

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

**Uma Revisão Sistemática sobre a Implementação de
Variabilidades de LPS utilizando POA**

Antonio Carlos Contieri Júnior
Thelma Elita Colanzi

Maringá
2009

RESUMO

A abordagem de Linha de Produto de Software tem sido utilizada em larga escala para a produção de software devido a vários fatores, principalmente a questão da reusabilidade, que traz consigo diminuição do tempo e esforço para desenvolvimento e, conseqüentemente, redução de custo. Porém, tal abordagem envolve questões chaves, como a implementação das variabilidades, por exemplo. Neste contexto, dentre muitas propostas, surge a Programação Orientada a Aspectos. Desta forma, propõe-se uma revisão sistemática com o intuito de investigar a produção científica relacionada à aplicação de aspectos na implementação de variabilidades de LPS. Assim, este relatório contém todo o desenvolvimento da revisão sistemática. São apresentadas as etapas de planejamento, incluindo o protocolo, e de condução da revisão. Também são demonstrados os números obtidos, bem como a análise dos trabalhos incluídos ao final de todo o processo.

SUMÁRIO

Capítulo 1 – Introdução.....	5
Capítulo 2 – Planejamento e Condução da Revisão.....	7
2.1 Planejamento da Revisão.....	7
2.1.1 Objetivos.....	7
2.1.2 Questões de Pesquisa.....	7
2.1.2.1 Itens relacionados ao escopo ou especificidade das questões.....	8
2.1.3 Estratégia de busca para seleção de estudos primários.....	8
2.1.4 Critérios e Procedimentos para Seleção de Estudos.....	9
2.1.4.1 Critérios de Inclusão.....	9
2.1.4.2 Critérios de Exclusão.....	10
2.1.4.3 Processo de Seleção Preliminar.....	10
2.1.4.4 Processo de Seleção Final.....	10
2.1.5 Extração dos Resultados.....	10
2.2 Condução da Revisão.....	11
2.2.1 Definição das Strings de Busca.....	11
2.2.2 Buscas Realizadas e Obtenção dos Trabalhos.....	12
2.2.3 Seleção Preliminar dos Trabalhos.....	19
2.2.4 Seleção Final dos Trabalhos.....	26
Capítulo 3 – Análise dos Resultados.....	32
3.1 Caracterização do problema.....	32
3.2 Panorama sobre os Resultados Obtidos.....	34
3.2.1 Propostas e Abordagens para a implementação de variabilidades de LPS utilizando POA.....	35
3.2.1.1 Feature Oriented Approachs e Programação Orientada a Aspectos.....	35
3.2.1.2 Aplicação de Orientação a Aspectos em LPS Orientadas a Objetos.....	36
3.2.1.3 Decisões de implementação envolvendo POA.....	36

3.2.1.4 Refactoring de Sistemas Orientados a Objetos.....	37
3.2.1.5 Separação de Interesses.....	37
3.2.1.6 Abordagens generativas e POA.....	38
3.2.1.7 Model-Driven Development e POA.....	38
3.2.1.8 Utilização de POA em testes de LPS.....	39
3.2.1.9 Utilização de POA em arquiteturas de LPS.....	39
3.2.1.10 Agent Development e POA.....	39
3.2.1.11 Utilização de POA na implementação de requisitos funcionais.....	39
3.2.2 Estudos de caso.....	40
3.2.2.1 LPS para gerenciamento de Igrejas Cristãs.....	40
3.2.2.2 LPS de aplicações para aparelhos móveis.....	40
3.2.2.3 Frameworks.....	41
3.2.2.4 Marcapasso (<i>Pacemaker Product Line</i>).....	41
3.2.2.5 Sistemas Legados.....	41
3.2.2.6 Sistemas Automotivos.....	42
3.2.2.7 GPL (<i>Graph Product Line</i>).....	42
3.2.2.8 Comércio Eletrônico (e-shop).....	42
Capítulo 4 – Conclusão.....	43
Referências.....	45

Capítulo 1 – Introdução

A abordagem de Linha de Produto de Software (LPS) representa a criação de vários produtos de um determinado domínio de aplicação que apresentem características comuns e variáveis, estas últimas também conhecidas por variabilidades. As características comuns representam a base do funcionamento do produto e são compartilhadas por todos os membros da LPS. São conhecidas como o núcleo da LPS. As variabilidades, por sua vez, são as características que diferenciam os produtos entre si, permitindo então a criação de produtos específicos dentro do mesmo domínio.

O desenvolvimento de uma LPS engloba duas fases bem definidas: a Engenharia de Domínio e a Engenharia de Aplicação. A Engenharia de Domínio corresponde ao desenvolvimento de todas as características relativas a uma família de produtos, iniciando-se pelo desenvolvimento de núcleo. Nesta fase também podem ser definidos os meios pelos quais as características serão combinadas. Já a Engenharia de Aplicação corresponde à etapa de instanciação (ou derivação) do produto por meio da seleção das características desenvolvidas. É nesta etapa que os produtos são devidamente “fabricados”, a partir do que foi construído em relação ao domínio.

Várias técnicas têm sido empregadas para a implementação de LPS, especialmente no que diz respeito a variabilidades. A Programação Orientada a Aspectos (POA) é uma alternativa interessante para a definição de variabilidades e superação dos problemas apresentados [Mehner, Reiser e Weber 2006]. Ela provê os meios para separação de código que contém interesses transversais, encapsulando-os em uma unidade denominada aspecto, elemento principal da POA, que proporciona um maior reuso e modularização do software [Kiczales et al. 1997]. Neste contexto, é importante conhecer trabalhos que relatem técnicas, abordagens ou experiências do uso de POA para a implementação de variabilidades de LPS.

Uma revisão sistemática consiste de um método para realizar pesquisa bibliográfica cujo intuito é conhecer, de maneira imparcial, o estado-da-arte de um determinado assunto. Para que essa análise seja realmente executada de modo imparcial, os critérios de busca, de inclusão e de exclusão de trabalhos são definidos antes mesmo do início das buscas, possibilitando e incentivando a repetição da pesquisa segundo os mesmos critérios [Kitchenham 2004]. A revisão possui as seguintes fases: planejamento, condução e avaliação dos trabalhos encontrados.

Este trabalho tem por objetivo demonstrar uma revisão sistemática realizada com o intuito de investigar a produção científica relacionada à utilização de POA na implementação de variabilidades de LPS.

O trabalho está organizado da seguinte maneira: o Capítulo 2 apresenta as etapas de planejamento e condução da revisão sistemática, bem como os números obtidos. O Capítulo 3 apresenta uma visão geral dos trabalhos encontrados e estudados. Por fim, o Capítulo 4 apresenta as conclusões da pesquisa.

Capítulo 2 – Planejamento e Condução da Revisão

As revisões sistemáticas apresentam fases bem definidas de acordo com um protocolo previamente estruturado. Neste capítulo, serão tratadas as etapas de planejamento e condução. A primeira atividade da revisão é o planejamento, que consiste na definição das questões de pesquisa, metodologia para executar a revisão, fontes e estratégias de busca de artigos. A condução, por sua vez, trata-se da execução, consistindo da busca, análise e seleção de estudos primários de acordo com os critérios de inclusão e exclusão estabelecidos na etapa anterior.

Assim, neste capítulo serão apresentadas as atividades de planejamento e condução da revisão sistemática realizada.

2.1 Planejamento da Revisão

A atividade de planejamento da revisão sistemática foi realizada seguindo o modelo de protocolo apresentado por Biolchini et al. (2005) e Kitchenham et al. (2004). O plano foi revisado por um especialista, e as sugestões de ajuste foram discutidas com os revisores e implementadas no plano final.

Nesta seção, são apresentados os principais pontos definidos no planejamento da revisão apresentada.

2.1.1 Objetivos

- Identificar e analisar técnicas para a implementação de variabilidades de LPS utilizando POA.
- Observar estudos experimentais nos quais houve a utilização de aspectos para a implementação ou evolução de uma LPS.

2.1.2 Questões de Pesquisa

Tendo em vista os objetivos apresentados anteriormente, foi definida uma questão principal de pesquisa, bem como três questões secundárias, cada uma delas

com seus próprios critérios de inclusão e exclusão dos trabalhos. As questões definidas foram:

Questão Primária: Quais técnicas têm sido investigadas e utilizadas para implementar variabilidades de uma LPS usando aspectos?

Questão Secundária 1: Em quais situações a utilização de POA tem se mostrado favorável à implementação de variabilidades de LPS?

Questão Secundária 2: Quais tipos de problemas/obstáculos têm sido enfrentados para usar POA na implementação de LPS?

Questão Secundária 3: Quais tipos de estudos experimentais têm relatado a implementação de variabilidades de LPS usando aspectos?

2.1.2.1 Itens relacionados ao escopo ou especificidade das questões

Intervenção: Utilização de aspectos para a implementação de variabilidades de LPS.

Controle: Leitura e análise de artigos e trabalhos científicos.

População: Pesquisadores e desenvolvedores de software que utilizam recursos da programação orientada a aspectos em LPS.

Resultados: Técnicas e situações favoráveis à utilização de aspectos para evolução de LPS.

Aplicação: projetos de desenvolvimento de software que envolvam o uso de POA para evolução de LPS.

2.1.3 Estratégia de busca para seleção de estudos primários

Para a busca dos estudos primários, foi elaborada uma estratégia composta por três elementos: fontes, idioma dos trabalhos e palavras-chave. Desta forma, serão buscados estudos em determinados locais (fontes), escritos em determinado(s) idioma(s), encontrados por determinadas palavras-chave bem definidas.

Fontes: bases de dados eletrônicas amplamente indexadas (IEEE e ACM), busca em máquinas de busca eletrônica (Google e Scirus) e consultas a especialistas.

Idioma dos Trabalhos: Inglês, por ser uma língua universalmente aceita para trabalhos científicos; Português, de modo a obter publicações nacionais sobre o assunto e pelo fato de ser a língua nativa dos revisores.

Palavras-Chave: “*product line*” e “*aspects-oriented program*” (em inglês) e “linha de produto” e “Programação Orientada a Aspectos” (em português), com os seguintes termos e variações relacionados:

- **product line:** *product-line, product family, product-family, family of products.*
- **aspect-oriented program:** *aspects programming, AOP, aspect-oriented software development, aspect oriented software development, AOSD.*
- **linha de produto:** família de produto.
- **Programação Orientada a Aspectos:** Orientação a aspectos, POA, Desenvolvimento de Software Orientado a Aspectos, DSOA.

2.1.4 Critérios e Procedimentos para Seleção de Estudos

Aqui são apresentados os critérios de inclusão e seleção, bem como os processos de seleção preliminar e final dos estudos.

2.1.4.1 Critérios de Inclusão

Os critérios para a inclusão de estudos primários definidos para cada uma das questões de pesquisa são:

Questão Primária: Técnicas para implementar variabilidades de uma LPS usando aspectos.

Questão Secundária 1: Situações em que a utilização de POA tem se mostrado favorável à implementação de variabilidades de LPS.

Questão Secundária 2: Problemas e/ou dificuldades enfrentados para usar POA na implementação de variabilidades de uma LPS.

Questão Secundária 3: Estudos experimentais que relatam a implementação de variabilidades de LPS utilizando aspectos.

2.1.4.2 Critérios de Exclusão

Os critérios para a exclusão de estudos primários definidos para cada uma das questões de pesquisa são:

Questão Primária: Técnicas para a implementação de variabilidades de LPS que não utilizam Orientação a Aspectos.

Questão Secundária 1: Situações em que POA não tenha sido utilizada para a implementação de variabilidades de LPS.

Questão Secundária 2: Problemas/dificuldades do uso de POA em LPS que não estejam relacionados à implementação de variabilidades .

Questão Secundária 3: Estudos experimentais nos quais POA não foi usada para a implementação de variabilidades de LPS.

2.1.4.3 Processo de Seleção Preliminar

Para o processo de seleção preliminar serão elaboradas algumas *strings* de busca através das palavras-chaves e sinônimos identificados. Tais *strings* serão submetidas às máquinas de busca relatadas anteriormente.

Será efetuada a leitura dos resumos dos trabalhos encontrados. Caso os trabalhos sejam considerados relevantes, de acordo com os critérios anteriormente estabelecidos, serão separados para posteriormente serem estudados de forma detalhada.

2.1.4.4 Processo de Seleção Final

No processo de seleção final os trabalhos selecionados na etapa preliminar serão lidos e qualificados pelos revisores, responsáveis por elaborar um resumo contendo os principais dados e abordagens daqueles trabalhos.

2.1.5 Extração dos Resultados

Após a leitura dos artigos e trabalhos selecionados, será elaborado um relatório contendo as principais conclusões e análises dos revisores. Desse modo, os trabalhos considerados adequados à proposta da revisão serão separados e utilizados em um

projeto de pesquisa que visa a comparação da implementação de duas versões de uma LPS para Gerenciamento de Bilhetes Eletrônicos de Transporte (LPS-BET) (Donegan, 2008). A diferença entre elas está na forma de implementação das variabilidades. Uma delas usa componentes de software e a outra usará aspectos.

O relatório poderá conter tanto sínteses discursivas quanto tabuladas, de forma a comparar as diferentes técnicas aplicadas no contexto de aplicação de POA para implementação de variabilidades de LPS.

2.2 Condução da Revisão

A revisão sistemática foi conduzida por um período de três meses, sendo realizada de Agosto/2008 a Outubro/2008. Foram encontrados 194 trabalhos, que foram submetidos às etapas de seleção preliminar, seleção final e extração dos resultados. As seções seguintes apresentam as atividades realizadas na condução da revisão sistemática, envolvendo estratégias utilizadas na construção das *strings* de busca e os resultados da busca em cada uma das fontes definidas.

A condução da revisão apresentou as seguintes etapas:

- Definição das *strings* de busca;
- Aplicação das *strings* de busca e obtenção de trabalhos nas fontes definidas;
- Seleção preliminar dos trabalhos;
- Seleção final dos trabalhos.

Observa-se ainda que nas etapas de seleção preliminar e final, foram realizadas reuniões entre os revisores para decidir sobre a inclusão ou exclusão dos trabalhos.

As próximas seções demonstram cada uma dessas etapas detalhadamente.

2.2.1 Definição das *Strings* de Busca

A definição das *strings* de busca foi realizada com a combinação dos sinônimos das palavras-chave identificadas na seção 2.1.3. Foram elaboradas *strings* para os dois idiomas da pesquisa, português e inglês. Para a definição, foram utilizados os operadores lógicos “e” (AND) e “ou” (OR), conforme demonstrado a seguir:

Português:

(linha de produtos OR família de produtos)

AND

(Programação Orientada a Aspectos OR Orientação a Aspectos OR POA OR Desenvolvimento de Software Orientado a Aspectos OR DSOA)

Inglês:

(product line OR product-line OR product family OR product-family OR family of products)

AND

(aspect-oriented program OR aspects programming OR AOP OR aspect-oriented software development OR aspect oriented software development OR AOSD)

2.2.2 Buscas Realizadas e Obtenção dos Trabalhos

A seguir são demonstradas as buscas realizadas por meio da aplicação das *strings* customizadas para cada fonte. Ao final desta seção, os números obtidos são ilustrados por meio de gráficos.

Busca na IEEE (*Institute of Electrical and Electronics Engineers*)

A busca na IEEE foi realizada em setembro de 2008, por meio da aplicação das *strings* de busca citadas na seção anterior e adaptadas ao mecanismo de busca avançada da IEEE.

A busca avançada da IEEE está disponível no endereço eletrônico <http://ieeexplore.ieee.org/search/advsearch.jsp> em que é possível fazer dois tipos de busca: por frases e palavras-chave (opção 1); e por palavras-chave, frases e expressões booleanas (operadores AND e OR) (opção 2). Além disso, é possível definir o ano e onde a busca será realizada como, por exemplo, periódicos, anais de eventos, normas IEEE (standards), e livros.

Como a *string* de busca já estava definida, optou-se pelo uso de expressão booleana (opção 2). A seguir estão demonstradas as duas *strings* de busca aplicadas, devido à questão do idioma.

Português:

```
((("linha de produto de software" OR "linha de produtos de software" OR "família de produtos de software") AND ("Programação Orientada a Aspectos" OR "Orientação a aspectos" OR "POA" OR "Desenvolvimento de Software Orientado a Aspectos" OR "DSOA"))<in>(metadata))
```

Inglês:

```
((("software product line" OR "software product-line" OR "software product family" OR "software product-family" OR "family of software products") AND ("aspect-oriented programming" OR "AOP" OR "aspect-oriented software development" OR "aspect oriented software development" OR "AOSD"))<in>(metadata))
```

A *string* foi aplicada juntamente com o parâmetro *metadata*, que indica a busca em todos os campos possíveis dos trabalhos publicados. As demais opções do mecanismo de busca ficaram preenchidas como padrão. Foram obtidos 09 trabalhos, todos disponíveis em formato PDF, representando 4,64% dos trabalhos encontrados.

Busca na ACM (Association for Computer Machinery)

A busca na ACM foi realizada em dois momentos: a submissão da *string* em português, em setembro de 2008; a submissão da *string* em inglês, em outubro de 2008. Ambas as *strings* de busca foram citadas na seção 2.2.1 e adaptadas ao mecanismo de busca avançada da ACM.

A busca avançada da ACM está disponível no endereço eletrônico <http://portal.acm.org/advsearch.cfm> em que é possível fazer a busca na biblioteca digital (The ACM Digital Library) e no guia de referências (The Guide). A biblioteca digital disponibiliza os trabalhos na íntegra no formato PDF, enquanto o guia serve somente como uma referência de trabalhos, incluindo os da biblioteca digital, em outras fontes sem disponibilizá-los. Nesta revisão a *string* foi submetida apenas para a primeira opção.

A seguir são apresentadas as *strings* de busca customizada para o mecanismo de busca da ACM.

Português:

```
((("+abstract:" linha de produtos de software ""+" orientação a aspectos "")) ((("+abstract:" linha de produtos de software "" +"" POA "")))))  
  
(((("+abstract:" linha de produtos de software "" +"" Desenvolvimento de Software orientado a Aspectos "")))))  
  
(((("+abstract:" linha de produtos de software "" +"" DSOA "")))))
```

Inglês:

```
((("+abstract:" software product line ""+" aspects-oriented programming "")))))  
  
(((("+abstract:" software product line "" +"" AOP "")))))  
  
(((("+abstract:" software product line ""+" aspects oriented software development "")))))  
  
(((("+abstract:" software product line "" +"" AOSD "")))))
```

Como pode ser observado, a busca foi realizada no resumo (*abstract*) dos trabalhos. As demais opções ficaram preenchidas como padrão do mecanismo de busca.

Foram obtidos 30 trabalhos, todos eles em formato PDF, representando 15,46% dos trabalhos encontrados.

Busca na Scirus (Elsevier)

A busca na Scirus foi realizada em dois momentos: a submissão da string em português, em agosto de 2008; a submissão da *string* em inglês, em outubro de 2008. Ambas as *strings* de busca foram citadas na seção 2.2.1 e adaptadas ao mecanismo de busca avançada da Scirus.

A busca avançada da Scirus está disponível no endereço eletrônico <http://www.scirus.com/srsapp/advanced/index.jsp> em que é possível fazer a busca de trabalhos disponíveis na íntegra. Além disso, é possível definir ainda em que local do trabalho a *string* será avaliada como, por exemplo, o ano de publicação e o formato dos trabalhos (PDF, HTML, DOC ou formato livre). Um detalhe particular acerca do mecanismo de busca é que ele apresenta dois campos, onde são inseridos os principais termos da pesquisa e seus sinônimos. O sistema, então, realiza a busca por trabalhos que estejam relacionados às duas classes de termos inseridas.

A seguir são apresentadas as *strings* de busca customizada para o mecanismo de busca da Scirus.

Português:

Campo 1: *“linha de produto de software” OR “linha de produtos de software” OR “família de produtos de software”*

Campo 2: *“Programação Orientada a Aspectos” OR “Orientação a aspectos” OR POA OR “Desenvolvimento de Software Orientado a Aspectos” OR DSOA*

Inglês:

Campo 1: *“software product line” OR “software product-line” OR “software product family” OR “software product-family” OR “family of software products”*

Campo 2: *“aspect-oriented programming” OR AOP OR “aspect-oriented software development” OR “aspect oriented software development” OR AOSD*

Conforme pode ser observado, os dois campos foram assim preenchidos: para o primeiro foram relacionados os sinônimos de “linha de produto”, enquanto para o segundo os sinônimos de “programação orientada a aspectos”. Além disso, foram preenchidas algumas opções a mais como seguem:

- *Information Types: articles, abstracts e thesis and presentations.*
- *File Formats: pdf e doc.*

Foram obtidos 48 trabalhos, todos eles em formato PDF, representando 24,74% dos trabalhos encontrados.

Busca no Google

A busca no Google foi realizada entre agosto e outubro de 2008, por meio da aplicação das *strings* de busca citadas na seção 2.2.1 e adaptadas ao mecanismo de busca avançada do Google.

A busca avançada do Google está disponível no endereço eletrônico http://www.google.com/advanced_search?hl=en em que é possível fazer a busca de trabalhos disponíveis na íntegra. Além disso, é possível definir ainda em que local do trabalho a *string* será avaliada como, por exemplo, o título ou a URL dos trabalhos.

A seguir, são apresentadas as *strings* de busca adaptadas para o mecanismo de busca avançada do Google.

Português:

“linha de produto de software” OR “linha de produtos de software” OR “família de produtos de software” + “Programação Orientada a Aspectos” OR “Orientação a aspectos” OR POA OR “Desenvolvimento de Software Orientado a Aspectos” OR DSOA

Inglês:

“software product line” OR “software product-line” OR “software product family” OR “software product-family” OR “family of software products” + “aspect-oriented program” OR “aspects programming” OR AOP OR “aspect-oriented software development” OR “aspect oriented software development” OR AOSD

Foram obtidos 104 trabalhos, dos quais 103 estão em formato PDF e 01 está em formato DOC, representando 53,61% dos trabalhos encontrados.

Consulta a Especialistas

Para auxiliar na busca por trabalhos relevantes em relação ao tema da revisão sistemática, foram consultados alguns especialistas. Tais consultas foram realizadas durante o mês de junho de 2009 e seguidas pelo estudo dos trabalhos indicados, encerrando, assim, a revisão.

Paulo Cesar Masiero (masiero@icmc.usp.br)

Paulo Cesar Masiero é professor titular do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP), em São Carlos-SP. Dentre seus interesses de pesquisa estão Linha de Produto de Software e Programação Orientada a Aspectos. Assim, suas indicações resultaram em um total de 04 trabalhos, listados a seguir.

1. ***Guidelines for Using Aspects to Evolve Product Lines***, (Pacios, Masiero e Braga, 2006), disponível em: <http://www.lac.inf.puc-rio.br/~viterbo/SBES-2006/wasp-anais.pdf>;
2. ***Uma Abordagem Orientada a Aspectos para o Desenvolvimento de Linhas de Produto de Software*** (Pacios, 2006), disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08052007-171603/>;
3. ***AIPLE-IS: An Approach to Develop Product Lines for Information Systems Using Aspects*** (Braga et al., 2007), disponível em: <http://www.ic.unicamp.br/sbcars2007/tecnicas/files/sbcars2007-braga-aiple.pdf>.
4. ***GenArch: A Model-Based Product Derivation Tool*** (Cirilo, Kulesza e Lucena 2007), disponível em: <http://www.ic.unicamp.br/sbcars2007/tecnicas/sbcars2007-tools-cirilo-genarch.pdf>.

Márcio de Medeiros Ribeiro (mmr3@cin.ufpe.br)

Márcio de Medeiros Ribeiro é doutorando em Ciência da Computação pela Universidade Federal de Pernambuco (UFPE), em Recife-Pernambuco. Dentre seus interesses de pesquisa estão: Linguagens Específicas de Domínio (DSLs), Programação Orientada a Objetos e Aspectos, Linhas de Produtos de Software e Modularidade de Software. Assim, suas indicações resultaram em um total de 05 trabalhos, além dos já encontrados por meio das buscas nas máquinas eletrônicas, listados a seguir.

1. ***Feature Relation and Dependency Management: An Aspect-Oriented Approach*** (Cho, Lee e Kang, 2008), disponível em: <http://portal.acm.org/citation.cfm?id=1440006>.
2. ***An evaluation of aspect-oriented programming as a product line implementation technology*** (Anastasopoulos e Muthig, 2004), disponível em: <http://www.springerlink.com/content/nhukhh9jx4jver62>.

3. ***When to Use Features and Aspects? A Case Study*** (Apel e Battory, 2006), disponível em: <http://portal.acm.org/citation.cfm?id=1173706.1173716>.
4. ***Variability Management with Feature-Oriented Programming and Aspects*** (Mezini e Ostermann, 2004), disponível em: <http://portal.acm.org/citation.cfm?id=1029915>.
5. ***Separating Features in Source Code: An Exploratory Study*** (Murphy et al., 2001), disponível em: pages.cpsc.ualgary.ca/~rwalker/publications/murphy2001b.pdf.

Alessandro F. Garcia (afgarcia@inf.puc-rio.br)

Alessandro Fabrício Garcia é Professor Assistente do Departamento de Informática da Pontifícia Universidade Católica (PUC) do Rio de Janeiro. Dentre seus interesses de pesquisa estão: desenvolvimento de software orientado a aspectos, sistemas multi-agentes, tratamento de exceções, métricas e arquitetura de software. Suas indicações resultaram em um total de 02 trabalhos, além dos já encontrados por meio das buscas nas máquinas eletrônicas, listados a seguir.

1. ***Overview of Extensions/Improvements to Existing Implementation Technologies*** (Gasiunas et al., 2007), disponível em: <http://ample.holos.pt/pageview.aspx?pageid=68&langid=1>.
2. ***Survey of existing implementation techniques with respect to their support for product lines*** (Pohl et al., 2007), disponível em: <http://ample.holos.pt/pageview.aspx?pageid=68&langid=1>.

Alguns números obtidos na busca

Ao final das buscas por estudos nas fontes anteriormente citadas, é possível ilustrar os números obtidos por meio de gráficos. Os gráficos têm por objetivo demonstrar todos os dados extraídos levando-se em consideração:

- Quantidade de trabalhos encontrados, segundo a classificação por fonte;
- Quantidade de trabalhos selecionados, segundo seleção preliminar;
- Quantidade de trabalhos incluídos, após leitura na íntegra.

Com a realização desta revisão foram encontrados 202 estudos, separados por fonte da seguinte forma: IEEE, 9 estudos; ACM, 30 estudos; Scirus, 48 estudos; Google, 104 estudos; consulta a especialistas, 11. A Figura 1 ilustra a participação de cada fonte em relação ao número total de trabalhos encontrados.

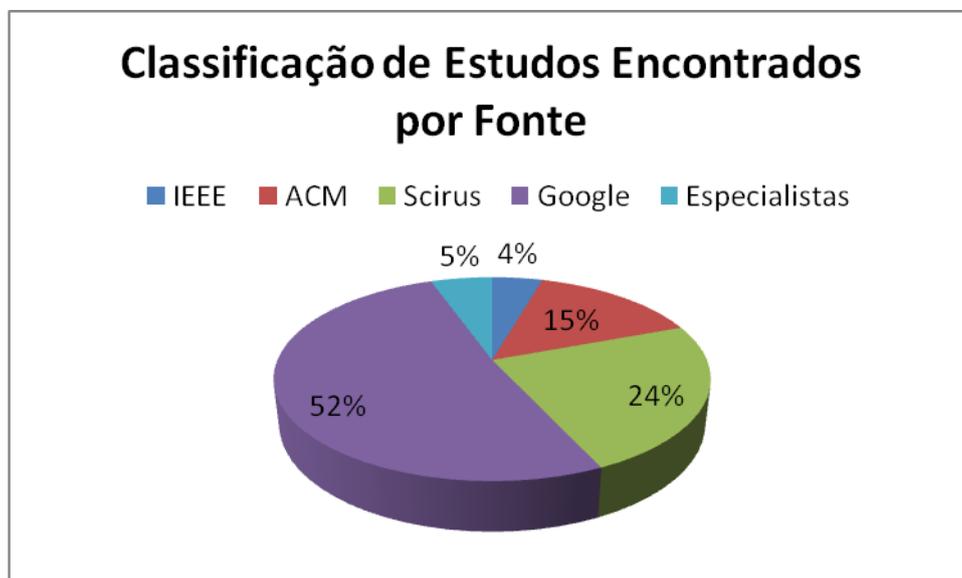


Figura 1: Percentual de estudos encontrados classificados por fonte.

2.2.3 Seleção Preliminar dos Trabalhos

Nesta etapa foram lidos os resumos (*abstracts*) de todos os trabalhos encontrados na etapa anterior conforme o planejamento apresentado na seção 2.1. Como resultado desta seleção foram obtidos: 115 estudos excluídos e 87 selecionados para a leitura na íntegra. A Figura 2 demonstra as porcentagens de trabalhos excluídos e incluídos em relação ao total de estudos encontrados.

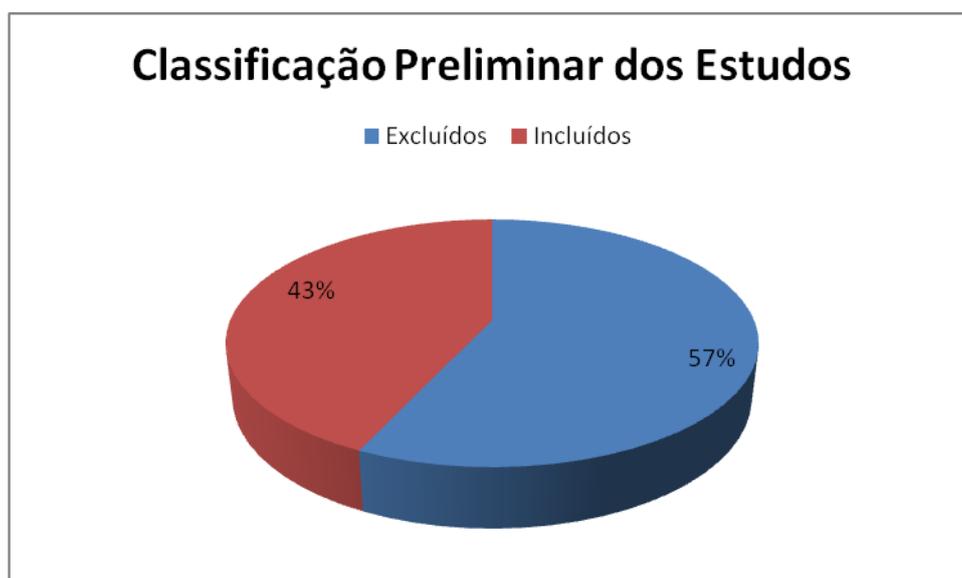


Figura 2: Resultado da Seleção Preliminar.

Dessa forma, os itens a seguir apresentam os 87 trabalhos selecionados para leitura na íntegra, representando 43,07% do total de trabalhos encontrados.

IEEE

Dos 09 trabalhos encontrados na IEEE, 07 foram selecionados para a leitura na íntegra, os quais são listados a seguir:

1. *A Case Study Implementing Features Using AspectJ* (Kastner, Apel e Batory, 2007);
2. *Aspect Refinement and Bounding Quantification in Incremental Designs* (Apel, Leich e Saake, 2005b);
3. *Aspect-Oriented Requirements Engineering for Software Product Lines* (Kuloor e Eberlein, 2003);
4. *Combining Feature-Oriented Analysis and Aspect-Oriented Programming for Product Line Asset Development* (Lee et al., 2006);
5. *Product Line Implementation Using Aspect-Oriented and Model-Driven Software Development* (Voelter e Groher, 2007a);
6. *Representation of Variability in Software Product Line Using Aspect-Oriented Programming* (Heo e Choi, 2006);
7. *Aspectual Feature Modules* (Apel, Leich e Saake, 2008).

ACM - Digital Library

Dos 30 trabalhos encontrados na ACM – Digital Library, somente 14 foram selecionados para leitura na íntegra, os quais estão listados a seguir:

1. *A Generic Weaver for Supporting Product Lines* (Morin et al., 2008);
2. *A Reflective Aspect-Oriented Model Editor Based on Metamodel Extension* (Ubayashi, Sano e Otsubo, 2007);
3. *An Integrated Aspect-Oriented Model-Driven Software Product Line Tool Suite* (Groher, Schwanninger e Voelter, 2008);

4. *Analysis of Crosscutting Features in Software Product Lines* (Conejero e Hernández, 2008);
5. *Aspectual Support for Specifying Requirements in Software Product Lines* (Siy et al., 2007);
6. *ATOLL: Aspect-Oriented Toll System* (Navarro et al., 2007);
7. *Decision Model for Implementing Product Lines Variabilities* (Ribeiro, Matos Jr e Borba, 2008);
8. *Edicts: Implementing Features with Flexible Binding Times* (Chavrarvarthy, Regehr e Eide, 2008);
9. *Evolving Software Product Lines with Aspects: an Empirical Study on Design Stability* (Figueiredo et al., 2008);
10. *From Scenarios to Aspects: Exploring Product Lines* (Krüger, Mathew e Meisinger, 2005);
11. *Implementing Product Line Variabilities* (Anastasopoulous e Gracek, 2001);
12. *Separation of Concerns in Software Product Line Engineering* (Saleh e Gomma, 2005);
13. *Using Design Structure Matrices to Assess Modularity in Aspect-Oriented Software Product Lines* (Matos Jr et al., 2007);
14. *Xweave: Models and Aspects in Concert* (Groher e Voelter, 2007).

Scirus (Elsevier)

Dos 48 trabalhos encontrados na Scirus, 14 foram selecionados para leitura na íntegra, os quais são listados a seguir:

1. *Abordagem Aspecting: Migração de Sistemas OO para Sistemas AO* (Ramos, 2004);
2. *Are Aspects Useful for Managing Variability in Software Product Lines? A Case Study* (Nyßen, Tyszberowicz e Weiler, 2005);
3. *Ferramenta para Construção de Linha de Produtos no Eclipse* (Vasconcelos, 2005);

4. *Frameworks Transversais: Definições, Classificações, Arquitetura e Utilização em um Processo de Desenvolvimento de Software* (Camargo, 2006);
5. *Handling Variants In A Product Family* (Zhang, 2001);
6. *Implementing Software Product Line Adoption Strategies* (Alves, 2007);
7. *Modular Integration through Aspects: Making Cents of Legacy Systems* (Gibbs et al., 2007);
8. *On the Separation of Concerns in Program Families* (Colyer, Hashid e Blair, 2004);
9. *Reducing the Complexity of AspectJ Mechanisms for Recurring Extensions* (Kuhlemann e Kästner, 2007);
10. *Refactorings to Evolve Object-Oriented Systems with Aspect-Oriented Concepts* (Monteiro, 2005);
11. *The Role of Aspects in Domain Engineering* (Siy, Zand e Winter, 2005);
12. *The Role of Aspects in Modeling Product Line Variabilities* (Liu, Lutz e Rajan, 2006);
13. *Towards a Standard Design Language for AOSD* (Clarke e Walker, 2002);
14. *Using AspectJ to Build a Software Product Line for Mobile Devices* (Young, 2005).

Google

Dos 104 trabalhos encontrados no Google, 42 foram selecionados para leitura na íntegra, os quais são listados a seguir:

1. *A Feature Model of an Aspect-Oriented Middleware Family for Pervasive Systems* (Fuentes e Gamez, 2008);
2. *A Variant Metamodel Language to Support Software Product Lines* (Guedes, 2008);
3. *An Aspect-Oriented Approach to Framework Development* (Kulesza e Lucena, 2006);

4. *An Enhanced Framework for Validation of Aspectual Requirements* (Khan, Ali e Jaffar-ur-Rehman, 2005);
5. *Analyzing Aspects in Production Plans for Software Product Lines* (Noordhuizen, 2006);
6. *Applying Aspect-Oriented Techniques in Automotive Software Product-Line Engineering* (Mehner, Reiser e Weber, 2006);
7. *Aspect Refinement in Software Product Lines* (Apel, Leich e Saake, 2005a);
8. *Aspect-Oriented Use Case Modeling for Software Product Lines* (Anthonysamy e Somé, 2008);
9. *Aspects and Constraints for Implementing Configurable Product-Line Architectures* (Lesaint e Papamargaritis, 2004);
10. *Assessment of Product Line Architecture and Aspect Oriented Software Architecture Methods* (Verdin e Olalde, 2006);
11. *Avoiding Incorrect and Unpredictable Behaviour with Attribute-Based Crosscutting* (Lafferty, 2005);
12. *Can Aspects Model Product Lines?* (Oldevik, 2008);
13. *Combining Components, Aspects, Domain Specific Languages and Product Lines for Ambient Intelligent Application Development* (Fuentes e Jiménez, 2006);
14. *Concern Hierarchies* (Spinczyk, Lohmann, and Schröder-Preikschat, 2006);
15. *Customizing Aspect-Oriented Variabilities Using Generative Techniques* (Kulesza et al., 2008);
16. *Declaring Static Crosscutting Dependencies in AspectJ* (Neto, Alves e Borba, 2008);
17. *Designing Features as Pluggable Collaborations* (Groher, Bleicher e Schwanninger, 2005);
18. *Developing and Evolving Multi-Agent System Product Lines* (Nunes et al., 2008);
19. *Extensão para UML Destinada à Modelagem de Variabilidade Transversal em Componentes através da Orientação a Aspectos* (Winck, 2005);
20. *Finegrain Application Specific Customization for Embedded Software* (Beuche, Spinczyk e Schröder-Preikschat, 2002);

21. *Generative Aspect-Oriented Component Adaptation* (Liu, Feng e Kerridge, 2008);
22. *Handling Variability in Model Transformations and Generators* (Voelter e Groher, 2007b);
23. *Implementing Framework Crosscutting Extensions with EJP's and AspectJ* (Kulesza et al., 2008);
24. *Implementing Product-Line Features by Composing Component Aspects* (Griss, 2000);
25. *Instantiating and Customizing Product Lines Architectures Using Aspects and Crosscutting Feature Models* (Kulesza et al., 2005);
26. *Ligo: Uma Linha de Produtos de Software para Gerenciamento de uma Igreja Cristã* (Teixeira, 2007);
27. *Modeling and Aspect Weaving* (Jézéquel, 2006);
28. *Modeling Software Product Lines with AOURN* (Mussbacher et al., 2008);
29. *Multi-Step Concern Refinement* (Yie et al., 2008);
30. *On the Modularity of Aspect-Oriented and Other Techniques for Implementing Product Lines Variabilities* (Ribeiro, et al., 2007);
31. *Product Line Evolution and AOP* (Burgstaller, 2006);
32. *Supporting Product Line Evolution with Framed Aspects* (Loughran, 2004);
33. *Técnicas de Construção de Linha de Produto para Jogos Móveis* (Carmo, 2005);
34. *Towards a Model for Specifying and Composing Concerns in Software Product Line Engineering* (Kuttruff, 2008);
35. *Tracing Variability from Implementation to Test Using Aspect-Oriented Programming* (Knauber e Schneider, 2004);
36. *Um Processo de Reestruturação de Código Baseado em Aspectos* (Ramos, Penteado e Masiero, 2004);
37. *Uma Abordagem Orientada a Aspectos para o Desenvolvimento de Framework* (Kulesza, 2007);

38. *Understanding Feature Modularity in Feature Oriented Programming and its Implications to Aspect Oriented Programming* (Lopez-Herreron, 2005);
39. *Using AspectJ to Implement Product Lines: A Case Study* (Herrejon e Batory, 2002a);
40. *Using Goal-Models to Analyze Variability* (González-Baixauli, Leite e Laguna, 2007);
41. *Using Hyper/J to Implement Product Lines: A Case Study* (Herrejon e Batory, 2002b);
42. *Variações em Linhas de Produto de Software Usando Programação Orientada a Aspectos* (Borba, Alves e Matos Júnior, 2004).

Paulo César Masiero

Dos 04 trabalhos indicados pelo especialista, 03 foram selecionados para leitura na íntegra. Os trabalhos estão listados a seguir:

1. *AIPLE-IS: An Approach to Develop Product Lines for Information Systems Using Aspects* (Braga et al., 2007);
2. *Guidelines for Using Aspects to Evolve Product Lines* (Pacios, Masiero e Braga, 2006);
3. *Uma Abordagem Orientada a Aspectos para Desenvolvimento de Linhas de Produtos de Software* (Pacios, 2006).

Márcio de Medeiros Ribeiro

Os 05 trabalhos indicados pelo especialista foram selecionados para leitura na íntegra. Os trabalhos estão listados a seguir:

1. *An evaluation of aspect-oriented programming as a product line implementation technology* (Anastasopoulos e Muthig, 2004);
2. *Feature Relation and Dependency Management: An Aspect-Oriented Approach* (Cho, Lee e Kang, 2008);
3. *Separating Features in Source Code: An Exploratory Study* (Murphy et al., 2001).

4. *Variability Management with Feature-Oriented Programming and Aspects* (Mezini e Ostermann, 2004);
5. *When to Use Features and Aspects? A Case Study* (Apel e Battory, 2006);

Alessandro F. Garcia

Os 02 trabalhos indicados pelo especialista foram selecionados para leitura na íntegra. Os trabalhos estão listados a seguir:

1. *Overview of Extensions/Improvements to Existing Implementation Technologies* (Gasiunas et al., 2007);
2. *Survey of existing implementation techniques with respect to their support for product lines* (Pohl et al., 2007).

2.2.4 Seleção Final dos Trabalhos

Após a leitura dos 87 trabalhos selecionados na etapa anterior, foi avaliada a qualidade efetiva de cada um deles em relação às questões de estudo. A Figura 3 mostra a participação desses trabalhos distribuídos por fonte. Foram selecionados: 10 estudos referentes à consulta especialistas; 7 estudos encontrados na IEEE; 14 estudos encontrados na ACM; 14 estudos encontrados na Scirus; 42 estudos encontrados no Google.

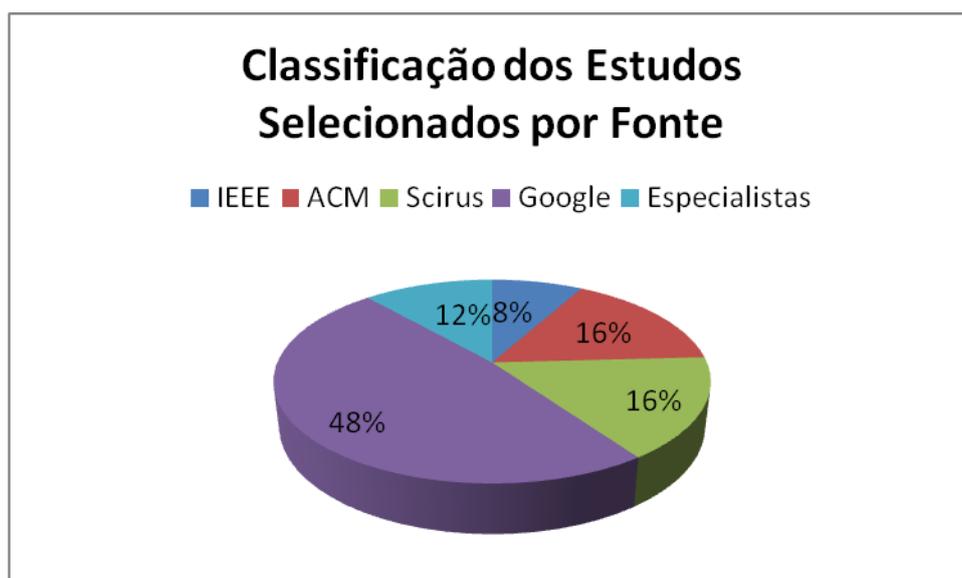


Figura 3: Estudos selecionados classificados por fonte.

Assim, dos 87 trabalhos lidos na íntegra, 46 foram incluídos e 41 foram excluídos por não estarem de acordo com as expectativas e os objetivos do estudo em questão. Daqueles incluídos, observa-se que: 3 foram obtidos em consulta ao especialista Paulo César Masiero; 2 foram obtidos na IEEE; 5 foram obtidos na ACM; 7 foram obtidos na Scirus; 23 foram obtidos no Google. A Figura 4 ilustra a participação de cada fonte no total de trabalhos incluídos.

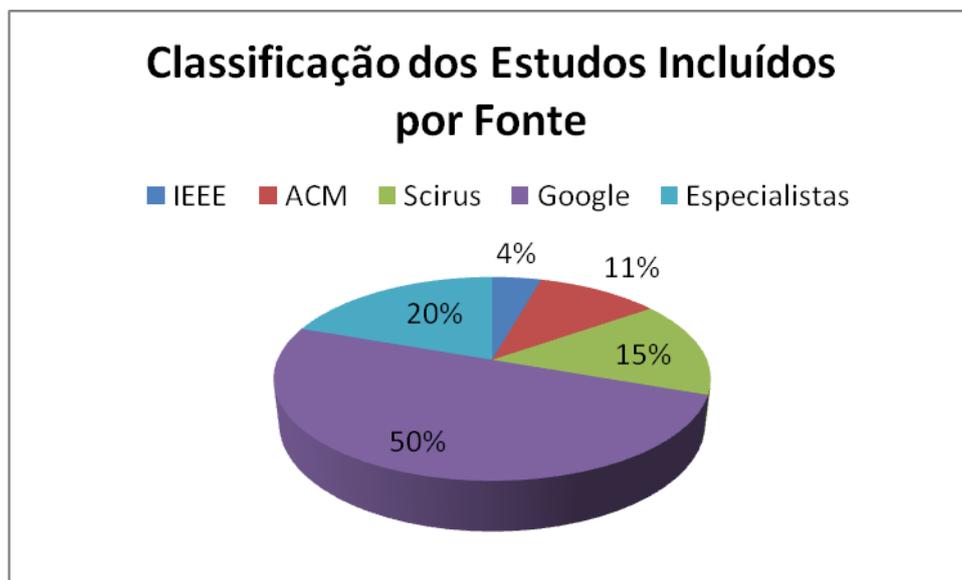


Figura 4: Trabalhos incluídos classificados por fonte.

Em relação a todos os trabalhos encontrados, esta revisão obteve por resultado final: 46 estudos incluídos e 156 excluídos em relação ao tema pesquisado, cujos percentuais são apresentados na Figura 5.



Figura 5: Classificação Final dos Estudos.

Os estudos obtidos com a classificação final foram analisados de acordo com dois grandes grupos: Técnicas e propostas para implementação de LPS utilizando POA e Estudos de Caso (esta classificação será mais bem definida no Capítulo 3). A Figura 6 ilustra esta classificação, sendo que dos 46 trabalhos incluídos, 32 estão identificados como técnicas e abordagens e 14 como estudos de caso.

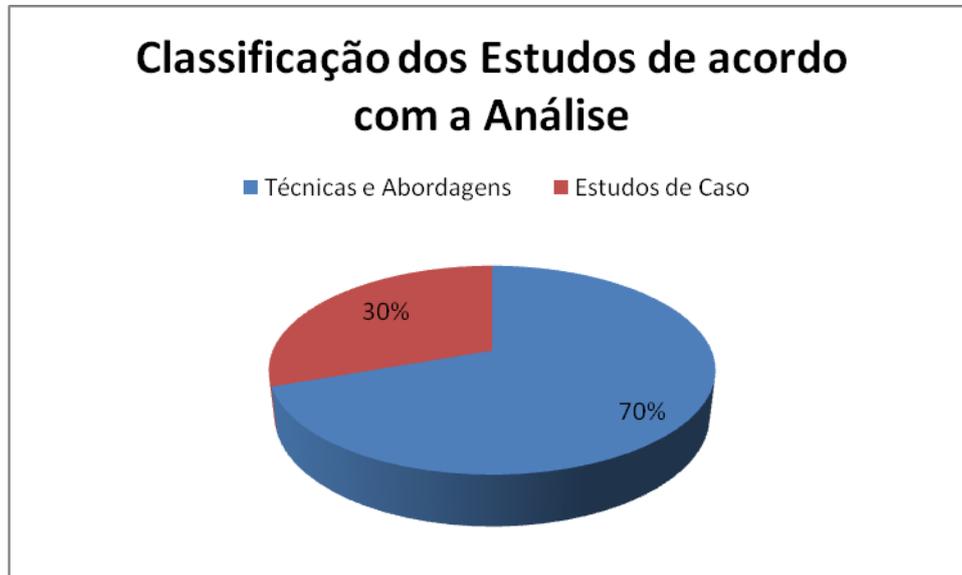


Figura 6: Classificação dos Estudos Incluídos de acordo com os critérios de análise.

Os 46 trabalhos incluídos nessa revisão sistemática são relacionados a seguir. No próximo capítulo, apresenta-se uma análise dos resultados obtidos após a leitura completa desses estudos.

1. *Combining Feature-Oriented Analysis and Aspect-Oriented Programming for Product Line Asset Development* (Lee et al., 2006);
2. *Analysis of Crosscutting Features in Software Product Lines* (Conejero e Hernández, 2008);
3. *When to Use Features and Aspects? A Case Study* (Apel e Battory, 2006);
4. *Variability Management with Feature-Oriented Programming and Aspects* (Mezini e Ostermann, 2004);
5. *ATOLL: Aspect-Oriented Toll System* (Navarro et al., 2007);
6. *Product Line Evolution and AOP* (Burgstaller, 2006);
7. *An Aspect-Oriented Approach to Framework Development* (Kulesza e Lucena, 2006);

8. *Feature Relation and Dependency Management: An Aspect-Oriented Approach* (Cho, Lee e Kang, 2008);
9. *Decision Model for Implementing Product Lines Variabilities* (Ribeiro, Matos Jr e Borba, 2008);
10. *Edicts: Implementing Features with Flexible Binding Times* (Chavravarthy, Regehr e Eide, 2008);
11. *Implementing Framework Crosscutting Extensions with EJP's and AspectJ* (Kulesza et al., 2008);
12. *On the Modularity of Aspect-Oriented and Other Techniques for Implementing Product Lines Variabilities* (Ribeiro, et al., 2007);
13. *Implementing Product-Line Features by Composing Component Aspects* (Griss, 2000);
14. *Implementing Software Product Line Adoption Strategies* (Alves, 2007);
15. *Applying Aspect-Oriented Techniques in Automotive Software Product-Line Engineering* (Mehner, Reiser e Weber, 2006);
16. *Survey of existing implementation techniques with respect to their support for product lines* (Pohl et al., 2007).
17. *Representation of Variability in Software Product Line Using Aspect-Oriented Programming* (Heo e Choi, 2006);
18. *Refactorings to Evolve Object-Oriented Systems with Aspect-Oriented Concepts* (Monteiro, 2005);
19. *On the Separation of Concerns in Program Families* (Colyer, Hashid e Blair, 2004);
20. *Concern Hierarchies* (Spinczyk, Lohmann, and Schröder-Preikschat, 2006);
21. *Instantiating and Customizing Product Lines Architectures Using Aspects and Crosscutting Feature Models* (Kulesza et al., 2005);
22. *Customizing Aspect-Oriented Variabilities Using Generative Techniques* (Kulesza et al., 2008);
23. *Product Line Implementation Using Aspect-Oriented and Model-Driven Software Development* (Voelter e Groher, 2007a);

24. *Multi-Step Concern Refinement* (Yie et al., 2008);
25. *Designing Features as Pluggable Collaborations* (Groher, Bleicher e Schwanninger, 2005);
26. *Overview of Extensions/Improvements to Existing Implementation Technologies* (Gasiunas et al., 2007);
27. *Tracing Variability from Implementation to Test Using Aspect-Oriented Programming* (Knauber e Schneider, 2004);
28. *Can Aspects Model Product Lines?* (Oldevik, 2008);
29. *Developing and Evolving Multi-Agent System Product Lines* (Nunes et al., 2008);
30. *Guidelines for Using Aspects to Evolve Product Lines* (Pacios, Masiero e Braga, 2006);
31. *Uma Abordagem Orientada a Aspectos para Desenvolvimento de Linhas de Produtos de Software* (Pacios, 2006).
32. *AIPLE-IS: An Approach to Develop Product Lines for Information Systems Using Aspects* (Braga et al., 2007);
33. *Variações em Linhas de Produto de Software Usando Programação Orientada a Aspectos* (Borba, Alves e Matos Júnior, 2004).
34. *Técnicas de Construção de Linha de Produto para Jogos Móveis* (Carmo, 2005);
35. *Evolving Software Product Lines with Aspects: an Empirical Study on Design Stability* (Figueiredo et al., 2008);
36. *Using AspectJ to Build a Software Product Line for Mobile Devices* (Young, 2005).
37. *Declaring Static Crosscutting Dependencies in AspectJ* (Neto, Alves e Borba, 2008);
38. *Analyzing Aspects in Production Plans for Software Product Lines* (Noordhuizen, 2006);
39. *An evaluation of aspect-oriented programming as a product line implementation technology* (Anastasopoulos e Muthig, 2004);
40. *Ligo: Uma Linha de Produtos de Software para Gerenciamento de uma Igreja Cristã* (Teixeira, 2007);

41. *Uma Abordagem Orientada a Aspectos para o Desenvolvimento de Framework* (Kulesza, 2007);
42. *The Role of Aspects in Modeling Product Line Variabilities* (Liu, Lutz e Rajan, 2006);
43. *Are Aspects Useful for Managing Variability in Software Product Lines? A Case Study* (Nyßen, Tyszberowicz e Weiler, 2005);
44. *Modular Integration through Aspects: Making Cents of Legacy Systems* (Gibbs et al., 2007);
45. *Using AspectJ to Implement Product Lines: A Case Study* (Herrejon e Batory, 2002a);
46. *Using Hyper/J to Implement Product Lines: A Case Study* (Herrejon e Batory, 2002b).

Capítulo 3 – Análise dos Resultados

Neste capítulo são apresentados os resultados obtidos de acordo com a condução da revisão sistemática, levando em consideração os objetivos e as questões de pesquisa propostas. É importante destacar que a revisão teve como objetivo identificar os trabalhos que envolvem a implementação de variabilidades de LPS utilizando POA.

A leitura na íntegra dos 46 trabalhos, ao final selecionados, permitiu analisar várias propostas, incluindo técnicas e abordagens, para aplicar POA na implementação de LPS. Além disso, foram encontrados alguns estudos de caso, onde vantagens e até mesmo desvantagens dessas propostas são discutidos.

As seções seguintes apresentam uma caracterização sobre POA e LPS, bem como um panorama a respeito da implementação do assunto em questão.

3.1 Caracterização do problema

De acordo com Kulesza (2007), uma LPS pode ser definida como um conjunto de sistemas de software que compartilham funcionalidades em comum e que satisfazem necessidades específicas de um determinado segmento do mercado. Além das funcionalidades em comum, existem as características (*features*) que diferenciam os sistemas entre si, as chamadas variabilidades.

Segundo Teixeira (2007), a variabilidade é o ponto chave de uma LPS, uma vez que a representação explícita desta torna possível a geração de produtos específicos de uma LPS. Um outro conceito importante é o de *concerns* ou interesses que, correspondem a requisitos, funcionais ou não, que devem estar presentes no sistema.

O problema apresentado diz respeito à implementação de LPS, com uma ênfase maior nos pontos de variação, isto é, nas variabilidades. Durante esta implementação, podem ser observados alguns fenômenos descritos em Nyßen, Tyszberowicz e Weiler (2005), a saber:

- Código emaranhado (*tangling code*): refere-se ao fato de que muitos interesses de uma dimensão são realizados por um interesse de outra dimensão;
- Código espalhado (*scattering code*): refere-se ao fato de que um interesse de uma dimensão é realizado por vários interesses de outra dimensão;

- Código entrelaçado (*crosscutting code*): refere-se ao fato de que um interesse de uma dimensão é realizado de forma redundante por vários interesses de uma outra dimensão.

De uma maneira superficial, os problemas estão relacionados à organização do código-fonte, que pode estar “emaranhado” em alguns pontos; alguns trechos podem estar “espalhados” por todo o artefato; e, o pior caso, onde as linhas podem estar “entrecortadas”, causando redundância e dificultando o entendimento e a legibilidade do código.

Para evitar quaisquer destes fenômenos, foram propostas e analisadas várias técnicas, tais como, Agregação, Herança, Compilação Condicional, Parametrização, Reflexão e Orientação a Aspectos (Teixeira, 2007).

Esta revisão sistemática tem por objetivo analisar a mais promissora destas técnicas, a Programação Orientada a Aspectos (POA). Isto porque, de acordo com Carmo (2005), POA lida com um problema específico: capturar unidades consistentes de um sistema de software que as limitações dos modelos de programação tradicionais forçam a ficar espalhados por diversos pontos do sistema.

A Orientação a Aspectos (OA), portanto, provê os meios para separação de código que contém interesses transversais, encapsulando-os em uma unidade denominada aspecto, elemento principal da POA (Teixeira, 2007). Além dos aspectos, outros conceitos são importantes neste contexto:

- Pontos de junção (*join points*): pontos bem definidos da execução de um programa;
- Pontos de atuação ou de corte (*pointcuts*): permitem a seleção de conjuntos de pontos de junção;
- Adendos (*advices*): determina o que será executado pelo aspecto quando for atingido um ponto de junção;
- Declarações inter-tipos (*Intertype declarations*): declarações de atributos e métodos a serem inseridos em classes, no código resultante.

Com base nos conceitos desta seção, uma breve análise dos trabalhos coletados e estudados é apresentada na seção a seguir.

3.2 Panorama sobre os Resultados Obtidos

Com a leitura na íntegra dos trabalhos selecionados foi possível identificar vários tipos de técnicas e estudos de caso que aplicam POA na implementação de LPS. Assim, os estudos estão agrupados em dois grandes grupos:

Propostas e Abordagens para a implementação de variabilidades de LPS utilizando POA: estudos que apresentam técnicas e abordagens para a aplicação de aspectos no contexto de LPS (questão primária desta revisão);

Estudos de Caso: estudos realizados para demonstrar e avaliar o emprego de POA na implementação de variabilidades de LPS.

A classificação permitiu ainda identificar propostas que seguem padrões semelhantes e podem ser, portanto, analisados em conjunto. De acordo com os dados coletados, grande parte dessas propostas é acompanhada de exemplos como forma de apresentar as principais características da abordagem, sem conduzir estudos mais detalhados para comprovar sua efetividade. Todos os trabalhos que apresentaram algum tipo de estudo experimental (questão secundária 3) estavam limitados a estudos de caso, que, para a análise, foram agrupados de acordo com o seu domínio de aplicação.

As questões secundárias 1 e 2 desta revisão referem-se a benefícios e problemas da utilização de POA para a implementação de variabilidades de LPS, respectivamente. De uma forma geral, pode-se afirmar que os principais benefícios relatados nos trabalhos incluídos são: o aumento de modularização, de reuso de código e de configurabilidade, melhor legibilidade e facilidade de manutenção. Dentre os problemas destaca-se o alto acoplamento entre código fonte e aspectos. Nas duas tabelas, as últimas colunas mostram se os trabalhos abordam benefícios e/ou problemas da utilização de aspectos no desenvolvimento de LPS.

A seguir é apresentada uma breve análise sobre os estudos classificados de acordo com a seleção apresentada. As propostas são apresentadas por escopo. Uma grande parte dos trabalhos propõe o uso combinado de POA com outras técnicas como MDD, FOA e Programação Generativa. Outros trabalhos apresentam abordagens com diretrizes para o uso de POA no desenvolvimento de LPS, por exemplo, alguns sugerem o uso de POA como ligação entre o núcleo e as variabilidades, outros indicam o uso de POA para implementação de requisitos funcionais, há ainda os que indicam o uso de *Mixin Layers* ou *Editcs* para determinados tipos de variabilidades.

Ressalta-se ainda que, dentre os trabalhos incluídos, dois deles (estudos 14 e 15) contêm tanto uma abordagem para uso de POA no desenvolvimento de

variabilidades de LPS como um estudo de caso, por isso, eles estão listados em ambas as tabelas. Dois outros estudos (42 e 43) abordam comparações entre POA e OO.

3.2.1 Propostas e Abordagens para a implementação de variabilidades de LPS utilizando POA

Na tentativa de evitar inúmeros problemas com a implementação, dentre os quais foram destacadas as questões de código espalhado, emaranhado e entrelaçado (*scattering*, *tangling* e *crosscutting code*), muitos autores analisaram técnicas e abordagens, algumas envolvendo domínios específicos, outras de modo mais generalizado. Assim, esta revisão obteve uma coletânea diversificada em técnicas, explicitadas na sequência. A classificação foi feita levando-se em consideração o estilo da aplicação de POA e a combinação desta com outros paradigmas e modelos de programação.

3.2.1.1 Feature Oriented Approachs e Programação Orientada a Aspectos

Lee et al. (2006) utilizam *Feature Oriented Analysis* (FOA) em conjunto com POA, de modo a reduzir os problemas que podem ocorrer com a implementação somente em POA. Dessa forma, obteve um aumento na reusabilidade, configurabilidade e adaptatividade da LPS.

Conejero e Hernández (2008) propõem um *framework* que indica características transversais em estágios iniciais do projeto de LPS, utilizando então POA para modularizar esses interesses e reduzir dependências a partir do início do projeto. O *framework* inclui FOA, análise dos requisitos não-funcionais e modelagem de requisitos.

Apel e Battory (2006) falam sobre a combinação entre *Feature-Oriented Programming* (FOP) e POA para implementação de LPS, uma vez que os pontos fortes de uma são os fracos da outra. A forma de combinação entre as tecnologias usadas pelos autores é denominada *Aspectual Mixin Layers*.

Mezini e Ostermann (2004) falam sobre problemas envolvendo FOA e POA na implementação de variabilidades e demonstram quando e quais estruturas são mais eficazes na resolução dos problemas. Apresentam a ferramenta *Caesar* que combina FOA e POA, tendo suporte para implementação de variabilidades.

3.2.1.2 Aplicação de Orientação a Aspectos em LPS Orientadas a Objetos

Navarro et al. (2007) discutem propostas para aplicar aspectos em sistemas Orientados a Objetos (OO), de modo a aumentar modularização, melhorar legibilidade e diminuir dependências.

Burgstaller (2006) relata sobre evolução de LPS utilizando aspectos. Seu foco está na separação das características comuns e variáveis, separação esta realizada por meio da ferramenta NAPLES.

Kulesza e Lucena (2006) apresentam uma abordagem para a extensão de *frameworks*(OO usando OA, sendo que os *frameworks* são tratados como uma família de produtos. Várias características, tidas como transversais, são representadas usando aspectos e *extension join points* (EJPs). Ainda é apresentado um modelo para instanciação automática das variabilidades de um determinado *framework*.

Cho, Lee e Kang (2008) apresentam uma abordagem orientada a aspectos aplicada especialmente para reduzir problemas entre relação de características e gerenciamento de dependências, que são muito comuns na orientação a objetos, como forma de aumentar a modularização e o reúso do código.

3.2.1.3 Decisões de implementação envolvendo POA

Ribeiro, Matos Júnior e Borba (2008) apresentam um modelo de decisões, utilizado para distinguir em quais tipos de características deve ser utilizada a POA.

Chakravarthy, Regehr e Eide (2008) tratam sobre a implementação de variabilidades levando-se em consideração a decisão acerca de variabilidades com *binding times* flexíveis, isto é, tempo de seleção de características variáveis. Para tal, combina *design patterns* e aspectos alvos denominados *edicts*.

Kulesza et al. (2006a) apresentam um estudo sobre implementação de *frameworks*, os quais são tratados como LPS. É utilizada a abordagem EJPs, que integra OA e OO, envolvendo decisões sobre a aplicação dos EJPs. Apresenta exemplos relacionados às variabilidades. A implementação dos EJPs é realizada por meio de AspectJ.

Ribeiro et al. (2007) apresentam um catálogo resumido de tipos de variabilidades encontrados em LPS. Apresenta uma análise, por meio de matrizes, para a qual conclui que a utilização de aspectos na implementação pode gerar problemas, sendo que para cada tipo de variabilidade, existe uma técnica de implementação correspondente.

Griss (2000) apresenta uma abordagem para a configuração de requisitos de LPS de acordo com características, sendo que cada uma delas pode representar as variabilidades ou partes de um núcleo comum. Assim, estas são implementadas por meio de aspectos. Dessa forma, existe a possibilidade do reuso de software.

Alves (2007) apresenta estratégias que permitem utilização de POA no contexto de LPS, de modo a obter um código mais adequado e organizado.

Mehner, Reiser e Weber (2006) definem uma técnica precisa para definir aspectos em LPS para sistemas automotivos, demonstrando quais as melhores decisões a serem tomadas neste contexto.

Em relação às decisões, Pohl et al. (2007) falam sobre a utilização de diversas técnicas para a implementação de variabilidades de LPS. POA está relacionada, sendo que são apresentados problemas em relação a esta, entre os quais se destaca o alto acoplamento entre código fonte e aspectos. Também é proposto um meio de avaliar linguagens de programação no contexto de LPS, como forma de analisar se determinada linguagem é viável do neste âmbito.

Heo e Choi (2006) apresentam as estruturas da linguagem OA AspectJ e discutem sobre a aplicação de conceitos envolvendo aspectos inclusive no processo de combinação, onde as características são selecionadas e agrupadas para a geração do produto final, de forma que os aspectos funcionam como um *glue code* entre o núcleo e as características. Também levam em consideração a aplicação em análise de requisitos e modelagem do sistema. As técnicas foram exemplificadas com um software embutido para microondas.

3.2.1.4 Refactoring de Sistemas Orientados a Objetos

Monteiro (2005) apresenta técnicas para realização de *refactoring* de sistemas OO para sistemas orientados a aspectos. Tais técnicas objetivam modularização e reuso.

3.2.1.5 Separação de Interesses

Colyer, Rashid e Blair (2004) tratam especificamente sobre a separação de interesses transversais, envolvendo POA. Apresenta bases teóricas e matemáticas para a definição e realização da separação.

Spinczyk, Lohmann, e Schröder-Preikschat (2006) apresentam sobre hierarquias de interesses (*concern hierarchies*), um modelo que auxilia na separação de interesses transversais e ordinários, de modo a aplicar os conceitos da orientação a aspectos. Tal abordagem é uma evolução de outras mais antigas desenvolvidas para LPS e demonstra em alguns pontos como tratar as variabilidades.

3.2.1.6 Abordagens generativas e POA

Kulesza et al. (2005) falam sobre a utilização de abordagens generativas orientadas a aspectos para o desenvolvimento e personalização de arquiteturas de LPS. A implementação das variabilidades é baseada nos modelos de características (*feature models*) refinados por meio da aplicação de abstrações orientadas a aspectos.

Kulesza et al. (2006b) apresentam uma abordagem que engloba orientação a aspectos e técnicas generativas (*generative techniques*) para a representação, modelagem e implementação de variabilidades em linhas de produto, especialmente as variabilidades transversais. Também são demonstradas instruções a serem realizadas nas etapas de engenharia de domínio e de aplicação. Todas as variabilidades são explicitadas por meio de modelos de características.

3.2.1.7 Model-Driven Development e POA

Voelter e Groher (2007) lidam com o tratamento das variabilidades em LPS. A orientação a aspectos é utilizada como forma de implementar as variabilidades transversais. Também é utilizada em combinação com o *Model Driven Development* (MDD), de forma a melhorar os resultados, uma vez que as duas tecnologias podem se complementar. A técnica é aplicada por meio de um exemplo de automação doméstica.

Yie et al. (2008) falam sobre a utilização de aspectos para a evolução de LPS desenvolvidas com base no MDD. Essa evolução está ligada à adição de interesses transversais. Desta forma, são reduzidos os problemas de legibilidade do código, facilitando entendimento, manutenção e documentação.

Groher, Bleicher e Schwanninger (2005) apresentam uma técnica para criação de LPS que envolve POA e MDD. A parte da implementação das características, isto é, da engenharia de domínio, é feita por meio de aspectos, enquanto o MDD é utilizado para a instanciação de produtos.

Gasiunas et al. (2007) apresentam uma análise das tecnologias e exemplos mais recentes envolvendo aplicação de POA e MDD no contexto de LPS. Demonstram exemplos, entre os quais se destaca automação doméstica. Por fim demonstra quando e como POA e MDD podem ser utilizadas em conjunto, de forma a propiciar uma melhor implementação para as LPS.

3.2.1.8 Utilização de POA para implementação de variabilidades e código de testes de LPS

Knauber e Schneider (2004) demonstram a utilização de orientação a aspectos tanto no *workflow* de implementação quanto no de teste, em relação a variabilidades em LPS. Um exemplo de máquina receptora de moedas é apresentado. São separados quatro tipos de variabilidades, explicitando o método de implementação e de testes para cada. Foi utilizada a linguagem Java e os plugins AspectJ e JUnit. A proposta principal dos autores é de que um mesmo aspecto possua o código de uma variabilidade e o seu respectivo código de teste.

3.2.1.9 Utilização de POA em arquiteturas de LPS

Oldevik (2008) fala sobre a utilização de aspectos para modelagem de arquiteturas de LPS e as vantagens trazidas para implementação das características, que no caso representam as variabilidades.

3.2.1.10 Agent Development e POA

Nunes et al. (2008) falam sobre a implementação de LPS utilizando *Agent Development*. Para algumas características, as transversais, foi utilizada POA. Também demonstra a integração entre as características comuns e as variabilidades, apresentando assim uma nova forma de abordagem para LPS.

3.2.1.11 Utilização de POA na implementação de requisitos funcionais

Pacios, Masiero e Braga (2006) apresentam a utilização de aspectos na implementação de requisitos funcionais de uma LPS. As características referentes a variabilidades são analisadas e implementadas por meio da orientação a aspectos. A abordagem apresentada pode ser tanto utilizada na criação de uma LPS, quanto no *refactoring* de uma linha já existente. O trabalho é exemplificado por meio de uma LPS para clínicas de Psicologia.

Pacios (2006) também trata sobre a implementação de requisitos funcionais em LPS utilizando POA.

Braga et.al. (2007) apresentam a abordagem AIPLES-IS, para o incremento de sistemas de informação utilizando aspectos.

3.2.2 Estudos de caso

Na tentativa de avaliar as técnicas ou mesmo reconhecer limitações de determinados modelos que envolvem POA na implementação de LPS, alguns trabalhos abordaram a realização de estudos de casos. Assim, nessa revisão obtiveram-se alguns estudos interessantes no sentido de verificar o que já foi pesquisado envolvendo as áreas citadas. As pesquisas estão apresentadas a seguir, tendo como critério de classificação a utilização em domínios comuns.

3.2.2.1 LPS para gerenciamento de Igrejas Cristãs

Teixeira (2007) apresenta a implementação de uma LPS para gerenciamento de igrejas Cristãs, a qual foi denominada Ligo. A família de produtos foi desenvolvida para funcionar via Web. Para tanto, foi utilizada como tecnologia a linguagem *phpAspect*. Além da contribuição por meio do exemplo, o autor também propõe melhorias no combinador (*weaver*) utilizado, uma vez que o mesmo apresenta alguns defeitos e pontos que ainda precisam ser estendidos.

3.2.2.2 LPS de aplicações para aparelhos móveis

Borba, Alves e Matos Júnior (2004) tomam como objeto de estudo a implementação de um jogo para celular, avaliando características das tecnologias OO e OA. Os autores chegaram à conclusão de que POA traz benefícios consideráveis à implementação de LPS, especialmente no que se refere a problemas com interesses transversais.

Carmo (2005) e Alves (2007) também realizam análises envolvendo a implementação de um jogo para aparelhos móveis. Trata-se de uma maneira de possibilitar uma maior portabilidade ao jogo, lidando especialmente com os requisitos não-funcionais, como, por exemplo, diferenças entre tamanho e resolução da tela. A linguagem de programação orientada a aspectos utilizada foi AspectJ. Ambos chegaram à conclusão que apesar da linguagem apresentar algumas limitações, a mesma pode ser utilizada para LPS de alta complexidade.

Figueiredo et al. (2008) apresentam a análise da implementação baseada em aspectos de duas LPS envolvendo manipulação de fotos para aparelhos móveis, de modo a estudar a estabilidade do modelo em cada uma. O estudo leva em consideração modularidade, propagação de mudanças e dependência entre as características.

Young (2005) apresenta a comparação entre uma mesma LPS para aparelhos móveis, implementada de duas formas: uma orientada a aspectos e outra orientada a objetos. Assim, pretende avaliar a ferramenta AspectJ em relação a linguagens OO.

Neto, Alves e Borba (2008) apresentam um estudo de caso envolvendo jogos para aparelhos móveis, onde procuram determinar dependências, de modo a eliminá-la, uma vez que esta atrapalha o desempenho e a organização do código baseado em aspectos. Assim, são demonstrados mecanismos dentro da própria linguagem AspectJ para resolver tais problemas.

Noordhuizen (2006) analisa várias linguagens orientadas a aspectos, bem como pontos para reduzir problemas devido a interesses transversais. O objetivo do trabalho é demonstrar um exemplo, de Arcade games, analisando arquitetura da LPS e problemas com interesses, especialmente os não-funcionais.

Anastasopoulos e Muthig (2004) avaliam a utilização de POA no contexto de LPS, por meio de um estudo de caso relativo a aparelhos móveis, chegando à conclusão de que a aplicação de aspectos é extremamente adequada para o contexto devido à flexibilidade, principalmente em relação à portabilidade, gerada por requisitos não-funcionais.

3.2.2.3 Frameworks

Kulesza (2007) utiliza POA na implementação de *frameworks* transversais, onde estes são tratados como uma família de produtos. As variabilidades ficam por conta dos pontos de extensão (*hot spots*), enquanto as características comuns correspondem aos pontos imutáveis (*frozen spots*). Apresenta também uma combinação com o processo de desenvolvimento generativo, de modo a melhorar seus domínios e produtos finais.

3.2.2.4 Marcapasso (Pacemaker Product Line)

Liu, Lutz e Rajan (2006) apresentam a implementação de uma mesma LPS para marcapassos utilizando OO e OA e analisa ambos os artefatos, obtendo um quadro comparativo entre os paradigmas.

3.2.2.5 Sistemas Legados

Gibbs et al. (2007) apresentam POA como uma forma de evoluir sistemas legados, em especial, para atendimento de clínicas médicas. Este tipo de sistema apresenta problemas em relação à modularização normal, problemas estes solucionáveis por meio da aplicação de aspectos.

3.2.2.6 Sistemas Automotivos

Mehner, Reiser e Weber (2006) apresentam técnicas para a definição e implementação de variabilidades em artefatos de uma LPS para sistemas automotivos. Desta forma, eles comparam abordagens existentes e uma orientada a aspectos, de forma a analisar os resultados.

3.2.2.7 GPL (Graph Product Line)

Herrejon e Batory (2002a) falam sobre a utilização de AspectJ para a implementação de uma LPS de aplicações envolvendo grafos. Os autores apresentam várias estruturas que compõem a linguagem e que podem ser utilizadas para a implementação inclusive das variabilidades.

Herrejon e Batory (2002b) apresentam um estudo de caso para LPS envolvendo grafos por meio da linguagem orientada a aspectos Hyper/J. Desta forma, procuram mostrar recursos e elementos da tecnologia que podem ser empregados para implementar variabilidades e características comuns.

3.2.2.8 Comércio Eletrônico (e-shop)

Nyßen, Tyszberowicz e Weiler (2005) apresentam um estudo de caso referente a sistemas de lojas eletrônicas com o objetivo de analisar a implementação de LPS utilizando POA, e posteriormente realizar a comparação com a orientação a objetos.

Capítulo 4 – Conclusão

A realização desta revisão sistemática possibilita uma análise dos estudos encontrados e dos números gerais sobre a revisão em si, bem como uma classificação acerca de técnicas e estudos de caso envolvendo implementação de variabilidades de LPS utilizando POA.

Com relação à revisão sistemática, percebe-se que embora o tema seja relativamente recente, várias publicações foram encontradas a partir da aplicação das *strings* nas fontes de busca. No total, 202 estudos foram coletados, sendo que 87 foram selecionados. Destes 87 estudos lidos na íntegra, somente 46 foram considerados na análise final dos resultados, ou seja, cerca de 23% do total de estudos encontrados. A partir dessas informações pode-se concluir que, para uma busca mais efetiva por estudos, ou as *strings* devem ser analisadas e elaboradas novamente, ou os mecanismos de busca das fontes não se encontram devidamente implementados para suportar revisões sistemáticas. Além disso, pode-se citar que foram encontrados vários estudos cujos resumos não demonstram com clareza o seu conteúdo ou que prometem em seus resumos mais do que realmente apresentam, o que pode explicar um índice tão alto de rejeição dos artigos na análise final.

Levando-se em consideração os resultados da revisão sistemática, os estudos foram classificados em dois grupos: estudos de caso e técnicas para aplicação de POA para implementação de variabilidades de LPS.

Quanto aos estudos de caso, foram destacados os trabalhos que apresentam experimentos, isto é, apresentam avaliação das implementações por meio de testes e análises comparativas, destacando assim vários exemplos como forma de demonstrar a aplicação dos conceitos envolvidos, no caso utilização de POA na implementação de variabilidades de LPS. São estudos nos quais engenheiros de software podem se basear para desenvolver novas LPS, aperfeiçoar abordagens ou até mesmo propor novas. Os estudos de caso encontrados apresentam domínios diferenciados, como: gerenciamento de Igrejas Cristãs, *Frameworks*, Comparações entre Aspectos e Objetos, Sistemas Legados, Sistemas Automotivos, LPS para algoritmos de grafos. Contudo, o principal foco dos estudos de caso está relacionado à implementação de LPS de aplicações para aparelhos móveis, destacando-se a criação de jogos para celulares. Neste âmbito, POA é utilizada com maior ênfase para os requisitos não-funcionais, uma vez que estes estão ligados diretamente ao fator portabilidade. Assim, os aspectos são os modificadores de resolução de tela, teclas de funcionamento da aplicação, entre outros diferenciais entre os mais diversos aparelhos.

Quanto às técnicas e abordagens, foram identificadas várias formas e contextos para aplicação de POA em LPS, isto é, vários mecanismos de decisão, de avaliação ou mesmo de construção de LPS por meio de aspectos. Assim, engenheiros de software podem se basear nestas técnicas para resolverem determinados problemas que apareçam na construção de software de qualidade, além do que podem promover extensões e outras técnicas para que o Desenvolvimento de Software Orientado a Aspectos atinja uma maturidade dentro do contexto de LPS. Dentre as técnicas encontradas estão: (i) FOA e POA, isto é, técnicas que englobam as principais características destas abordagens de modo a obter um resultado mais preciso; (ii) separação e modularização dos interesses, especialmente os transversais, por meio do encapsulamento em aspectos; (iii) aperfeiçoamento de LPS já existentes; (iv) momentos onde devem ser tomadas decisões relacionadas a POA; (v) transformação de sistemas existentes; (vi) programação generativa aliada à POA; (vii) aplicação simultânea de POA e MDD em LPS; (viii) realização de teste em LPS por meio de aspectos; (ix) estruturação de arquiteturas de LPS por meio de POA; (x) utilização conjunta de POA e *Agent Development*; (xi) implementação de requisitos funcionais de LPS por meio de aspectos. Dentre as técnicas supracitadas, a última está diretamente relacionada aos objetivos da pesquisa.

De acordo com os dados coletados, pode-se perceber que AspectJ foi a linguagem de programação de aspectos mais adotada. Ainda podem ser citadas outras linguagens, tais como Hyper/J, AspectC e phpAspect, que correspondem a outras tecnologias menos utilizadas em comparação com a primeira.

Alguns dos estudos selecionados apresentam também os benefícios alcançados e/ou problemas e dificuldades quando do uso de POA para desenvolver LPS.

Uma grande parte dos trabalhos incluídos apresenta o uso de POA para resolver problemas com interesses transversais e implementação de requisitos não-funcionais. Ainda assim, existem trabalhos nos quais POA é usada para implementar variabilidades de uma forma geral, independente do tipo de requisito ou tipo de interesse.

Finalmente, os resultados e análises obtidos pela revisão sistemática serão a base para uma pesquisa posterior. Tal pesquisa refere-se à implementação de variabilidades de uma LPS, construída baseada em componentes, por meio de POA, de forma a demonstrar e analisar as vantagens da aplicação de aspectos no mesmo contexto.

Referências

Alves, V. R.: **Implementing Software Product Line Adoption Strategies**. Recife, Março, 2007. Tese de Doutorado (Doutorado em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco. 178 p.

Anastasopoulos, M. and Gacek, C. (2001).: Implementing Product Line Variabilities. In the 2001 Symposium on Software Reusability (SSR'01). **Proceedings...**, Pages 109–117, New York, NY, USA. ACM Press.

Anastasopoulos, M.; Muthig, D.: An evaluation of aspect-oriented programming as a product line implementation technology. In: ICSR '04: **Proceedings of the 8th International Conference Software Reuse**, Springer, 2004, p. 141–156.

Anthony P., Somé, S. S.: Aspect-oriented use case modeling for software product lines. In the 7th International Conference on Aspect-Oriented Software Development. **Proceedings of the Early Aspects Workshop**. Bruxelas, Bélgica. Março 2008.

Apel, S.; Batory, D.: When to Use Features and Aspects? A Case Study. In: **Proceedings of the 5th International Conference on Generative Programming and Component Engineering**, ACM Press New York, NY, USA, 2006, p. 59–68.

Apel, S., Leich, T., Saake, G.: Aspect Refinement in Software Product Lines. In the 9th International Software Product Line Conference. **Proceedings of the Workshop on Aspects and Product-Lines – An Early Aspects Workshop**. Rennes, França. Setembro 2005a.

Apel, S., Leich, T., Saake, G.: Aspect Refinement and Bounding Quantification in Incremental Designs. In the 12th Asia-Pacific Software Engineering Conference. **Proceedings...**, Taipei, Taiwan. Dezembro 2005b.

Apel, S., Leich, T.; Saake, G.: Aspectual Feature Modules. *Software Engineering*, **IEEE Transactions in Software Engineering**, IEEE Transactions, Vol. 34, No. 2. (2008), pp. 162-180.

Beuche, D., Spinczyk, O., Schröder-Preikschat, W.: Finegrain Application Specific Customization for Embedded Software. In the International IFIP TC10 Workshop on Distributed and Parallel Embedded Systems. **Proceedings...**, Montreal, Canada. Agosto 2002.

Biolchini, J.; Mian, P. G.; Natali, A. C. C.; Travassos, G. H. **Systematic review in software engineering**. Technical report RT-ES 679/05, PESC - COPPE/UFRJ, 2005.

Borba, P., Alves, V., Matos Júnior, P. O. A.: **Variações em Linhas de Produto de Software usando Programação Orientada a Aspectos**. Relatório de atividades de projeto de pesquisa. 2004.

Braga, R. T. V., Germano, F. S. R., Pacios, S. F., Masiero, P. C.: AIPLE-IS: An Approach to Develop Product Lines for Information Systems Using Aspects. In: Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software 2007, Campinas - SP. **Proceedings...**, Campinas - SP: Unicamp, 2007. v. 1. p. 17-30.

Burgstaller, R.: **Product Line Evolution and AOP**. Munique, Junho, 2006. Elaboração para Seminário.

Camargo, V. V.: **Frameworks transversais: definições, classificações, arquitetura e utilização em um processo de desenvolvimento de software**. São Carlos, Agosto, 2006. Tese de Doutorado (Doutorado em Ciência da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. 280 p.

Carmo, H. V.: **Técnicas de Construção de Linha de Produto para Jogos Móveis**. Recife, Agosto, 2005. Trabalho de Graduação (Trabalho para Graduação em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco. 47 p.

Chavrarathy, V., Regehr, J., Eide, E.: Edicts: implementing features with flexible binding times. In the 7th International Conference on Aspect-Oriented Software Development. **Proceedings...**, Bruxelas, Bélgica. Março 2008.

Cho, H., Lee, K., Kang, K. C.: Feature Relation and Dependency Management: An Aspect-Oriented Approach. In 12th Software Product Line Conference (SPLC2008). **Proceedings...**, Limerick, Ireland, Sep 8-12, 2008, pp.3-11.

Cirilo, E., Kulesza, U., Lucena, C. J. P.: GenArch: A Model-Based Product Derivation Tool. In Simpósio Brasileiro de Componentes, Arquitetura e Reutilização de Software. **Proceedings of Software Product Line Technical Session**. São Paulo, Brasil. Agosto 2007.

Clarke, S., Walker, R. J.: Towards a standard design language for AOSD. In the 1st International Conference on Aspect-Oriented Software Development. **Proceedings...**, Enschede, Holanda. Abril 2002.

Conejero, J. M., Hernández, J.: Analysis of Crosscutting Features in Software Product Lines. In the 30th International Conference on Software Engineering. **Proceedings of the 13th International Workshop on Software Architectures and Mobility**. Leipzig, Alemanha. Maio, 2008.

Colyer, A., Rashid, A., Blair, G.: **On the Separation of Concerns in Program Families**. Technical Report, Computing Department, Lancaster University, 2004.

Donegan, P. M.: **Geração de Famílias de Produtos de Software com Arquitetura Baseada em Componentes**. São Carlos, junho, 2008. Dissertação (Mestrado em Ciência da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. 182 p.

Figueiredo, E., Cacho, N., Sant'Anna, C., Monteiro, M., Kulesza, U., Garcia, A., Soares, S., Ferrari, F., Khan, S., Castor Filho, S., Dantas, F.: Evolving Software Product Lines With Aspects: An Empirical Study On Design Stability. In the 30th International Conference on Software Engineering. **Proceedings...**, Leipzig, Alemanha. Maio 2008.

Fuentes, L., Gamez, N.: A Feature Model Of An Aspect-Oriented Middleware Family For Pervasive Systems. In the 7th International Conference on Aspect-Oriented Software Development. **Proceedings of the 2008 workshop on Next generation aspect oriented middleware**. Bruxelas, Bélgica. Março 2008.

Fuentes, L., Jiménez, D.: Combining Components, Aspects, Domain Specific Languages and Product Lines for Ambient Intelligent Application Development. In the 4th International Conference on Pervasive Computing. **Proceedings...**, Dublin, Irlanda. Maio 2006.

Gasiunas, V., Sánchez, P., Nebrera, C., Gámez, N., Fuentes, L., Noyé, J., Südholdt, M., Núñez, A., Pohl, C., Rummler, A., Groher, I., Schwanninger, C., Völter, M.: **Overview of Extensions/Improvements to Existing Implementation Technologies**. Technical Report, AMPLE (Aspect –Oriented, Model-Driven, Product Line Engineering) Specific Targeted Research Project: IST-33710, 2007.

Gibbs, C., Lohmann, D., Liu, C. R., Coady, Y.: Modular Integration Through Aspects: Making Cents of Legacy Systems. In the 40th Annual Hawaii International Conference on System Sciences. Big Island, Hawaii. **Proceedings...**, Janeiro 2007.

González-Baixauli, B., Leite, J. C. S. P., Laguna, M.: **Using Goal-Models to Analyze Variability**. In First International Workshop on Variability Modelling of Software-Intensive Systems. **Proceedings...**, Limerick, Ireland : Lero Technical Report 2007-01, 2007. v. 1. p. 101-108.

Griss, M. L.: Implementing Product-Line Features by Composing Component Aspects. In the First International Software Product-Line Conference. **Proceedings...**, Denver, EUA. Agosto 2000.

Groher, I., Bleicher S., Schwanninger C.: Designing Features as Pluggable Collaborations. In the 9th International Software Product Line Conference.

Proceedings of the Workshop on Aspects and Software Product Lines. Rennes, França. Setembro 2005.

Groher, I., Schwanninger, C., Voelter, M.: An Integrated Aspect-Oriented Model-Driven Software Product Line Tool Suite. In the 30th International Conference on Software Engineering. **Proceedings...**, Leipzig, Alemanha. Maio 2008.

Groher, I., Voelter, M.: Xweave: Models And Aspects In Concert. In the 10th International Workshop on Aspect-Oriented Modeling. **Proceedings...**, Vancouver, Canada. Março 2007.

Guedes, A. M.: **A Variant Metamodel Language to Support Software Product Lines.** 2008.

Heo, S., Choi, E. M.: Representation of Variability in Software Product Line Using Aspect-Oriented Programming. In the Fourth International Conference on Software Engineering Research, Management and Applications. **Proceedings...**, Porto, Portugal. Setembro 2009.

Herrejon, R. E. L.: Understanding Feature Modularity in Feature Oriented Programming and its Implications to Aspect Oriented Programming. In the 15th International Network for PhD Students in Object Oriented Systems Workshop. **Proceedings...**, Glasgow, Escócia. Julho, 2005.

Herrejon, R. L., Batory, D.: **Using AspectJ to Implement Product-Lines: A Case Study.** Technical Report, Dept. of Computer Sc., University of Texas at Austin, 2002a.

Herrejon, R. L., Batory, D.: **Using Hyper/J to Implement Product-Lines: A Case Study.** Technical Report, Dept. of Computer Sc., University of Texas at Austin, 2002b.

Jézéquel, J. M.: Modeling and Aspect Weaving. In the Methods for Modelling Software Systems 2006. **Proceedings...**, Dagstuhl, Alemanha. Agosto 2006.

Kastner, C., Apel, S., Batory, H.: A Case Study Implementing Features Using AspectJ. In the 11th International Software Product Line Conference. **Proceedings...**, Kyoto, Japão. Setembro 2007.

Khan, S.S., Ali, S., Jaffar-ur-Rehman, M.: An Enhanced Framework For Validation Of Aspectual Requirements. In the IEEE Symposium on Emerging Technologies 2005. **Proceedings...**, Islamabad, Paquistão. Setembro 2005.

Kitchenham, B. A.; Dybå, T.; Jørgensen, M. Evidence-Based Software Engineering. In the 26th International Conference on Software Engineering (ICSE'04). **Proceedings...**, Scotland, Washington, DC, USA: IEEE Computer Society, 2004. Disponível em

<http://portal.acm.org/citation.cfm?id=999432&coll=Portal&dl=ACM&CFID=19556200&CFTOKEN=76099919>

Knauber, P., Schneider, J.: **Tracing Variability from Implementation to Test Using Aspect-Oriented Programming**, Avaya Labs Technical Report: ALR-2004-031: International Workshop on Software Product Line Testing (SPLiT 2004), Co-located with the 3rd International Software Product Line Conference, SPLC 2004, August 2004, Boston, Massachusetts, USA.

Krüger, I., Mathew R., Meisinger, M.: From Scenarios to Aspects: Exploring Product Lines. In the 27th International Conference on Software Engineering. **Proceedings of the ICSE 2005 Workshop on Scenarios and State Machines (SCESM'05)**, ACM Press, Maio 2005.

Kuhlemann, M., Kästner, C.: Reducing the Complexity of AspectJ Mechanisms for Recurring Extensions. In the 6th International Conference on Generative Programming and Component Engineering. **Proceedings of the GPCE Workshop on Aspect-Oriented Product Line Engineering (AOPLE)**. Salzburg, Austria. Outubro 2007.

Kulesza, U.: **Uma Abordagem Orientada a Aspectos para o Desenvolvimento de Frameworks**. Rio de Janeiro, Abril, 2007. Tese de Doutorado (Doutorado em Informática) – Programa de Pós-Graduação em Informática, Pontifícia Universidade Católica do Rio de Janeiro. 205 p.

Kulesza, U., Coelho, R., Alves, V., Costa Neto, A., Garcia, A., Lucena, C. J. P., Staa, A. V., Borba, P.: Implementing Framework Crosscutting Extensions with EJP's and AspectJ. In ACM SIGSOFT 20th Brazilian Symposium on Software Engineering (SBES'06), 2006a, Florianópolis. **Proceedings of the 20th Brazilian Symposium on Software Engineering**. p. 177-192.

Kulesza, U.; Garcia, A.; Bleasby, F.; Lucena, C.J.P.; Instantiating and Customizing Product Line Architectures using Aspects and Crosscutting Feature Models. In the Object-Oriented Programming, Systems, Languages & Applications (OOPSLA'2005). **Proceedings of the Workshop Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design**. San Diego, Outubro 2005.

Kulesza, U., Lucena, C. J. P. de.: An Aspect-Oriented Approach To Framework Development; Software Product Lines Doctoral Symposium, In conjunction with the Software Product Lines International Conference (SPLC), 10th; **Proceedings of the Software Product Lines Doctoral Symposium**, pp. 67-79, ISBN: 0-7695-2599-7, Baltimore, MD, USA, 22 de Agosto, 2006, Fraunhofer: IESE Technical Report.

Kulesza, U., Lucena, C. J. P., Alencar, P., Garcia, A.: Customizing Aspect-Oriented Variabilities Using Generative Techniques. In the 18th International Conference on Software Engineering Knowledge Engineering. **Proceedings...**, São Francisco, EUA. Julho 2006b.

Kuloor, C., Eberlein, A.: Aspect-Oriented Requirements Engineering For Software Product Lines. In the 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems. **Proceedings...**, Huntsville, EUA. Abril 2003.

Kuttruff, V.: Towards a Model for Specifying and Composing Concerns in Software Product Line Engineering. In the PRIMMIUM Subconference at the Multikonferenz Wirtschaftsinformatik (MKWI) 2008. **Proceedings...**, Munique, Alemanha. Fevereiro 2008.

Lafferty, D.: Avoiding Incorrect and Unpredictable Behaviour with Attribute-based Crosscutting. In the 4th AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software. **Proceedings...**, Chicago, EUA. Março 2005.

Lee, K., Kang, K. C., Kim, M., Park, S.: Combining Feature-Oriented Analysis and Aspect-Oriented Programming for Product Line Asset Development. In the 10th International on Software Product Line Conference. **Proceedings...**, Baltimore, EUA. Agosto 2006.

Lesaint, D., Papamargaritis, G.: Aspects and Constraints for Implementing Configurable Product-Line Architectures. In the Fourth Working IEEE/IFIP Conference on Software Architecture. **Proceedings...**, Oslo, Noruega. Junho 2004.

Liu, J., Lutz, R. R., Rajan, H.: The Role of Aspects in Modeling Product Line Variabilities. In the 5th International Conference on Generative Programming and Component Engineering. **Proceedings of the GPCE 2006 Workshop on Aspect-Oriented Product Line Eng.**, 2006. Portland, EUA. Outubro 2006.

Liu, X., Feng, Y., Kerridge, J.: **Generative Aspect-Oriented Component Adaptation**. Publicado no Institute of Engineering and Technology Software, 2008, Vol. 2, No. 2, pp. 149–160/149.

Loughran, N., Rashid, A., Zhang, W., Jarzabek, S.: Supporting Product Line Evolution with Framed Aspects. In the 2004 International Conference on Aspect-Oriented Software Development. **Proceedings of the Workshop on Aspects, Components and Patterns for Infrastructure Software**. Lancaster, Reino Unido. Março 2004.

Matos Jr, P., Duarte, R., Cardim, I., Borba, P.: Using Design Structure Matrices to Assess Modularity in Aspect-Oriented Software Product Lines. In the First International Workshop on Assessment of Contemporary Modularization Techniques. **Proceedings...**, Minneapolis, EUA. Maio 2007.

Mehner, K., Reiser, M. O., Weber, M.: Applying Aspect-Oriented Techniques in Automotive Software Product-Line Engineering. In the International Automotive Requirements Engineering Workshop. **Proceedings...**, Mineeapolis, EUA. Setembro 2006.

Mezini, M.; Ostermann, K.: Variability Management with Feature-Oriented Programming and Aspects. In: **Proceedings of the 12th ACM SIGSOFT International Symposium on Foundations of Software Engineering**, ACM Press New York, California, USA, 2004, p. 127–136.

Monteiro, M. J. T. P.: **Refactorings to Evolve Object-Oriented Systems with Aspect-Oriented Concepts**. Braga, Março, 2005. Tese de Doutorado (Doutorado em Informática) – Departamento de Informática, Universidade do Minho. 242 p.

Morin, B., Klein, J., Barais, O., Jézéquel, J.M.: A Generic Weaver for Supporting Product Lines. In the 13th International Workshop on Software Architectures and Mobility. **Proceedings...**, Leipzig, Alemanha. Maio, 2008.

Murphy, G. C., Lai, A., Walker, R. J., Robilliard, M. P.: Separating Features in Source Code: An Exploratory Study. In the 23rd International Conference on Software Engineering. **Proceedings...**, Toronto, Canadá. Maio, 2001.

Mussbacher, G., Amyot, D, Araújo, J., Moreira, A.: Modeling Software Product Lines with AOURN. In the 7th International Conference on Aspect-Oriented Software Development. **Proceedings of the Early Aspects Workshop**. Bruxelas, Bélgica. Março 2008.

Navarro, L. D. B., Schwanninger, C., Sobotzik, R., Südholt, M.: ATOLL: Aspect-Oriented Toll System. In the 6th Workshop on Aspects, Components, and Patterns for Infrastructure Software. **Proceedings...**, Vancouver, Canadá. Março 2007.

Neto, A. C., Alves, E., Borba, P.: Declaring Static Crosscutting Dependencies in AspectJ. In the III Brazilian Workshop on Aspect-Oriented Software Development. **Proceedings...**, Florianópolis, Brasil. Outubro 2006.

Noordhuizen, P.: **Analyzing Aspects in Production Plans for Software Product Lines**. Enschede, Julho, 2006. Dissertação de Mestrado (Mestrado em Engenharia de Software) – Departement of Electrical Engineering, Mathematics and Computer Science, University of Twente. 176 p.

Nunes, C. P. B., Nunes, I. O., Kulesza, U., Lucena, C. J. P.: **Developing and Evolving Multi-Agent System Product Lines**. Rio de Janeiro, Janeiro, 2008. Monografia em Ciência da Computação – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro. 25 p.

Nyßen, A., Tyszberowicz, S., and Weiler, T.: Are Aspects useful for Managing Variability in Software Product Lines? A Case Study. In the 9th International Software Product Line Conference. **Proceedings of the Workshop on Aspects and Product-Lines – An Early Aspects Workshop**. Rennes, France. Setembro 2005.

Oldevik, J.: Can aspects model product lines?. In the 2008 International Conference on Aspect-Oriented Software Development. **Proceedings of the 2008 AOSD Workshop on Early aspects**. Bruxelas, Bélgica. Março 2008.

Pacios, S. F.: **Uma Abordagem Orientada a Aspectos para Desenvolvimento de Linhas de Produto de Software**. São Carlos, Dezembro, 2006. Dissertação de Mestrado (Mestrado em Ciência da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo. 164 p.

Pacios, S. F., Masiero, P. C., Braga, R. T. V.: Guidelines for Using Aspects to Evolve Product Lines. In the Simpósio Brasileiro de Engenharia de Software 2006. **Proceedings of the III Workshop Brasileiro de Desenvolvimento de Software Orientado a Aspectos**. Florianópolis. v.1. p.111 - 120. Outubro 2006.

Pohl, C., Rummler, A., Gasiunas, V., Loughran, N., Arboleda, H., Fernandes, F. A., Noyé, J., Núñez, A., Passama, R., Royer, J. C., Südholt, M.: **Survey of existing implementation techniques with respect to their support for product lines**. Technical Report, AMPLE (Aspect –Oriented, Model-Driven, Product Line Engineering) Specific Targeted Research Project: IST- 33710, 2007.

Ramos, R. C.: **Aspecting: Abordagem para Migração de Sistemas OO para Sistemas OA**. São Carlos, Setembro 2004. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de São Carlos. 141 p.

Ramos, R. A., Penteado, R., Masiero, P. C.: Um Processo de Reestruturação de Código Baseado em Aspectos. In the XVIII Simpósio Brasileiro de Engenharia de Software. **Proceedings...**, Brasília, Brasil. Outubro 2004.

Ribeiro, M., Matos Junior, P., Borba, P., Cardim, I.: On the Modularity of Aspect-Oriented and other Techniques for Implementing Product Lines Variabilities. In the XXI Simpósio Brasileiro de Engenharia de Software (SBES 2007), 2007, João Pessoa. **Proceedings of the I Latin American Workshop on Aspect-Oriented Software Development (LA-WASP 2007)**, 2007. p. 119-130.

Ribeiro, M. M., Matos Jr, P., Borba, P.: Decision Model for Implementing Product Lines Variabilities. In the 2008 ACM Symposium on Applied Computing. **Proceedings...**, Fortaleza, Brasil. Março 2008.

- Saleh M., Gomaa, H.: **Separation of concerns in software product line engineering**. ACM SIGSOFT Software Engineering Notes 30 (4): 1-5 (2005).
- Siy, H., Aryal, P., Winter, V., Zand, M.: Aspectual Support for Specifying Requirements in Software Product Lines. In the 2007 International Conference on Software Engineering. **Proceedings of the Early Aspects at ICSE: Workshops in Aspect-Oriented Requirements Engineering and Architecture Design**. Minneapolis, EUA. Maio 2007.
- Siy, H., Zand, M., Winter, V.: The Role of Aspects in Domain Engineering. In the 2005 Software Product Line Conference. **Proceedings of the Aspects and Software Product Lines: An Early Aspects Workshop at SPLC-Europe 2005**. Rennes, France.
- Spinczyk, O., Lohmann, D., Schröder-Preikschat, W.: Concern hierarchies. In the 2004 International Conference on Generative Programming and Component Engineering. **Proceedings of the 1st GPCE Workshop on Aspect-Oriented Product Line Engineering**. Portland, EUA. Outubro 2006.
- Teixeira, M. L.: **Ligo: Uma linha de produtos de software para gerenciamento de igrejas cristãs**. Recife, Dezembro, 2007. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) – Escola Politécnica de Pernambuco, Universidade de Pernambuco. 56 p.
- Ubayashi, N., Sano, S., Otsubo, G.: A Reflective Aspect-Oriented Model Editor Based on Metamodel Extension. In the International Workshop on Modeling in Software Engineering. **Proceedings...**, Minneapolis, EUA. Maio 2007.
- Vasconcelos, A. T.: **Ferramenta para Construção de Linha de Produtos no Eclipse**. Recife, Agosto, 2005. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco. 65 p.
- Verdin, K. C., Olalde, C. L.: Assessment of Product Line Architecture and Aspect Oriented Software Architecture Methods. In the First Workshop on Aspect Oriented Product Line Engineering. **Proceedings...**, Portland, EUA. Outubro 2006.
- Voelter, M., Groher, I.: Product Line Implementation using Aspect-Oriented and Model-Driven Software Development. In the 11th International Software Product Line Conference. **Proceedings...**, Kyoto, Japão. Setembro 2007.
- Voelter, M., Groher, I.: Handling variability in model transformations and generators. In 7th OOPSLA Workshop on Domain-Specific Modeling. **Proceedings...**, Montreal, Canada. Outubro 2007.
- Winck, D. V.: **Extensão para UML Destinada à Modelagem de Variabilidade Transversal em Componentes através da Orientação a Aspectos**. Florianópolis, Junho,

2005. Dissertação de Mestrado (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina. 87 p.

Yie, A., Casallas, R., Deridder, D., Van der Straeten, R.: Multi-step concern refinement. In the 2008 International Conference on Aspect-Oriented Software Development. **Proceedings of the 2008 AOSD Workshop on Early aspects**. Bruxelas, Bélgica. Março 2008.

Young, T. J.: **Using AspectJ to Build a Software Product Line for Mobile Devices**. Dissertação de Mestrado (Mestrado em Ciência da Computação) – University of British Columbia. 73 p.

Zhang, H.: Handling Variants in a Product Family. In the Third International Symposium on Generative and Component-based Software Engineering (GCSE 2001). **Proceedings of the Young Researchers Workshop**. Erfurt, Germany, September 2001.