Stefan And Unland, Rainer. 2002. An UML-Based Aspect-Oriented Design Notation For Aspectj. University Of Essen, Germany. AOSD 2002.
- Uetanabara, J. J., Camargo, V. V.; Flach, C. V. UML-AOF: A Profile For Modeling Aspect-Oriented Frameworks. In: 13th ACM Workshop On Aspect-Oriented Modeling (AOM) In Conjunction With 8th International Conference On Aspect-Oriented Software Development, 2009a, Charlottesville.
- Uetanabara, J.J.;Parreira, P. A. ; Lanza, R.; Camargo, V.V.; Pentead, R.A.D. Preliminary Comparative Study Using UML-AOF - a UML Profile for Aspect-Oriented Frameworks. In: 14th IEEE/ACM Workshop on Aspect-Oriented Modeling (AOM), 2009, Denver. IEEE/ACM International Conference on Model Driven Engineering, Languages and Systems (MODELS 2009), 2009b.
- Uetanabara, J.J.; Pentead, R.A.; Camargo, V.V. An Overview and An Empirical Evaluation of UML-AOF – A UML Profile for Aspect-Oriented Frameworks. In: ACM SAC, Sierre, Suíça, 2010.
- Vanhaute, B., de WIN, B., Decker, B. Building Frameworks in AspectJ. In: *Proc. of the 15<sup>th</sup> European Conference on Object-Oriented Programming (ECOOP)*, Separation of Concerns Workshop. pp. 1-6, June, 2001.
- Zanon, I. ; Camargo, Valter Vieira De ; Pentead, Rosângela . Reestruturação De Um Framework De Aplicação Com Um Framework Transversal De Persistência. In:

Workshop De Manutenção De Software Moderna (WMSWM), 2009, Ouro Preto.  
Simpósio Brasileiro De Qualidade De Software, 2009. *Artigo Aceito Premiado e  
Convidado Para Publicação Na Revista INFOCOMP (Submetido).*