

Programa Nacional de Cooperação Acadêmica
PROCAD / CAPES

**Desenvolvimento de Linhas de Produtos de Software
usando Técnicas Orientadas a Aspectos**

Coordenador: Prof. Paulo Henrique Monteiro Borba
Universidade Federal de Pernambuco (UFPE)
Centro de Informática

Edital Procad CAPES 01/2007

Sumário

1. Introdução e Justificativa	3
2. Objetivos e Metas	4
3. Metodologia e Plano de Atividades	4
4. Cronograma de Atividades	7
5. Metas e Resultados Esperados	7
6. Fundamentação Teórica e Linhas de Pesquisa.....	8
6.1 Análise de Domínio	9
6.2 Projeto de Domínio	10
6.3 Implementação de Domínio.....	11
6.4 Ferramentas de Linha de Produto de Software	11
7. Equipes Participantes do Projeto	12
Referências.....	14

1. Introdução e Justificativa

Ao longo dos últimos anos, diversas abordagens para o desenvolvimento de linhas de produto de software (LPSs) foram propostas [23, 26, 35, 66]. Uma LPS pode ser vista como uma família de sistemas [53] que focaliza um segmento de mercado específico, compartilhando funcionalidades comuns e definindo funcionalidades específicas que variam de acordo com o produto sendo considerado. Nas abordagens já propostas, cada LPS é especificada, modelada e implementada em termos de *features* comuns e variáveis. Um *feature* [25] é uma funcionalidade ou propriedade relevante da LPS, que pode ser comum a toda a linha de produto ou estar presente em apenas um ou mais de seus produtos.

O desenvolvimento de uma linha de produto [23, 26, 35] envolve tipicamente a execução de atividades de: (i) *engenharia de domínio* – objetiva o desenvolvimento de uma arquitetura comum para a linha de produtos, assim como um conjunto de artefatos reusáveis (*core assets*) que possam ser usados na implementação de tal arquitetura; e (ii) *engenharia de aplicação* – define como produtos podem ser implementados, através do reuso dos artefatos gerados durante a engenharia de domínio. Algumas abordagens de linha de produto [23] ressaltam também a importância de estabelecer um conjunto de atividades de gerenciamento organizacional e técnico responsáveis pela definição de estratégias de sucesso de implantação da abordagem de LPSs.

Experiências industriais têm comprovado a eficiência das abordagens de LPS no aumento da produtividade e qualidade de software produzido por grandes empresas [2, 65]. Apesar do sucesso obtido e das vantagens trazidas pelo uso de abordagens de LPSs, diversos pesquisadores têm relatado problemas e dificuldades [6, 8, 10, 48, 50] que as técnicas atuais de especificação, projeto e implementação podem trazer para o gerenciamento de variabilidades. Uma variabilidade é uma característica da LPS que varia de acordo com o membro da família sendo considerado [35]. Novas técnicas de modelagem e programação estão sendo propostas com o objetivo de melhorar a modularização de *features* em LPSs. Boa parte dessas técnicas é centrada nos princípios e mecanismos de composição do paradigma de Desenvolvimento de Software Orientado a Aspectos (DSOA) [33, 40]. Tais técnicas são, em geral, complementares a abordagens de LPSs já propostas.

Diversos benefícios têm sido relatados por pesquisadores a partir da experiência do uso de técnicas de DSOA no desenvolvimentos de LPSs [6-8, 10, 19, 42, 45, 48, 50, 68], tais como: (i) separação clara de *features* transversais e não transversais já desde estágios preliminares de desenvolvimento; (ii) mapeamento direto de *features* transversais em aspectos; (iii) aumento da capacidade de reuso de artefatos de software (*core assets*) relacionados a *features* transversais; e (iv) facilidade para configuração e customização de produtos específicos da LPS. Apesar da identificação desses benefícios, diversos desafios ainda permanecem no que se refere ao uso sistemático e integrado de técnicas de DSOA no desenvolvimento de LPS. Vários esforços individuais de grupos de pesquisa têm sido relatados, mas apenas um pequeno conjunto de estudos de caso tem sido realizado. Boa parte dos trabalhos propostos têm focalizado, sobretudo, atividades de projeto e implementação da LPS. Contudo, evidências empíricas provenientes de estudos qualitativos e quantitativos que demonstrem os benefícios do uso de técnicas de DSOA no desenvolvimento de LPS precisam também ser investigadas.

Nesse contexto, o objetivo desse projeto de cooperação acadêmica é promover o intercâmbio entre pesquisadores de quatro grupos de pesquisa de universidades brasileiras – Universidade Federal de Pernambuco (UFPE), Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Universidade de São Paulo – São Carlos (USP-SC) e Universidade Federal do Rio Grande do Norte (UFRN) – que já vêm atuando nas áreas de LPSs e DSOA. O projeto será centrado no aprimoramento e integração das técnicas e ferramentas de orientação a aspectos (OA) já propostas por tais grupos de pesquisas, e que contemplam as diferentes atividades da engenharia de domínio e aplicação (análise de domínio, arquitetura e projeto de domínio, implementação de

domínio, e derivação de produtos) de desenvolvimento de uma LPS. Dois estudos de caso serão realizados como parte da validação de integração e refinamento das abordagens. Será também investigado o impacto da combinação de técnicas de LPS e Desenvolvimento Dirigido a Modelos (DDM) [63] para permitir uma melhor integração entre as abordagens e ferramentas já propostas pelos grupos. Nós acreditamos que a combinação entre tais técnicas pode permitir um gerenciamento mais consistente de variabilidades ao longo do desenvolvimento de LPS, através da manutenção da rastreabilidade entre seus artefatos.

As seções seguintes deste documento estão organizadas da seguinte forma: na seção 2 são apresentados os objetivos gerais do projeto proposto. A seção 3 descreve a metodologia e o plano de atividades do projeto. As seções 4 e 5 descrevem, respectivamente, o cronograma e resultados esperados para o projeto. A seção 6 apresenta a fundamentação teórica e linhas de pesquisa que pretendemos explorar tanto durante quanto após o projeto. A seção 7 discorre sobre as qualificações da equipe do projeto. Finalmente, o Anexo I detalha o cronograma financeiro de missões de pesquisa e bolsas solicitadas para o projeto.

2. Objetivos e Metas

O objetivo central desse projeto de cooperação científica é promover o intercâmbio entre membros (professores, pesquisadores, doutorandos, mestrandos e alunos de iniciação científica) de grupos de pesquisa atuando na área de LPS e DSOA. Em particular, o projeto tem os seguintes objetivos:

- Permitir a disseminação das abordagens (técnicas, linguagens e ferramentas) OA para desenvolvimento de LPS já propostas pelos grupos de pesquisa participantes do projeto, através da sua aplicação em um conjunto de estudos de caso comuns;
- Promover discussões e avaliações críticas das abordagens já propostas, através da identificação de seus benefícios e limitações durante a sua respectiva aplicação nos estudos de caso. Isso permitirá identificar necessidades de melhoria e direcionar refinamentos específicos em cada uma das abordagens já propostas;
- Avaliar e promover a integração das abordagens propostas por cada grupo de acordo com a atividade de desenvolvimento ao qual se aplica (análise, projeto e implementação de domínio, derivação de produto). Como resultado dessa atividade, o projeto pretende produzir uma metodologia integrada das abordagens, que ofereça diretrizes claras de quando e como usar as técnicas durante o desenvolvimento de uma LPS;
- Aplicar as abordagens de acordo com as diretrizes oferecidas pela metodologia de integração proposta, de forma a avaliar a efetividade tanto dos refinamentos individuais propostos para cada uma das abordagens, quanto das diretrizes oferecidas pela metodologia de integração.

3. Metodologia e Plano de Atividades

A metodologia do projeto prevê uma série de atividades a serem realizadas ao longo dos seus quatro anos de duração. Tais atividades serão acompanhadas continuamente pelo coordenador do projeto, professor Paulo Borba (UFPE, e por mais 3 professores representando cada uma das outras três universidades participantes, Carlos Lucena (PUC-Rio), Paulo Masiero (USP-SC) e Paulo Pires (UFRN).

Cada atividade do projeto irá gerar um ou mais relatórios técnicos descrevendo os resultados obtidos até aquele estágio. Além dos relatórios técnicos serão produzidos diversos artigos científicos de qualidade a serem submetidos para conferências e periódicos nacionais e internacionais. Outro resultado importante do projeto serão as ferramentas produzidas pelo grupo

assim como os estudos de caso realizados no contexto do projeto. Além disso, os membros do projeto pretendem elaborar tutoriais e mini-cursos para disseminar tanto o estado-da-arte da área quanto os resultados parciais do projeto em eventos regionais, nacionais e internacionais. Todo esse material produzido durante o projeto será gradativamente disponibilizado em um portal. Finalmente, o projeto prevê a realização de *workshops* anuais com a participação de professores e pesquisadores das 4 instituições envolvidas no final de cada ano de atividades.

A seguir são descritas as fases do projeto, através da apresentação de um conjunto de macro-atividades e a respectiva metodologia para sua execução.

Macro-Atividade 1 (MA1) – Identificação de Diferenciais e Limitações de Abordagens Atuais

Essa macro-atividade prevê a aplicação das técnicas já desenvolvidas por cada um dos grupos na especificação, projeto e implementação de uma linha de produto existente. Na fase inicial, cada grupo de pesquisa deverá aplicar separadamente suas respectivas abordagens de acordo com a atividade da engenharia de domínio a que se aplicam. Os seguintes artefatos serão produzidos para serem usados como referência por todos os grupos de pesquisa: (i) um modelo de *features* – que expresse as *features* comuns e variáveis da LPS sendo desenvolvida; (ii) um conjunto de casos de uso – que expressem os requisitos a serem considerados para toda a LPS; e (iii) uma arquitetura de referência – que será usada como base para o projeto detalhado e implementação da LPS. Grupos de pesquisa que possuam abordagens para um mesmo estágio de desenvolvimento (análise, projeto e implementação de domínio, derivação de produto) irão interagir para compartilhar artefatos sendo gerados, ainda que com notações ou técnicas de documentação/especificação distintas. A linha de produto adotada nesse estágio será a *Arcade Game Maker*¹, desenvolvida pelo *Software Engineering Institute* (SEI) da *Carnegie Mellon University*, para fins pedagógicos.

Após a aplicação das abordagens dos grupos de pesquisa ao estudo de caso, serão identificados os pontos fortes trazidos por cada uma delas, assim como suas respectivas limitações. Essa análise será realizada inicialmente de forma individual para cada abordagem, e em seguida, será feita uma análise comparativa das abordagens pertencentes a um mesmo estágio. Finalmente, serão também analisadas outras abordagens propostas por grupos de pesquisa internacionais de destaque na área, e os resultados do estudo serão também confrontados com a análise de tais abordagens. No que se refere às ferramentas de LPS já propostas pelo grupo (Seção 6.4), durante essa macro-atividade os membros de cada grupo de pesquisa irão utilizar e avaliar as ferramentas já propostas com o objetivo de analisar a sua usabilidade e mecanismos e estratégias de integração dessas ferramentas.

Como resultado dessa atividade, pretende-se gerar vários relatórios técnicos cada um contemplando a análise de abordagens (técnicas, linguagens e ferramentas) para uma atividade específica do desenvolvimento. Cada relatório conterá: (i) os resultados de aplicação de cada uma das técnicas sobre o estudo de caso na atividade de desenvolvimento considerada; (ii) uma análise comparativa das técnicas utilizadas no que se refere às suas forças e limitações; (iii) discussões acerca dos benefícios trazidos pelas técnicas dos parceiros do projeto, em comparação com outras existentes; e (iv) direcionamentos para melhoria e integração das abordagens pertencentes a cada estágio.

Macro-Atividade 2 (MA2) – Integração e Evolução das Abordagens

¹ <http://www.sei.cmu.edu/productlines/ppl/>

Essa macro-atividade irá se concentrar: (i) na análise de integração das abordagens propostas para cada estágio de desenvolvimento; (ii) na definição de diretrizes explícitas para a realização dessa integração; e (iii) no refinamento e melhoria de cada uma das abordagens em função das limitações identificadas na MA1 e nas adaptações necessárias para a sua integração. Inicialmente, serão desenvolvidos refinamentos e melhorias para cada uma das abordagens de acordo com os resultados obtidos na MA1. Em seguida, serão elaboradas diretrizes para o uso integrado das abordagens, e dependendo das limitações anteriormente identificadas, uma ou mais abordagens poderão ser refinadas antes da sua completa integração. Alternativamente, poderá ser observado que abordagens para um mesmo estágio têm propósito similar, mas cada uma delas se aplica a cenários específicos.

Além de promover a integração de abordagens pertencentes a um mesmo estágio, essa atividade também irá analisar a integração entre abordagens pertencentes a diferentes atividades da engenharia de domínio. Grupos de pesquisa trabalhando em atividades subsequentes (por exemplo, análise e projeto de domínio, projeto e implementação de domínio) irão interagir para analisar as possibilidades de integração de suas abordagens. Como resultado dessa macro-atividade serão apresentadas diretrizes de como as abordagens propostas para diferentes atividades podem ser adotadas de forma integrada. Esperamos também oferecer suporte para a transição gradativa de um artefato para o outro, através da definição de mapeamentos e transformações, e possivelmente suportado pelas ferramentas já sendo exploradas pelos grupos de pesquisa.

Um ponto importante a ressaltar nessa macro-atividade é que a análise da integração das abordagens irá focalizar diferentes estratégias de adoção [41] de desenvolvimento de LPS, sendo elas: (i) proativas – nessa estratégia todos os produtos pertencentes a LPS são considerados desde o início do desenvolvimento e todos os artefatos da LPS são desenvolvidos do “zero”; (ii) extrativa – nessa estratégia de adoção a LPS é desenvolvida a partir de um conjunto de artefatos existentes para um dado produto; e (iii) reativa – nessa estratégia a LPS é desenvolvida de forma incremental, inicialmente são considerados apenas um subconjunto dos produtos, e gradativamente vai reagindo para atender novas *features* de outros produtos.

Macro-Atividade 3 (MA3) – Aplicação em um novo Estudo de Caso

Essa macro-atividade irá avaliar os resultados das diretrizes propostas para integração (oriundos da MA2), através da aplicação das abordagens em uma nova LPS. As seguintes LPS são candidatas para serem usadas nessa fase: (i) um middleware, tal como, o Ginga para TV Digital [32]; (ii) uma LPS de jogos para dispositivos móveis [7, 30]; e/ou (iii) uma LPS para sistemas multi-agentes [39].

Baseado nas diretrizes de integração propostas e nos refinamentos elaborados para cada uma das abordagens, os grupos de pesquisa irão especificar, projetar e implementar uma nova LPS. Os resultados, dificuldades e lições aprendidas serão documentados em relatórios técnicos. O projeto pretende gerar como resultado dessa atividade uma metodologia para uso das abordagens dos grupos de pesquisa envolvidos. Um relatório técnico também será escrito para documentar e ilustrar o uso de tal metodologia. Finalmente, os artefatos resultantes do desenvolvimento do estudo de caso escolhido usando as nossas abordagens serão também disponibilizados no portal *web* do projeto.

4. Cronograma de Atividades

O cronograma abaixo está previsto para realização das macro-atividades descritas na seção 3. Cada uma delas será posteriormente detalhada e acompanhada continuamente pelo coordenador do projeto juntamente com os professores líderes em cada uma das outras universidades participantes.

Macro-Atividade	2008	2009	2010	2011
MA1				
MA2				
MA3				

5. Metas e Resultados Esperados

As seguintes metas e resultados diretos são esperados como resultado desse projeto de pesquisa ao longo de seus quatro anos:

- 8 relatórios técnicos gerados como resultado das macro-atividades MA1, MA2 e MA3 descritas anteriormente;
- 8 teses de doutorado concluídas;
- 16 dissertações de mestrado concluídas;
- 2 estudos de caso ilustrando o uso integrado das abordagens propostas pelos grupos de pesquisa;
- 50 artigos científicos publicados em periódicos e conferência internacionais e nacionais de qualidade;
- 3 tutoriais ou mini-cursos descrevendo o uso combinado de técnicas de LPS e DSOA;
- 1 ferramenta para refatoração de LPS e 1 ferramenta de derivação de produto, propostas pelo grupo, na forma de plugins Eclipse. Espera-se que outras ferramentas para desenvolvimento de LPS sejam desenvolvidas e prototipadas ao longo desse projeto;
- organização de no mínimo 2 *workshops* nacionais para divulgar e compartilhar os resultados do projeto com a comunidade científica brasileira;
- preparação e oferta de novas disciplinas avançadas de engenharia de software que contemplem o estado da arte das áreas de LPS e DSOA abordadas no projeto.

Toda a produção mencionada acima será disponibilizada em um portal *web* de divulgação do projeto e das áreas de pesquisa envolvidas. Além desses resultados diretos, o projeto irá contribuir para aumentar a cooperação entre os grupos de pesquisa envolvidos no projeto, consolidando-os estrategicamente num tópico de extrema relevância internacional para a comunidade de engenharia de software. O projeto também contribuirá significativamente para melhorar a formação dos alunos participantes em todos os níveis (doutorado, mestrado, graduação), através

da troca de experiências entre professores e pesquisadores dos diferentes grupos de pesquisa e aprofundamento em técnicas avançadas de modelagem e implementação de LPS e DSOA. Esses tópicos não são, em geral, contemplados por cursos de graduação e pós-graduação do país. A realização de *workshops* e apresentação de tutoriais/mini-cursos em eventos nacionais contribuirá para disseminar os resultados do projeto para outros professores e pesquisadores e conseqüentemente atrair o interesse de mais gente para conhecer a área.

6. Fundamentação Teórica e Linhas de Pesquisa

Essa seção descreve a fundamentação teórica do projeto, assim como as linhas de pesquisa que serão exploradas. Essas linhas de pesquisa são, portanto, as pretendidas após o término do projeto. Para cada linha de pesquisa são apresentados os trabalhos atuais na área de LPS e DSOA já produzidos por professores e pesquisadores dos grupos de pesquisa envolvidos, finalmente alguns dos desafios enfrentados por pesquisadores naquele tópico são discutidos.

Como mencionado na seção introdutória desse documento, abordagens de desenvolvimento de LPS são tipicamente organizadas em um conjunto de atividades de engenharia de domínio e engenharia de aplicação. A engenharia de domínio busca a elaboração de uma arquitetura comum para a LPS que contemple suas *features* comuns e variáveis e um conjunto de artefatos reusáveis a serem usados na produção de produtos da linha. A engenharia de aplicação se concentra no desenvolvimento de produtos a partir da reutilização dos artefatos produzidos na engenharia de domínio.

A fase de engenharia de domínio é normalmente estruturada em atividades de análise, projeto e implementação de domínio. A análise de domínio envolve a definição do escopo da linha de produto e a identificação de *features* comuns e variáveis. O projeto de domínio envolve a definição de uma arquitetura comum que contemple todos os produtos da LPS. Tipicamente um plano de produção é também gerado nessa fase mostrando como um dado produto pode ser obtido baseado naquela arquitetura. Finalmente, a etapa de implementação de domínio envolve a codificação de artefatos reusáveis presentes na arquitetura da LPS, tais como, *frameworks*, componentes, bibliotecas, geradores e linguagens específicas de domínio (DSLs).

Durante a fase de engenharia de aplicação ocorre a derivação de produtos, que consiste no processo de construção de membros da LPS a partir do conjunto de artefatos especificados ou implementados durante a engenharia de domínio. Tal processo de derivação de produtos pode ocorrer de forma: (i) manual – baseado em um plano de produção [66] produzido durante a engenharia de domínio; ou (ii) automática – guiado pelo uso de linguagens específicas de domínio (DSLs) [26] ou pelo próprio modelo de features [1, 3]. Abordagens modernas, como programação generativa [26] e fábricas de software [35], têm ressaltado a importância de prover mecanismos automáticos, baseados em linguagens específicas de domínio e geradores, de forma a aumentar a produtividade de reuso dos artefatos produzidos na engenharia de domínio. Várias ferramentas de derivação de produtos baseadas no modelo de features [1, 3] ou em DSLs [35] têm sido propostas.

Neste projeto de pesquisa serão exploradas todas as atividades da engenharia de domínio, sendo elas: análise, projeto e implementação de domínio. Na engenharia de aplicação, serão exploradas e refinadas ferramentas de derivação automática já propostas anteriormente por membros dos grupos de pesquisa. As subseções seguintes discutem brevemente desafios que serão explorados dentro de cada linha de pesquisa do projeto, assim como descreve abordagens já propostas pelos grupos de pesquisa participantes.

6.1 Análise de Domínio

A engenharia de requisitos tem sido considerada uma atividade chave no processo da engenharia de software. Sendo de amplo conhecimento que uma boa especificação de requisitos deva incluir, não apenas a especificação do próprio software, mas, sobretudo requisitos relacionados à estrutura organizacional e a outros tipos de informações que descrevem o contexto no qual o sistema desejado funcionará. No contexto de LPSs é possível diferenciar engenharia de requisitos para engenharia de domínio da engenharia de requisitos para engenharia de aplicação. Através da engenharia de requisitos para engenharia de domínio é possível especificar as partes comuns e a variabilidade de uma LPS. Durante as atividades de engenharia de aplicação, os requisitos para as aplicações são definidos e documentados levando em conta a variabilidade da LPS. Um pré-requisito para definição de aplicações individuais é a representação explícita da variabilidade da LPS na engenharia de domínio. De forma a atingir tal objetivo, é necessário desenvolver uma linguagem de modelagem que permita representar e gerenciar a variabilidade de LPS de forma que se possa ilustrar explicitamente tanto as variações possíveis e seus detalhes, como relacionar as variações através de *links* pra artefatos de requisitos (metas, cenários e requisitos orientados a soluções). Também é preciso desenvolver técnicas para garantir a consistência entre os artefatos de requisitos e a variabilidade da linha de produto.

No contexto desse projeto, na etapa de análise de domínio um dos objetivos será explorar os benefícios que técnicas OA [12, 13, 54] podem trazer para melhor expressar os relacionamentos transversais que podem ocorrer entre *features* e requisitos de uma LPS, em diferentes técnicas de especificação e notações já sendo investigadas pelo grupo. Serão investigados que tipos de relacionamentos transversais podem ocorrer entre *features* e requisitos de LPSs, e de que forma as abordagens já propostas permitem endereçá-las. Será também explorado como as abordagens atuais lidam com o problema de interação entre *features* e as limitações existentes. Outro ponto que merece atenção para essa linha de pesquisa é avaliar de que forma as abordagens de especificação de requisitos de LPS propostas pelos membros dos grupos de pesquisa envolvidos permite expressar suas variabilidades. Finalmente, serão também investigadas técnicas para manter o rastreamento entre *features/requisitos* de uma LPS e seus respectivos artefatos de projeto e implementação.

Dentro do contexto dos desafios mencionados, o grupo da UFRN pretende explorar juntamente com o grupo da PUC-Rio, a adaptação da linguagem AOV-Graph [61] já proposta para cenários de desenvolvimento de LPS. AOV-Graph é uma extensão do modelo V-Graph [67] que define mecanismos para separar, compor e visualizar conceitos transversais na fase de requisitos. Pretende-se avaliar a aplicabilidade de AOV-Graph para modelar os requisitos de uma LPS bem como especificar as variabilidades e a composição entre eles. A interação entre tais grupos no projeto também pretende explorar como se dá o mapeamento de representações em AOV-Graph para notações arquiteturais existentes para projeto de arquiteturas de LPS.

O grupo da UFPE irá dar continuidade a uma abordagem que busca formalizar técnicas de suporte a variação de requisitos em LPS, através do uso combinado de modelos de *features* e cenários/*use cases*. O trabalho busca atender a limitações de trabalhos já propostos [15, 29] que não permitem expressar claramente a separação entre os artefatos usados para descrever cenários de casos de uso e os artefatos usados para definir as variações e configurações específicas da LPS. Isso traz como consequência dificuldades para evoluir ambas as representações de forma isolada. Outra limitação dos trabalhos existentes é a falta de formalização – não está claro como os pontos de variação são instanciados com as especificações de requisitos. Um *framework* para modelagem de variação em requisitos está sendo desenvolvido pelo grupo. Tal *framework* está sendo utilizado para representar a semântica de diferentes mecanismos de variação (cenários opcionais, composição de cenários e parametrização de cenários), tornando claro o papel de cada uma das linguagens envolvidas (linguagens de especificação de cenários, definição de características e configuração de produtos). Estudos de caso mais elaborados serão realizados no

contexto desse projeto com o intuito de verificar os benefícios do *framework* proposto. O grupo pretende também refinar o *framework* proposto e especializá-lo para contemplar outras técnicas de modelagem e especificação de requisitos (*Frames*, Modelos *i**, Padrões Sociais) já considerando o uso de técnicas de LPS e DSOA. Vários trabalhos recentes [4, 47, 60] desenvolvidos pelo grupo servirão de base para propor tais refinamentos.

6.2 Projeto de Domínio

O projeto de LPSs se dá geralmente de forma incremental e evolui gradualmente para lidar com novas necessidades dos interessados [8, 23, 41]. No entanto, a longevidade de uma linha de produto depende fortemente da habilidade de seu projeto arquitetural para sustentar a estrutura modular das características obrigatórias e variáveis a medida que sofre mudanças. De fato, a evolução de linhas de produtos preocupa bastante os engenheiros de software devido às diferentes naturezas dessas mudanças, tais como: introdução e remoção de características transversais e não-transversais, e a transformação de características obrigatórias em opcionais ou alternativas e vice-versa.

O projeto de domínio envolve a definição de uma arquitetura comum que atenda os diferentes produtos sendo considerados na LPS. Um dos tópicos importantes a ser considerado durante o projeto de domínio é entender melhor o impacto que a tecnologia de DSOA pode trazer para a elaboração e evolução de arquiteturas de LPS. Diferentes perspectivas podem ser exploradas, tais como: (i) documentação de decisões arquiteturas; (ii) especificação de conectores aspectuais [14]; (iii) especificação de componentes aspectuais com variabilidades [21]. Diversos pesquisadores [6, 37, 42, 64] têm enfatizado a importância de definir interfaces entre classes e aspectos, na forma de *design rules* (DRs) [11], de forma a reduzir as dependências e acoplamento entre eles. A definição e especificação de DRs é um outro tópico de pesquisa que será explorado por essa linha de pesquisa, com o objetivo de reduzir a dependência entre o núcleo da LPS (*core assets*) e suas variações, contribuindo para melhorar a modularidade de ambos [42, 44]. Finalmente, deverão também ser confrontadas notações arquiteturais e de projeto detalhado para documentação de arquitetura de LPS, algumas dessas já propostas por membros dos grupos de pesquisa envolvidos nessa submissão de projeto. Muitas das notações já propostas não focalizam explicitamente a representação de variabilidades.

Considerando os desafios mencionados, pretende-se avaliar o uso e extensão de notações arquiteturais e de projeto detalhado propostos pelos grupos da UFRN, PUC-Rio e USP-SC, tais como, AspectualACME [14], aSideML [21] e AIPLE-IS [16] no desenvolvimento de arquiteturas de LPS. Serão também avaliadas e refinadas diretrizes já propostas pelos grupos para projeto de LPS contemplando o uso de aspectos. O grupo da PUC-Rio em conjunto com a UFPE propôs um conjunto de diretrizes [42, 44] para extensão de LPS baseada em *frameworks* através do uso de aspectos. Essas diretrizes são centradas na definição de DRs, na forma de um conjunto de pontos de junção de extensão oferecidas pelo núcleo da LPS para ser estendidas pelos aspectos. Atualmente, o grupo da UFPE vem investigando a existência de novos tipos de dependência em programas orientados a aspectos [28, 51, 56]. Uma das direções atuais de tal trabalho é ampliar a expressividade das DRs para permitir gerenciar as dependências entre artefatos do núcleo da LPS e de suas respectivas variações, muitas delas modularizadas como aspectos. O grupo da USP-SC também propôs um conjunto de diretrizes para uso de aspectos na evolução de linhas de produto de sistemas de informação [16, 52]. Um dos objetivos centrais de tais diretrizes é diminuir o acoplamento entre *features* de uma LPS e aumentar a coesão individual deles.

Finalmente, baseado em experiências anteriores [36, 57], pretende-se avaliar sistematicamente a efetividade do uso das diretrizes de projeto, através do desenvolvimento de métricas específicas que permitam a avaliação de modularidade da arquitetura de linhas de

produtos orientadas a aspectos. Em particular, pretende-se usar essas métricas em estudos experimentais com o objetivo de: (i) comparar a modularidade de arquiteturas de LPS convencionais e orientadas a aspectos, e (ii) avaliar como a modularidade arquitetural de LPS é afetada ao longo de sua evolução.

6.3 Implementação de Domínio

A arquitetura de uma LPS é formada por um conjunto de artefatos de implementação que juntos atendem as *features* obrigatórias e variáveis (opcionais, alternativas, etc) definido para a LPS. Decidir como combinar os artefatos que modularizam cada uma de tais *features* é uma tarefa bastante desafiadora. Em particular, a escolha da técnica correta para implementar variações pode acarretar efeitos consideráveis no custo de criação e evolução de uma LPS. Apesar da importância dessa questão, a escolha das técnicas de implementação de variações ainda ocorre de maneira pouco sistemática durante o projeto de muitas LPSs. Alguns estudos já foram realizados [9, 24] com o objetivo de comparar as principais técnicas usadas para implementar variações. A maioria de tais estudos, entretanto, não fornece uma maneira sistemática de escolher a melhor técnica de implementação para cada caso específico.

Nesse contexto, todos os grupos de pesquisa participantes desse projeto pretendem investigar os benefícios e limitações trazidos por novas técnicas de implementação de DSOA, tais como, aspectos [40] e *mix-in-layers* [50, 62]. Será também explorado como tais tecnologias podem ser combinadas com técnicas já estabelecidas, tais como, geração de código, *templates*, bibliotecas, compilação condicional e *frameworks* OO. Vários trabalhos desenvolvidos por membros dos grupos de pesquisa já iniciaram a exploração de tais tópicos [6, 8, 19, 20, 28, 42, 44, 51]. O grupo da UFPE em colaboração com os outros grupos de pesquisa pretende refinar um modelo de decisão [55] capaz de indicar a técnica, ou conjunto de técnicas, que melhor se aplica a cada situação. Mais especificamente, tal modelo de decisão poderá receber como entrada características como localização e natureza da variação a ser implementada, bem como restrições específicas de cada projeto, fornecendo como saída, a técnica (ou conjunto de técnicas) que melhor se adequam a implementação de uma dada variação. Realizaremos experimentos qualitativos e quantitativos para embasar as decisões do modelo. Os grupos da PUC-Rio e UFPE já vêm realizando vários experimentos quantitativos e qualitativos conjuntos que buscam avaliar e comparar a implementação tanto de sistemas OO e AO [17, 31, 34, 36, 46, 58], quanto de linhas de produto [8, 30].

Durante os estudos relacionados à implementação de domínio, será também dado enfoque para a resolução do problema de interação entre *features*, e como os mecanismos disponíveis em linguagem de programação OA atuais permite (ou não) tratá-los. Finalmente, esse projeto pretende também investigar o suporte lingüístico para a especificação de interfaces (na forma de *Design Rules*) entre o código base OO e aspectos, e como tais interfaces podem ser derivadas automaticamente a partir dos modelos e notações investigados na área de projeto de domínio.

6.4 Ferramentas de Linha de Produto de Software

Na linha de pesquisa de ferramentas para desenvolvimento de LPS, esse projeto pretende dar continuidade a investigação já iniciada por membros do grupo de pesquisa da PUC-Rio e USP-SC, na área de ferramentas de derivação de produto baseada em modelos [63]. Será explorado como uma ferramenta de derivação de produto cuja requisição de produtos é feita através de modelos de *features*, pode ser combinada com outras DSLs baseadas em modelos ou textuais para prover a geração de variabilidades para *features* cuja síntese seja mais complexa. As ferramentas GenArch [22, 43] e Captor [59], propostas pelos grupos da PUC-Rio e USP-SC, respectivamente,

serão usadas como base para realização de tais experimentos. Ambas as ferramentas já vem sendo estendidas para permitir a instanciação de artefatos de implementação OA.

Na parte de ferramentas, pretende-se também explorar a integração entre ferramentas de refatoração que permitem extrair/isolar determinados *features* em aspectos e as ferramentas de derivação mencionadas anteriormente. O grupo da UFPE desenvolveu a ferramenta FLIPex [18] que permite a adoção de duas abordagens: a primeira consiste em partir de um produto para a geração de uma linha de produtos; a segunda, permite partir de uma linha de produtos estruturada com compilação condicional para uma linha de produtos estruturada usando programação orientada a aspectos. O processo utilizado no FLIPex consiste de refatorações de código. Assim sendo, seguindo, por exemplo, a primeira abordagem o usuário pode selecionar o pedaço do código que ele deseja extrair. Posteriormente, a ferramenta extrai o código selecionado e o coloca em um aspecto.

Outra linha de investigação que será explorada no projeto é o impacto que tecnologias de Desenvolvimento Dirigido a Modelos (DDM) [63] combinadas com as de DSOA podem trazer para ferramentas de LPS durante a engenharia de domínio. O grupo da UFPE já iniciou investigação no desenvolvimento de ferramentas que permite manter o rastreamento entre artefatos de requisitos e testes em LPS. O projeto pretende dar continuidade e expandir essa linha de pesquisa para permitir a criação e manutenção das relações de rastreamento (*traceability links*) [38] existente entre os diferentes artefatos e modelos sendo gerados ao longo da análise, projeto e implementação de domínio. Manter o rastreamento entre esses diferentes artefatos ajuda não apenas a entender melhor como um dado *feature/requisito* está sendo atendido pelos artefatos de projeto e implementação, mas também avaliar o impacto de modificar um *feature/requisito* e possibilita um gerenciamento consistente das variabilidades da LPS. Dentro dessa linha, as abordagens AIPLE-IS da USP-SC [16, 27] e CrossMDA da UFRN [5] que cobrem todo o ciclo de desenvolvimento de uma LPS, definindo atividades, artefatos, refinamentos e transformações para todo esse processo, poderão ser usadas como referência para possibilitar a definição de tais relações de rastreamento.

Finalmente, além das ferramentas de rastreamento pretende-se também investigar a definição e especificação de transformações entre os diferentes artefatos da engenharia de domínio. Uma iniciativa nessa linha será avaliar e adaptar a ferramenta MARISA (*Mapping Requirements to Software Architecture*) [49], desenvolvida pelo grupo da UFRN, no contexto de LPS. Ela é baseada em tecnologias MDA e automatiza o mapeamento entre modelos de requisitos orientados a aspectos em AOV-Graph e modelos de descrição arquitetural orientados a aspectos em AspectualACME.

7. Equipes Participantes do Projeto

Este projeto terá a participação direta de 13 professores/pesquisadores doutores de 4 universidades brasileiras (UFPE, PUC-Rio, USP-SC, UFRN). Os programas de doutorado e mestrado da PUC-Rio, UFPE e USP-SC são todos de excelência na área de Ciência da Computação de acordo com avaliação da CAPES, tendo sido avaliados como programas 7, 6 e 5, respectivamente, no último triênio. O programa da UFRN recém criou seu doutorado e já vinha recebendo avaliação máxima 4 no mestrado pela CAPES. A cooperação oriunda desse projeto irá contribuir não apenas para manter o excelente nível de qualidade dos programas de pós-graduação da PUC-Rio, UFPE e USP-SC, como também para consolidar o novo programa de doutorado da UFRN.

Abaixo é apresentada a lista dos 12 professores/pesquisadores doutores participantes do projeto. Diversos alunos de doutorado, mestrado e iniciação científica associados diretamente a tais grupos de pesquisa serão também envolvidos nas atividades e missões de pesquisa do projeto.

Nome	Instituição / Cargo
1. Paulo Henrique Monteiro Borba, PhD (Coordenador do Projeto)	UFPE, Professor Adjunto
2. Jaelson Castro, PhD	UFPE, Professor Adjunto
3. Fernanda Campos, DSc	UFPE, Professora Adjunta
4. Uirá Kulesza, DSc	UFPE, Pesquisador Associado
5. Carlos José Pereira de Lucena, PhD	PUC-Rio, Professor Titular
6. Arndt von Staa, PhD	PUC-Rio, Professor Associado
7. Julio César Sampaio Leite, PhD	PUC-Rio, Professor Associado
8. Paulo César Masiero, DSc	USP-SC, Professor Titular
9. Rosana Vacare Braga, DSc	USP-SC, Professora Adjunta
10. Renata Pontin Fortes, DSc	USP-SC, Professora Associada
11. Paulo de Figueiredo Pires, DSc	UFRN, Professor Adjunto
12. Flávia Coimbra Delicato, DSc	UFRN, Professora Adjunta
13. Thaís Vasconcelos Batista, DSc	UFRN, Professora Adjunta

Todos os professores e pesquisadores envolvidos neste projeto possuem atuação de destaque na área de engenharia de software, o que pode ser verificado através de várias publicações em periódicos e conferências internacionais e nacionais na área. Segue abaixo uma lista parcial de contribuições dos professores envolvidos nessa submissão de projeto ao longo dos últimos anos. Detalhes adicionais podem ser obtidos em seus respectivos currículos Lattes.

- Os professores Paulo Borba (UFPE), Carlos Lucena (PUC-Rio), Paulo Masiero (USP-SC) e Thaís Batista (UFRN) têm liderado diversas pesquisas junto a seus respectivos grupos na área de DSOA e LPS ao longo dos últimos anos. Várias teses de doutorado e dissertações de mestrado foram orientadas por eles em tais tópicos ao longo dos últimos anos;
- Os professores e pesquisadores participantes do projeto ministraram ao longo dos últimos anos vários tutoriais e mini-cursos sobre o tema de DSOA no principal simpósio de engenharia de software do Brasil (SBES). Além disso, foram também apresentados diversos artigos de qualidade em tal evento, alguns destes recebendo inclusive prêmios de melhores artigos;
- Os grupos de pesquisa da PUC-Rio e da UFPE foram responsáveis pela criação do I Workshop Brasileiro de DSOA (WASP) em 2003. Junto com outros grupos de pesquisa, incluindo a USP-SC e UFRN, eles foram responsáveis pela co-organização dos *workshops* brasileiros em 2005 e 2006. Em 2007, o evento foi ampliado para ser latino-americano, e foi criado o I *Latin-American Workshop on Aspect-Oriented Software Development* (LA-WASP'2007);
- O professor Carlos Lucena (PUC-Rio) é atualmente coordenador da rede de pesquisas Latino-Americana na área de DSOA, denominada Latin-AOSD, financiada pelo CNPq. A rede conta com a participação de cerca de 9 universidades brasileiras e mais 5 estrangeiras do Chile, Argentina e Colômbia;
- Vários professores/pesquisadores participantes do projeto têm atuado como membro de comitê de programa ou revisores do principal evento da área, a conferência internacional AOSD (www.aosd.net), assim como do único periódico específico da área o *Lecture Notes Transactions on Aspect-Oriented Software Development*, da Springer-Verlag.

Referências

- [1] *Gears*. URL: <http://www.biglever.com>. 2007.
- [2] *Product Line - Hall of Fame*, http://www.sei.cmu.edu/productlines/plp_hof.html. 2007.
- [3] *Pure::Variants*. URL: <http://www.pure-systems.com/>. 2007.
- [4] Alencar, F., et al. *Integration of Aspects with i* Models*. *Agent-Oriented Information Systems IV*. Springer-Verlag. pp., 2007 (to appear)
- [5] Alves, M., et al. *CrossMDA: Arcabouço para Integração de Interesses Transversais no Desenvolvimento Orientado a Modelos*. *Proceedings of Simpósio Brasileiro de Componentes, Arquitetura e Reutilização de Software (SBCARS'2007)*. pp. 31-46, August 2007. Campinas, Brazil
- [6] Alves, V., *Implementing Software Product Line Adoption Strategies*, *PhD Thesis*. March 2007, Centro de Informática, Universidade Federal de Pernambuco.
- [7] Alves, V., et al., *Extracting and evolving code in product lines with aspect-oriented programming*. *LNCSTransactions on Aspect-Oriented Software Development (TAOSD): Special Issue on Software Evolution, 2007 (to appear), 2007*.
- [8] Alves, V., et al. *Extracting and Evolving Mobile Games Product Lines*. *Proceedings of Software Product Line Conference (SPLC'2005)*. Springer-Verlag. pp. 70-81, 2005
- [9] Anastasopoulos, M. and C. Gacek. *Implementing Product Line Variabilities*. *Proceedings of the Symposium on Software Reusability (SSR'01)*. ACM Press. pp. 109-117, 2001. New York, NY, USA
- [10] Apel, S. and D. Batory. *When to Use Features and Aspects?: a Case Study*. *Proceedings of the 5th international conference on Generative programming and component engineering*. ACM Press. pp. 59-68, 2006. Portland, Oregon, USA
- [11] Baldwin, C.Y. and K.B. Clark, *Design Rules: The Power of Modularity*. 2000, Cambridge, MA: MIT Press.
- [12] Baniassad, E., *Discovering Early Aspects*. *IEEE Software*, Special Issue on Aspect-Oriented Programming, 2006. **23**(1): pp. 61-70.
- [13] Baniassad, E. and S. Clarke, *Theme: An Approach for Aspect-Oriented Analysis and Design*, in *Proceedings of the 25th international conference on Software engineering (ICSE'04)*. 2004: Scotland.
- [14] Batista, T., et al. *Aspectual Connectors: Supporting the Seamless Integration of Aspects and ADLs*. *XX Simpósio Brasileiro de Engenharia de Software (SBES'2006)*. pp., 2006. Florianópolis
- [15] Bertolino, A. and S. Gnesi, *Use case-based testing of product lines*, in *Proceedings of the 11th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'03)*. 2003, ACM Press: New York, NY, USA.
- [16] Braga, R., et al. *An Approach to Develop Product Lines for Information Systems Using Aspects*. *Proceedings of Simpósio Brasileiro de Componentes, Arquitetura e Reutilização de Software (SBCARS'2007)*. pp. 17-30, August 2007. Campinas, Brazil
- [17] Cacho, N., et al. *Composing Design Patterns: a Scalability Study of Aspect-Oriented Programming*. *Proceedings of the 5th International Conference on Aspect-oriented Software Development*. ACM Press. pp. 109-121, 2006. Bonn, Germany
- [18] Calheiros, F., et al. *Product Line Variability Refactoring Tool*. *Proceedings of the 1st Workshop on Refactoring Tools (WRT'07), in conjunction with the 21st European Conference on Object-Oriented Programming (ECOOP'07)*. Berlin. pp. 33-34, 2007
- [19] Camargo, V. and P. Masiero. *Frameworks Orientado a Aspectos*. *IX Simpósio Brasileiro de Engenharia de Software (SBES'2005)*. pp., 2005. Uberlândia
- [20] Camargo, V.V.d., *Frameworks Transversais: Definições, Classificações, Arquitetura e Utilização em um Processo de Desenvolvimento de Software*, *PhD Thesis*. September 2006, Instituto de Ciências Matemáticas e de Computação (ICMC), Universidade de São Paulo.
- [21] Chavez, C., et al., *Crosscutting Interfaces for Aspect-Oriented Modeling*. *Journal of the Brazilian Computer Society*, 2006. **11**(3): pp. 43-58.
- [22] Cirillo, E., U. Kulesza, and C. Lucena. *GenArch – A Model-Based Product Derivation Tool*. *Proceedings of Simpósio Brasileiro de Componentes, Arquitetura e Reutilização de Software (SBCARS'2007)*. pp. 31-46, August 2007. Campinas, Brazil
- [23] Clements, P. and L. Northrop, *Software Product Lines: Practices and Patterns*. 2001: Addison-Wesley Professional.

- [24] Coplien, J., *Multi-Paradigm Design, PhD Thesis*. July 2000, Vrije Universiteit Brussel, Etterbeek, Belgium.
- [25] Czarnecki, K. *Overview of Generative Software Development. Proceedings of the European Commission and US National Science Foundation Strategic Research - Workshop on Unconventional Programming Paradigms*. pp., 2004. Mont Saint-Michel, France
- [26] Czarnecki, K. and U.W. Eisenecker, *Generative Programming: Methods, Tools, and Applications*. 2000: ACM Press/Addison-Wesley Publishing Co. 832.
- [27] Donegan, P. and P. Masiero. *Design Issues in a Component-based Software Product Line. Proceedings of Simpósio Brasileiro de Componentes, Arquitetura e Reutilização de Software (SBCARS'2007)*. pp. 3-16, August 2007. Campinas, Brazil
- [28] Dósea, M., et al. *Specifying Design Rules in Aspect-Oriented Systems. Anais do Latin Workshop on Aspect-Oriented Software Development (LA-WASP 2007)*. pp. 43-54, Outubro 2007. João Pessoa, PB
- [29] Eriksson, M., J. Borstler, and K. Borg. *The pluss approach - domain modeling with features, use cases and use case realizations. Proceedings of Software Product Line Conference (SPLC'2005)*. Springer-Verlag. pp., 2005
- [30] Figueiredo, E., et al., *Evolving Software Product Lines with Aspects: An Empirical Study on Design Stability, in Proceedings of the 29th international conference on Software engineering (ICSE'07)* 2008: Leinz, Germany (to appear).
- [31] Filho, F.C., et al., *Exceptions and Aspects: the Devil is in the Details*, in *Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. 2006, ACM Press.
- [32] Filho, G., L.E. Leite, and C.E. Freire, *Ginga-J: The Procedural Middleware for the Brazilian Digital TV System*. Journal of the Brazilian Computer Society, 2007. **13**(4): pp. 47-56.
- [33] Filman, R., et al., *Aspect-Oriented Software Development*. 2005: Addison-Wesley.
- [34] Garcia, A., et al. *Modularizing Design Patterns with Aspects: A Quantitative Study. Proceedings of the 4th international conference on Aspect-oriented software development*. ACM Press. pp. 3-14, 2005.
- [35] Greenfield, J. and K. Short, *Software Factories: Assembling Applications with Patterns, Frameworks, Models and Tools*. 2005: John Wiley and Sons.
- [36] Greenwood, P., et al. *On the Impact of Aspectual Decompositions on Design Stability: An Empirical Study. European Conference of Object-Oriented Programming (ECOOP'07)*. Springer-Verlag. pp. 176-200, 2007
- [37] Griswold, W., et al., *Modular Software Design with Crosscutting Interfaces*. IEEE Software, Special Issue on Aspect-Oriented Programming, 2006. **23**(1): pp. 51-60.
- [38] Jarke, M., *Requirements Tracing*. Commun. ACM, 1998. **41**(12): pp. 32-36.
- [39] Joaquin, P., G.H. Michael, and R.-C. Antonio, *Multi-agent system product lines: challenges and benefits*. Commun. ACM, 2006. **49**(12): pp. 82-84.
- [40] Kiczales, G. *Aspect-Oriented Programming. European Conference of Object-Oriented Programming (ECOOP'97)*. Springer-Verlag. pp. 220-242, 1997
- [41] Krueger, C. *Easing the Transition to Software Mass Customization. 4th International Workshop on Software Product-Family Engineering (PFE'2001)*. pp. 282-293, 2001
- [42] Kulesza, U., et al., *Improving Extensibility of Object-Oriented Frameworks with Aspect-Oriented Programming*, in *Proceedings of 9th International Conference on Software Reuse, ICSR 2006 Turin, Italy, June 12-15, 2006. Lecture Notes in Computer Science: Reuse of Off-the-Shelf Components*. 2006, Springer-Verlag. pp. 231-245.
- [43] Kulesza, U., et al. *Mapping Features to Aspects: A Model-Based Generative Approach. Early Aspects 2007 Workshop, AOSD'2007*. Springer-Verlag. pp. 155-174, 2007. Vancouver, Canada
- [44] Kulesza, U., et al. *Implementing Framework Crosscutting Extensions with XPIs and AspectJ. Proceedings of XX Simpósio Brasileiro de Engenharia de Software (SBES'2006)*. pp. 177-192, 2006. Florianópolis
- [45] Kulesza, U., et al. *Integrating Generative and Aspect-Oriented Technologies. Proceedings of XVIII Simpósio Brasileiro de Engenharia de Software (SBES'2004)*. pp. 130-146, October 2004. Brasília, Brazil
- [46] Kulesza, U., et al. *Quantifying the Effects of Aspect-Oriented Programming: A Maintenance Study. Proceedings of 22nd IEEE International Conference on Software Maintenance (ICSM 2006)*. IEEE Computer Society pp. 223-233, September 2006. Philadelphia, Pennsylvania, USA
- [47] Lencastre, M., et al., *Towards Aspectual Problem Frames: an example*. Expert Systems The Journal of Knowledge Engineering (to appear).

- [48] Lopez-Herrejon, R.E., D. Batory, and W. Cook. *Evaluating Support for Features in Advanced Modularization Technologies. Proceedings of the 19th European Conference on Object-Oriented Programming (ECOOP'05)*. Springer-Verlag. pp. 169-194, 2005
- [49] Medeiros, A., et al. *MARISA - Uma Ferramenta para Mapeamento bidirecional de Modelos Orientados a Aspectos: Requisitos e Arquitetura de Software. Anais do Latin Workshop on Aspect-Oriented Software Development (LA-WASP 2007)*. pp. 43-54, Outubro 2007. João Pessoa, PB
- [50] Mezini, M. and K. Ostermann. *Variability Management with Feature-Oriented Programming and Aspects. Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering*. ACM Press. pp. 127-136, 2004. Newport Beach, CA, USA
- [51] Neto, A.C., et al. *Semantic dependencies and modularity of aspect-oriented software. In ICSE Workshop on Assessment of Contemporary Modularization Techniques (ACoM.07), affiliated with ICSE 2007*. pp., May, 2007. Minneapolis, USA,
- [52] Pacios, S., P. Masiero, and R. Braga. *Guidelines for Using Aspects to Evolve Product Lines. III Workshop Brasileiro de Desenvolvimento de Software Orientado a Aspectos (WASP 2006)*. pp. 111-120, 2006. Florianópolis
- [53] Parnas, D.L., *On the Design and Development of Program Families*. IEEE Transactions on Software Engineering (TSE), 1976. 2(1): pp. 1-9.
- [54] Rashid, A., A. Moreira, and J. Araújo, *Modularisation and Composition of Aspectual Requirements, in Proceedings of the 2nd International Conference on Aspect-oriented Software Development*. 2003, ACM Press: Boston, Massachusetts.
- [55] Ribeiro, M., P.M. Jr., and P. Borba, *A Decision Model for Implementing Product Lines Variabilities, in Proceedings of the 23th ACM Symposium on Applied Computing (SAC'08)*. March 2008 (to appear).
- [56] Ribeiro, M.d.M., et al. *Analyzing class and crosscutting modularity with design structure matrixes. Proceedings of XXI Brazilian Symposium on Software Engineering - SBES'07*. pp. 130-146, October 2007. João Pessoa-PB, Brazil
- [57] Sant'Anna, C., et al. *On the Modularity of Software Architectures: A Concern-Driven Measurement Framework. 1st European Conference on Software Architecture (ECSA 2007)*. pp., 2007. Madrid, Spain
- [58] Sant'Anna, C., et al. *On the Reuse and Maintenance of Aspect-Oriented Software: An Assessment Framework. Proceedings of the XVII Brazilian Symposium on Software Engineering*. pp., October 2003. Manaus, Brazil
- [59] Shimabukuro, E., P. Masiero, and R. Braga. *Captor: Um Gerador de Aplicações Configurável. Anais da XIII Sessão de Ferramentas do SBES*. pp. 121-126, 2006. Florianópolis
- [60] Silva, C., et al. *Designing Social Patterns using Advanced Separation of Concerns. CAISE'07 - 19th Conference on Advanced Information Systems Engineering*. Springer-Verlag. pp. 309-323, 2007. Trondheim, Norway
- [61] Silva, L. *An Aspect-Oriented Approach to Model Requirements. Proc. of the Doctoral Consortium on Requirements Engineering, In conjunction with the Requirement Engineering Conference, 2005*. . pp., 2005
- [62] Smaragdakis, Y. and D. Batory, *Mixin Layers: an object-oriented implementation technique for refinements and collaboration-based designs*. ACM Trans. Softw. Eng. Methodol., 2002. 11(2): pp. 215-255.
- [63] Stahl, T. and M. Voelter, *Model-Driven Software Development: Technology, Engineering, Management*. 2006: Wiley.
- [64] Sullivan, K., et al. *Information Hiding Interfaces for Aspect-Oriented Design. Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering*. ACM Press. pp. 166-175, 2005. Lisbon, Portugal
- [65] Vijayan, S., P. Sooyong, and C.K. Kyo, *Software Product Line Engineering: Introduction (Special Issue)*. Commun. ACM, 2006. 49(12): pp. 28-32.
- [66] Weiss, D. and C. Lai, *Software Product-Line Engineering: A Family-Based Software Development Process*. 1999: Addison-Wesley Professional.
- [67] Yu, E., J. Leite, and J. Mylopoulos. *From goals to aspects: discovering aspects from requirements goal models. Proceedings. of IEEE Int. Symp. on Requirements Engineering (RE'04)*. pp. 38-47, 2004. Japan
- [68] Zhang, C. and H.-A. Jacobsen. *Resolving Feature Convolution in Middleware Systems. Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*. ACM Press. pp. 188-205, 2004. Vancouver, BC, Canada