



Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

Rodrigo Jardim Monteiro da Fonseca

**Modelagem conceitual da estação fotogramétrica digital
educacional E-FOTO usando UML, Padrões e Aspectos**

Rio de Janeiro

2008

Rodrigo Jardim Monteiro da Fonseca

**Modelagem conceitual da estação fotogramétrica digital educacional
E-FOTO usando UML, Padrões e Aspectos**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia da Computação, da Universidade do Estado do Rio de Janeiro. Área de concentração: Geomática.

Orientador: Prof. Dr. Guilherme Lúcio Abelha Mota

Rio de Janeiro

2008

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

F676 Fonseca, Rodrigo Jardim Monteiro da.
Modelagem conceitual da estação fotogramétrica digital educacional
E-FOTO usando UML, Padrões e Aspectos / Rodrigo Jardim Monteiro
da Fonseca – 2008.
130f.

Orientador: Guilherme Lúcio Abelha Mota.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro,
Faculdade de Engenharia.

1. Engenharia da Computação. 2. Engenharia de software -
Dissertações. 3. Fotogrametria - Dissertações. 4. UML (Linguagem de
modelagem padrão) - Dissertações. I. Mota, Guilherme Lúcio Abelha.
II. Universidade do Estado do Rio de Janeiro. III. Título.

CDU 528.7

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta
tese, desde que citada a fonte.

Assinatura

Data

Rodrigo Jardim Monteiro da Fonseca

**Modelagem conceitual da estação fotogramétrica digital educacional
E-FOTO usando UML, Padrões e Aspectos**

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia da Computação, da Universidade do Estado do Rio de Janeiro. Área de concentração: Geomática.

Aprovada em 24 de Novembro de 2008.

Banca Examinadora:

Prof. Dr. Guilherme Lúcio Abelha Mota (Orientador)
Faculdade de Engenharia - UERJ

Prof. Dr. Guilherme Lúcio Abelha Mota
UERJ

Prof. Dr. Orlando Bernardo Filho
UERJ

Prof. Dr. Jorge Luis Nunes Silva e Brito
UERJ

Prof. Dr. Alberto Tavares da Silva
UNICARIOCA

Rio de Janeiro

2008

DEDICATÓRIA

À Simone, À minha mãe, Às minhas irmãs.

AGRADECIMENTOS

À Simone pelo carinho, compreensão, paciência e ajuda durante todo o estudo e dedicação à esta dissertação.

À minha mãe e irmãs pelo amor, dedicação e educação.

À Universidade do Estado do Rio de Janeiro pela oportunidade concedida para o meu crescimento e afeiçoamento acadêmico e profissional.

Às Professoras Albanita Viana de Oliveira e Denise M. S. Gerscovich pelo apoio e confiança.

Ao meu orientador Professor Guilherme pela paciência e persistência.

Aos co-orientadores Professores Orlando e Nunes pelos esclarecimentos que tornaram esta dissertação possível.

RESUMO

FONSECA, Rodrigo Jardim Monteiro da. *Modelagem conceitual da estação fotogramétrica digital educacional E-FOTO usando UML, Padrões e Aspectos*. 2008. 130 f. Dissertação (Mestrado em Engenharia da Computação) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2008.

Engenharia de Software tem por objetivo o desenvolvimento e organização de sistemas complexos, impondo regras e processos bem delineados para a solução de problemas, objetivando o desenvolvimento de software de qualidade. Esta dissertação utiliza as técnicas de Engenharia de Software para a aplicação no projeto E-FOTO, que disponibiliza uma Estação Fotogramétrica Digital Educacional (EFED) para o aprendizado da fotogrametria. Aplicou-se a metodologia de processo unificado, utilizando a *Unified Model Language* (UML) propondo um modelo conceitual, convergindo os conhecimentos técnicos-científicos das áreas de sistemas e cartografia. Paralelamente outros dois modelos foram elaborados: UML com Padrões de Projeto e UML com Aspectos. O primeiro modela classes ou objetos que se relacionam entre si, gerando o núcleo da solução de um determinado problema, podendo ser utilizado em diversos contextos; o segundo permite a separação dos requisitos transversais do sistema. Realizou-se uma análise comparativa, utilizando-se a metodologia de Ponto de Função, entre os três modelos. A partir dessa análise, indica-se o modelo UML com a utilização de Padrões de Projeto para ser empregado em projetos científicos com características similares ao E-FOTO. O modelo conceitual apresentado poderá ser empregado em trabalhos futuros para a concretização do modelo físico e sua posterior implementação.

Palavras-chave: Engenharia de Software; Fotogrametria; Código-livre.

ABSTRACT

FONSECA, Rodrigo Jardim Monteiro da. *Conceptual modeling of the E-PHOTO educational digital photogrammetric station using UML, Standards and Aspects*. 2008. 130 f. Dissertação (Mestrado em Engenharia da Computação) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2008.

Software Engineering aims at developing and organizing complex systems by imposing rules and procedures outlined for the solution of problems focusing software quality. The present master dissertation applies software engineering techniques in the context of E-FOTO project which offers a free GNU/GPL educational Digital Photogrammetric Workstation (DPW) for computer assisted education in cartographic engineering. In this work, the Unified Process (UP) and Unified Model Language (UML) were applied for the production of conceptual models, encompassing a representative part of the knowledge in such scientific field. The result of such work derived three distinct conceptual models, so-called, Pure UML, Aspect UML and Design Patterns. The models were compared in order to get an answer for the question: Among these approaches, which one is the most compatible to scientific projects like E-FOTO? The comparison was supported by the function point analysis. The outcomes indicate Design Patterns as the approach to be used in scientific projects similar to E-FOTO. Thus, in future works, the Design Patterns conceptual model will be employed for the physical model derivation and its implementation.

Keywords: Software Engineering; Photogrammetry; Open source.

LISTA DE ILUSTRAÇÕES

Figura 1 - Símbolo que representa um Ator	20
Figura 2 - Símbolo que representa um Caso de Uso	20
Figura 3 - Exemplo de um relacionamento do tipo <i>extend</i>	21
Figura 4 - Exemplo de um relacionamento do tipo <i>include</i>	22
Figura 5 - Símbolo do relacionamento	22
Figura 6 - Símbolo de uma classe	23
Figura 7 - Associação entre classes	24
Figura 8 - Classes associadas com Agregação e Composição	24
Figura 9 - Generalização de uma classe	25
Figura 10 - Dependência entre classes	26
Figura 11 - Artefatos do Diagrama de Sequência	27
Figura 12 - Diagrama de Seqüência	28
Figura 13 - Interesse transversal sem uso de POA	29
Figura 14 - Combinador de Aspectos	30
Figura 15 - <i>Logging</i> com o uso de Aspecto	32
Figura 16 - Padrão Singleton	34
Figura 17 - Padrão Builder	35
Figura 18 - Padrão Abstract Factory	36
Figura 19 - Padrão Composite	37
Figura 20 - Representação esquemática da fotogrametria	39
Figura 21 - Linha Evolucionária da Fotogrametria	40
Figura 22 - Etapas do processo fotogramétrico em uma EFD	41
Figura 23 - Classes da modelagem de Definição do Projeto	42
Figura 24 - Classes da modelagem de Sensor	43
Figura 25 - Classes da modelagem de Imagem e Bloco de Imagem	45
Figura 26 - Diagrama de Classe preliminar da Orientação Interior	47
Figura 27 - Classe da Orientação Interior	48
Figura 28 - Esquema da Orientação Exterior	49
Figura 29 - Generalização da Orientação Exterior	51
Figura 30 - Interseção espacial	52
Figura 31 - Um par estereoscópico normalizado - geometria epipolar	53
Figura 32 - Relação entre fotografia e imagem digitalizada	54
Figura 33 - Classes da normalização	55
Figura 34 - Tipos de malha de MDS	56
Figura 35 - Modelagem da extração do MNE	57

Figura 36 - Imagem em perspectiva central apresentando o deslocamento devido ao relevo (imagem dos prédios à direita da figura)	57
Figura 37 - Imagem em perspectiva ortogonal, onde é eliminado o deslocamento devido ao relevo	58
Figura 38 - Modelagem da ortorretificação	59
Figura 39 - Retificação diferencial - processo	59
Figura 40 - Fluxo do método da Retificação diferencial	60
Figura 41 - Modelagem das classes da restituição digital	60
Figura 42 - Diagrama inicial do E-FOTO	64
Figura 43 - Diagrama de caso de uso da EFDE	65
Figura 44 - Diagrama de componentes da interface com o arquivo XML	66
Figura 45 - Modelo de classes preliminar da medição de qualidade	66
Figura 46 - Modelo de caso de uso para orientação exterior	67
Figura 47 - Modelo de caso de uso para cálculo do MDT com normalização	68
Figura 48 - Modelo de caso de uso para cálculo da ortorretificação	69
Figura 49 - Classe Projeto - Modelo conceitual	70
Figura 50 - Classe Sensor - Modelo conceitual	72
Figura 51 - Classe Bloco - Modelo conceitual	73
Figura 52 - Classe Ponto - Modelo conceitual	74
Figura 53 - Classe Ajustamento por Bloco - Modelo conceitual	75
Figura 54 - Restituição Fotogramétrica, DEM e Ortorretificação - Modelo conceitual	76
Figura 55 - Modelo de Caso de Uso com Aspectos	78
Figura 56 - Modelagem de um Aspecto	79
Figura 57 - Modelagem da Orientação Exterior, segundo Aspectos	80
Figura 58 - Classe XML com Padrão Singleton	81
Figura 59 - Classe Sensor, com Padrão Builder	82
Figura 60 - Classe Feição com Padrão Composite	83
Figura 61 - Modelagem de Pontos com Padrão Abstract Factory	84
Figura 62 - Diagrama de Caso de Uso para a EFDE	106
Figura 63 - Diagrama de Caso de Uso para o módulo Projeto	106
Figura 64 - Diagrama de Caso de Uso para o módulo Imagem	107
Figura 65 - Diagrama de Caso de Uso para o módulo Orientação Interior	107
Figura 66 - Diagrama de Caso de Uso para o módulo Orientação Exterior	108
Figura 67 - Diagrama de Caso de Uso para o módulo Extração do MNE	108
Figura 68 - Diagrama de Caso de Uso para o módulo Ortorretificação	109
Figura 69 - Diagrama de Caso de Uso para o modelo orientado a Aspecto	109
Figura 70 - Diagrama de Classe para o módulo Projeto	111
Figura 71 - Diagrama de Classe para a criação do bloco de imagens	111
Figura 72 - Diagrama de Classe para o módulo Orientação Interior	112

Figura 73 - Diagrama de Classe para o módulo Orientação Exterior por Resseção Espacial	112
Figura 74 - Diagrama de Classe para o módulo Orientação Exterior por Fototriangulação	113
Figura 75 - Diagrama de Classe para o módulo Normalização	113
Figura 76 - Diagrama de Classe para o módulo Extração do MNE	114
Figura 77 - Diagrama de Classe para os módulos Visualização Estéreo, Restituição Fotogramétrica e DEM	114
Figura 78 - Diagrama de Classe para o módulo Ortoretificação	115
Figura 79 - Diagrama de Classe para o módulo Projeto utilizando padrões Singleton e Builder	116
Figura 80 - Diagrama de Classe para o módulo Visão Estéreo utilizando padrão Composite	117
Figura 81 - Diagrama de Classe para Pontos/Coordenadas utilizando padrão Abstract Factory	117
Figura 82 - Diagrama de Classe modelagem dos Aspectos	118
Figura 83 - Diagrama de Classe do módulo Projeto com Aspecto	118
Figura 84 - Diagrama de Classe do módulo Imagem com Aspecto	119
Figura 85 - Diagrama de Classe do módulo Orientação Interior com Aspecto	119
Figura 86 - Diagrama de Classe do módulo Orientação Exterior com Aspecto	120
Figura 87 - Diagrama de Classe do módulo Extração do MNE com Aspecto	120
Figura 88 - Diagrama de Classe do módulo Ortoretificação com Aspecto	121
Figura 89 - Diagrama de Seqüência para o módulo Projeto - Novo Projeto	123
Figura 90 - Diagrama de Seqüência para o módulo Projeto - Abrir Projeto	123
Figura 91 - Diagrama de Seqüência para o módulo Imagem - Carregar Imagens	124
Figura 92 - Diagrama de Seqüência para o módulo Orientação Interior - Calcular	124
Figura 93 - Diagrama de Seqüência para o módulo Orientação Exterior - Calcular	125
Figura 94 - Diagrama de Seqüência para o módulo Extração do MNE	125
Figura 95 - Diagrama de Seqüência para o módulo Ortoretificação - Extração	126

LISTA DE TABELAS

Tabela 1 - Modelagem de uma classe de Aspectos	31
Tabela 2 - Diferentes Níveis de Descrição dos Requisitos	62
Tabela 3 - Requisitos da EFDE	63
Tabela 4 - Descrição do Requisito Criar Projeto da EFDE	71
Tabela 5 - Requisito - Carregar Imagens	73
Tabela 6 - Requisito - Calcular Orientação Exterior	76
Tabela 7 - Produtividade por tipo de linguagem	86
Tabela 8 - Complexidade ALI e AIE	89
Tabela 9 - Complexidade Entrada Externa	90
Tabela 10 - Complexidade Saída Externa	90
Tabela 11 - Complexidade Consulta Externa	90
Tabela 12 - Identificação das Funções do Tipo Dado	91
Tabela 13 - Identificação das Funções do Tipo Transação	92
Tabela 14 - Cálculo do Ponto de Função Não Ajustado para Ortoretificação	92
Tabela 15 - Cálculo do Ponto de Função Não Ajustado para os três modelos UML	93
Tabela 16 - Medidas Estatísticas do resultado da APF	94
Tabela 17 - Requisito - Criar Projeto	101
Tabela 18 - Requisito - Carregar Imagens	101
Tabela 19 - Requisito - Calcular Orientação Interior	102
Tabela 20 - Requisito - Calcular Orientação Exterior	102
Tabela 21 - Requisito - Extrair MNE	103
Tabela 22 - Requisito - Extrair Ortofoto	103
Tabela 23 - Requisito - <i>Stereo Plotter</i>	104
Tabela 24 - Requisito - Adicionar Feições	104
Tabela 25 - Requisito - Normalização	104
Tabela 26 - Pontos de Função não ajustados para o modelo UML	128
Tabela 27 - Pontos de Função não ajustados para o modelo UML Aspecto	129
Tabela 28 - Pontos de Função não ajustados para o modelo UML Padrões	130

SUMÁRIO

	INTRODUÇÃO	13
1	TÉCNICAS DE MODELAGEM DA ENGENHARIA DE SOFTWARE	18
1.1	Processo de Engenharia de Software	18
1.2	Modelagem UML	19
1.2.1	<u>Diagramas da UML</u>	20
1.2.1.1	Diagrama de Caso de Uso	20
1.2.1.2	Diagrama de Classe	23
1.2.1.3	Diagrama de Seqüência	26
1.3	Orientação a Aspectos	28
1.3.1	<u>AspectUML</u>	30
1.3.1.1	Estereótipo Aspectos	31
1.4	Padrões de Projeto	31
1.4.1	<u>Classificação dos Padrões</u>	33
1.4.2	<u>Padrão Singleton</u>	33
1.4.3	<u>Padrão Builder</u>	34
1.4.4	<u>Padrão Abstract Factory</u>	35
1.4.5	<u>Padrão Composite</u>	35
2	FUNDAMENTOS DA FOTOGRAMETRIA	38
2.1	Conceito de Fotogrametria	38
2.2	Breve Histórico	39
2.3	O processo fotogramétrico	41
2.3.1	<u>Definição do Projeto</u>	41
2.3.2	<u>Cadastramento dos Sensores</u>	42
2.3.3	<u>Cadastramento de Vãos Fotogramétricos</u>	43
2.3.4	<u>Inclusão de Imagens e Criação do Bloco Fotogramétrico</u>	44
2.3.5	<u>Orientação interior</u>	45
2.3.6	<u>Orientação Exterior</u>	49
2.3.7	<u>Normalização</u>	53
2.3.8	<u>Extração do MNE</u>	55
2.3.9	<u>Ortorretificação</u>	56
2.3.10	<u>Restituição Digital</u>	58
3	PROJETOS DA EFDE	61
3.1	Requisitos da EFDE	61
3.2	Modelo de Casos de Uso	64
3.2.1	<u>Casos de uso com esteriótipo <i>include</i></u>	66

3.2.2	<u>Caso de uso da Orientação Exterior</u>	66
3.2.3	<u>Caso de uso da Normalização</u>	67
3.2.4	<u>Caso de Uso da Ortorectificação</u>	67
3.3	Modelos de Classe	69
3.3.1	<u>Modelo de Classe com UML</u>	70
3.3.1.1	A classe Projeto	70
3.3.1.2	A hierarquia de classes Sensor	71
3.3.1.3	A classe Bloco de Imagem	71
3.3.1.4	A hierarquia da classe Ponto	72
3.3.1.5	A classe Ajustamento por Blocos	74
3.3.1.6	O modelo para restituição fotogramétrica, DEM e Ortorectificação	75
3.3.2	<u>Modelo Orientado a Aspectos</u>	77
3.3.2.1	Modelo de caso de uso orientado a Aspectos	77
3.3.2.2	Modelo de classes orientado a Aspectos	77
3.3.3	<u>Modelo de Classe com Padrões de Projeto</u>	81
3.3.3.1	A classe XML	81
3.3.3.2	A classe Sensor	82
3.3.3.3	A classe Feição	82
3.3.3.4	A modelagem de Pontos	83
4	ANÁLISE COMPARATIVA ENTRE OS MODELOS UML	85
4.1	Análise por ponto de função e caso de uso	85
4.1.1	<u>A contagem da APF</u>	86
4.1.1.1	Tipo de Contagem	87
4.1.1.2	Fronteira da aplicação	87
4.1.1.3	As funções	88
4.1.2	<u>Fator de Ajuste</u>	91
4.1.3	<u>Divisão do modelo UML para tipificação dos pontos de função</u>	91
4.2	Resultados da Análise	93
4.2.1	<u>Média e Desvio-padrão do resultado</u>	93
5	CONCLUSÕES	95
	REFERÊNCIAS	97
	APÊNDICE A – Requisitos Funcionais	100
	APÊNDICE B – Casos de Uso	105
	APÊNDICE C – Diagramas de Classes e Seqüência	110
	APÊNDICE D – Cálculo do Ponto de Função	127

INTRODUÇÃO

Apresentação

Todo sistema de informação para ser transformado em um sistema informatizado necessita de uma modelagem sistêmica para a identificação do conjunto de dados e funcionalidades pertinentes à solução dos problemas subjacentes. A modelagem é uma atividade essencial para qualquer sistema informatizado. Contudo, do ponto de vista prático, diversas aplicações, principalmente acadêmicas e/ou científicas, se iniciam sem o adequado trabalho de modelagem, produzindo, por vezes, software sem qualidade e, conseqüentemente, cujas utilidade e vida-útil são limitadas.

Dentro da área das Ciências da Computação, a Engenharia de Software tem por objetivo o desenvolvimento e organização de sistemas complexos, impondo regras e processos bem delineados para a solução de tais problemas. Existem várias definições para o termo Engenharia de Software e que, por vezes, transitam em caminhos distintos. Segundo Friedrich Ludwig Bauer ¹, "Engenharia de software é a criação e a utilização de sólidos princípios de engenharia a fim de obter software de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais".

Podem existir dúvidas sobre a distinção dos termos Engenharia de Sistemas e Engenharia de Software, entretanto, para Sommerville (2005) a Engenharia de Sistemas "se ocupa de todos os aspectos relacionados ao desenvolvimento de sistemas com base em computadores, incluindo hardware, software e engenharia de processos. A engenharia de software é parte desse processo".

Na Engenharia de Software, uma disciplina em particular merece uma maior atenção por parte dos profissionais: a Engenharia de Requisitos. De acordo com Leite (1994) a "Engenharia de Requisitos estabelece o processo de definição de requisitos como um processo no qual o que deve ser feito é elicitado, modelado e analisado. Esse processo deve lidar com diferentes pontos de vista, e usar uma combinação de métodos, ferramentas e pessoal.". De fato, a Engenharia de Requisitos é fundamental para o sucesso de um sistema e, por isso, são dedicadas inúmeras pesquisas nos métodos para a obtenção das informações necessárias à implementação de sistemas. Um método bastante pesquisado, disseminado e documentado é a Linguagem de Modelagem Unificada (UML²), que per-

¹ Friedrich Ludwig Bauer (10 de Junho de 1924, Regensburg) é um cientista da computação alemão e professor emérito na Universidade Munique de Tecnologia. Friedrich foi o autor da primeira definição de Engenharia de software, em 1969, na Conferência NATO Science Committee.

² UML é um acrônimo para Unified Model Language

mite, através de diagramas, expressar as necessidades do usuário.

Segundo Elmasri e Navathe (1994), a abordagem clássica de projetos de banco de dados divide a análise em três níveis de abstração: modelo conceitual, lógico e físico. No Modelo Conceitual as informações se apresentam de forma fiel a realidade observada. Já no modelo lógico as informações levam em conta a tecnologia na qual serão analisadas. E, por fim, o modelo físico apresenta as informações segundo uma estrutura rígida de armazenamento e utilização.

A presente dissertação, de forma análoga à definição de Elmasri e Navathe (1994), investiga e compara a aplicação de técnicas modernas da engenharia de software na modelagem conceitual de um software científico para fotogrametria digital. Além disso, procura valer-se de tais técnicas para a obtenção de software de qualidade. Nesse contexto, a análise e modelagem conceitual do sistema evidencia sua estrutura basal e permite que na fase de projeto de sistemas, o envolvimento entre os analistas e os desenvolvedores possa trazer benefícios à implementação. Os modelos lógico e físico, que foram deixados de fora do escopo da presente dissertação, podem ser objetos de futuros trabalhos acadêmicos.

Este trabalho se insere no contexto do projeto E-FOTO, tema da seção a seguir.

O projeto E-FOTO

O projeto E-FOTO (BRITO; COELHO, 2002a) visa à pesquisa, ao projeto e ao desenvolvimento de uma Estação Fotogramétrica Digital (EFD) de cunho educacional nos moldes do projeto GNU (STALLMAN, 1983). Portanto, atendendo às quatro liberdades defendidas pelo movimento do software livre³.

O projeto E-FOTO tem caráter interdisciplinar, envolvendo profissionais de computação e cartografia, tendo, portanto, subsidiado a maturação do conhecimento das disciplinas subjacentes ao desenvolvimento de uma EFD educacional, referida a partir de agora pelo acrônimo EFDE. Como resultado, a presente dissertação pôde produzir uma modelagem conceitual, englobando boa parte do conhecimento sobre fotogrametria digital disponível na literatura.

Em linhas gerais, fotogrametria é a ciência que objetiva a obtenção de informação geoespacial a partir de imagens obtidas por sensores remotos, em geral, transportados por aeronaves ou satélites. Contudo, até mesmo imagens de uma máquina fotográfica convencional podem ser utilizadas.

³ "Software Livre é uma questão de liberdade, não de preço. Software Livre respeita quatro liberdades essenciais: 0. de rodar o programa quando quiser; 1. de estudar o código fonte e modificá-lo para que faça o que você quiser; 2. de copiar o programa e distribuir as cópias quando quiser; 3. de publicar ou distribuir uma versão modificada quando quiser." (STALLMAN, 2008)

A fotogrametria digital, que é o foco do projeto E-FOTO, realiza o processo fotogramétrico por computador com base em imagens digitais ou digitalizadas.

Analisando as EFDs comerciais disponíveis no mercado, tais como Digi 3d⁴, DVP⁵ e Topol⁶, nota-se que são do tipo caixa-preta⁷. De fato, existem diversas soluções que realizam muito bem as etapas do processo fotogramétrico, sem contudo, demonstrar como as informações foram geradas. Na verdade, trata-se de uma estratégia de negócio que mantém muitos dos algoritmos em segredo. Os algoritmos fotogramétricos expressos no código fonte das EFDs comerciais ficam, desta forma, restritos a um grupo reduzido de pessoas. Neste ponto é preciso ser mencionado que este tipo de restrição ainda dificulta o aperfeiçoamento dos algoritmos, desfavorece a inovação e retarda o progresso científico. Além disto, devido à arquitetura fechada das EFDs proprietárias, não raramente, pesquisadores e estudantes encontram-se impossibilitados de incorporar às mesmas novos algoritmos que porventura estejam sendo desenvolvidos. Portanto, pode-se afirmar que a utilização de EFDs comerciais com fins didáticos e de pesquisa é demasiadamente limitada.

A argumentação anterior reforça o caráter inovador do projeto E-FOTO que disponibiliza em seu site (<http://www.efoto.eng.uerj.br>), sem custo e para qualquer fim, o código fonte da EFD Educacional, um livro sobre fotogrametria digital, a documentação técnica, a documentação de usuário, artigos, apostilas etc. O E-FOTO encontra-se em curso no programa de Pós-graduação em Engenharia de Computação, área de concentração Geomática, da Universidade do Estado do Rio de Janeiro.

A literatura apresenta poucos projetos com o intuito de mostrar à comunidade científica o *modus operandi* dessas soluções. A biblioteca de funções e classes para uso na fotogrametria UPTK (TOMMASELLI A.M.G.; HASEGAWA; GALO, 2005) é uma exceção e que traz ainda os algoritmos matemáticos e uma boa documentação.

Diferentemente da UPTK, que disponibiliza bibliotecas, o projeto E-FOTO vislumbra a construção de 100% de uma estação fotogramétrica digital, bastando para isso que o usuário baixe, no site do projeto, o código-fonte ou a aplicação já compilada e a execute em um computador pessoal. Esta dissertação possibilita que, em um futuro próximo, as classes e funções do E-FOTO sejam também disponibilizadas como bibliotecas para utilização em outros aplicativos.

A seção a seguir apresenta o estágio atual de desenvolvimento do E-FOTO.

⁴ <http://www.digi21.net>

⁵ <http://www.dvp.ca>

⁶ <http://www.topol.cz>

⁷ Expressão vinda da disciplina de Testes de Software, lecionada em cursos de Engenharia de Software, que significa não visível por dentro. Isto é, o usuário apenas visualiza o resultado final e não tem noção de como o resultado foi gerado.

O estágio atual do desenvolvimento do E-FOTO

Apesar dos algoritmos básicos de solução das etapas do processo fotogramétrico se encontrarem descritos na literatura, a forma de implementação e as funcionalidades necessárias para a interface do sistema precisam ser definidas em função das especificidades e dos objetivos acadêmicos do projeto E-FOTO. Atendendo à análise de riscos para o desenvolvimento da EFD educacional, foi tomada a decisão de iniciar o desenvolvimento pelos problemas específicos de cada etapa do processo fotogramétrico. Assim, para cada problema específico foi desenvolvido um módulo independente. No escopo do projeto E-FOTO foram desenvolvidos os seguintes módulos: Retificação de Imagens, Orientação Interior, Orientação Exterior, Fototriangulação, Normalização de Estereogramas, Restituição Fotogramétrica, Modelo Numérico de Elevações e Geração de Ortofotos.

Com o amadurecimento das soluções específicas foram construídas as condições para a compreensão e solução do problema da fotogrametria digital em sua totalidade. A partir de então, ficou definido que o software desenvolvido deveria ser integrado e que o E-FOTO deveria tornar-se uma EFD orientada a projetos fotogramétricos. Nesse contexto, um projeto fotogramétrico compreende desde a definição dos objetivos, passando pelo planejamento e realização da aquisição das imagens fotográficas de interesse e execução de diversas etapas do processo fotogramétrico, até o armazenamento e apresentação dos resultados de forma compreensível pelo usuário (SYSTEMS, 1994).

Para a integração dos módulos da EFD educacional se faz necessária a definição de um modelo conceitual que represente as entidades do domínio da fotogrametria digital e distribua as responsabilidades dentre as diferentes classes identificadas no domínio do problema. Essa é a principal motivação da presente dissertação, cujos objetivos são relacionados e discutivos na seção a seguir.

Deve ser enfatizado que a modelagem conceitual produzida para a presente dissertação foi concebida de forma genérica a fim de permitir sua utilização em outros projetos na área de conhecimento da fotogrametria digital.

Objetivos

A presente dissertação propõe, apresenta e compara três modelos conceituais, representados por diagramas de classe UML, para a camada de negócio da EFD educacional: UML, padrões de projeto e orientado a aspectos.

O presente estudo comparativo deseja reunir conclusões que dêem suporte a opção por uma dentre estas técnicas no tocante à análise e projeto de sistemas, em projetos científicos como E-FOTO.

Partindo do pressuposto que, ainda que com limitações, diagramas de classes UML

permitam a representação de conhecimento do domínio do problema, espera-se, também, contribuir para a difusão do conhecimento sobre fotogrametria digital dentre os conhecedores da UML.

Organização

A presente dissertação está organizada da seguinte forma. O capítulo a seguir traz a fundamentação teórica das técnicas de engenharia de software utilizadas na modelagem conceitual. O capítulo 3 aborda a teoria da fotogrametria, dentro do enfoque da análise e projeto orientado a objetos, seus processos e algoritmos matemáticos. No capítulo 4 são apresentadas as propostas de modelagem conceitual da Estação Fotogramétrica Digital Educacional (EFDE). No quinto capítulo é apresentado o estudo comparativo realizado entre as técnicas de modelagem utilizadas e, por fim, são apresentadas as conclusões e sugestões de trabalhos futuros.

1 TÉCNICAS DE MODELAGEM DA ENGENHARIA DE SOFTWARE

Este capítulo tem por objetivo descrever as técnicas de modelagem da Engenharia de Software que foram utilizadas para o desenvolvimento desta dissertação. Tais técnicas formaram a base de conhecimento em Engenharia de Software necessária para a modelagem de uma estação fotogramétrica digital baseada no paradigma da orientação a objetos. Três técnicas são descritas de forma didática e suscinta, equalizando os conhecimentos necessários para a compreensão desta dissertação.

1.1 Processo de Engenharia de Software

Para Matos (2002) a qualidade de um software não pode ser medida apenas por seu resultado, mas principalmente pelo processo de seu desenvolvimento. De fato, o processo de desenvolvimento utilizado poderá determinar o sucesso ou o fracasso de um projeto. Tão importante é a escolha de um processo de desenvolvimento que existem disciplinas especializadas em estudar novas formas e processos, que se adequem às diversas necessidades.

Atualmente, as empresas de tecnologia que desenvolvem software já podem ser “medidas” com relação a qualidade de software que produzem, de acordo com metodologias adotadas, como é o caso do CMMI ⁸, criado pelo Instituto de Engenharia de Software (SEI), da Carnegie-Mellon University (CMSEI, 2008). O CMMI provê às empresas uma série de normas e procedimentos que visam à melhoria constante do processo de desenvolvimento de software.

Os sistemas informatizados de cunho acadêmico/científicos nem sempre são frutos de um processo adequado e que, por vezes, inviabiliza sua utilização em larga escala. Para o projeto da EFDE, busca-se utilizar um processo já bem estabelecido e documentado e por isso optou-se pelo modelo incremental (ROYCE, 1970). Esse processo possui estrutura rígida e bem delineada, permitindo uma evolução gradual e com objetivos definidos. Tal escolha também baseou-se no fato da EFDE ser um sistema de cunho científico-acadêmico e dessa dissertação ter como principal objetivo a modelagem conceitual e não a implementação da estação.

⁸ CMMI é um acrônimo de Capability Maturity Model Integration, que em português significa Modelo Integrado de Maturidade de Capacitação

1.2 Modelagem UML

A UML ⁹ foi elaborada a partir da cooperação entre três pesquisadores (OMG, 2007): Grady Booch, James Rumbaugh e Ivar Jacobson. Unindo suas pesquisas em programação orientada-a-objetos, modelagem e engenharia de software, propuseram à comunidade técnico-científica uma nova abordagem em modelagem de sistemas. Essa proposta tem por objetivo o entendimento da análise tanto pelos especialistas em sistemas quanto pelos usuários, os quais transmitem suas necessidades e demandas. Esse entendimento se faz através de convenções simbólicas em um nível de abstração próximo ao mundo real. Sommerville (2005) descreve que “(...) A linguagem de modelagem unificada (UML), utilizada nesse método unificado, tornou-se um padrão efetivo para a modelagem de objetos (...)”. De fato, a UML passou a ser a ferramenta utilizada para a modelagem de sistemas orientado a objetos.

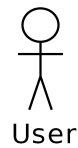
Esta metodologia de Engenharia de Software é composta de uma série de diagramas visuais que facilitam o entendimento do sistema a ser desenvolvido. Por outro lado, a substituição das especificações meramente descritivas por uma fortemente gráfica, permite aos especialistas uma análise mais rápida e fiel a realidade transmitida pelos usuários. Esses, também são beneficiados por essa nova técnica, uma vez que é bem mais fácil entender um diagrama do que simplesmente entender a descrição feita pelos especialistas e que, na maior parte das vezes, vem carregada de jargões próprios da área de tecnologia.

Atualmente a UML é uma marca registrada da OMG ¹⁰, que realiza constantes atualizações na documentação oficial. De acordo com Matos (2002), a proposta feita pela OMG divide os diagramas da UML em cinco níveis de abstração: (1) funções do sistema e seus respectivos responsáveis, (2) estruturas de informação, (3) comportamento e dinamismo das classes, (4) estados das execuções e (5) processo de implementação e implantação. Nesta dissertação, são apresentados os três primeiros níveis, por se tratar de um modelo conceitual.

⁹ UML é um acrônimo para Unified Modeling Language, traduzida para o português significa Linguagem de Modelagem Unificada

¹⁰ OMG é um acrônimo de Object Management Group, que em português significa grupo de gerenciamento de objetos. Um grupo que reúne as principais empresas mundiais de tecnologia.

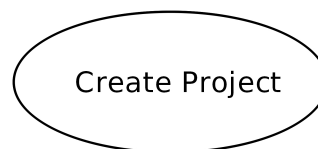
Figura 1 - Símbolo que representa um Ator



Legenda: O símbolo representa o ator *User*.

Fonte: O autor, 2008.

Figura 2 - Símbolo que representa um Caso de Uso



Legenda: O símbolo representa o caso de uso *Create Project*.

Fonte: O autor, 2008.

1.2.1 Diagramas da UML

1.2.1.1 Diagrama de Caso de Uso

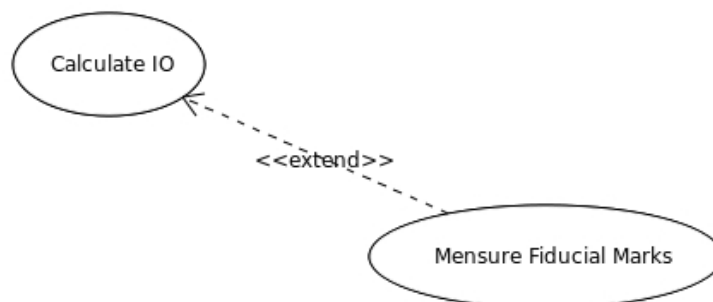
Os diagramas de caso de uso expressam claramente os papéis e funções de cada uma das partes envolvidas e as atividades realizadas pelo sistema. Os requisitos do sistema são modelados nesse diagrama, que é composto basicamente de três elementos: atores, casos de uso e relacionamentos.

Atores representam os usuários, o próprio sistema ou sistemas externos que possuam alguma interação com o sistema modelado. A Figura 1 mostra a convenção simbólica para atores, definido pela UML.

Casos de Uso representam os papéis ou funcionalidades de cada ator dentro do sistema. O símbolo de caso de uso está representado na Figura 2.

Os casos de uso se relacionam com os atores e entre si, através de relacionamentos

Figura 3 - Exemplo de um relacionamento do tipo *extend*



Legenda: Representação do relacionamento *extend* entre dois casos de uso.

Fonte: O autor, 2008.

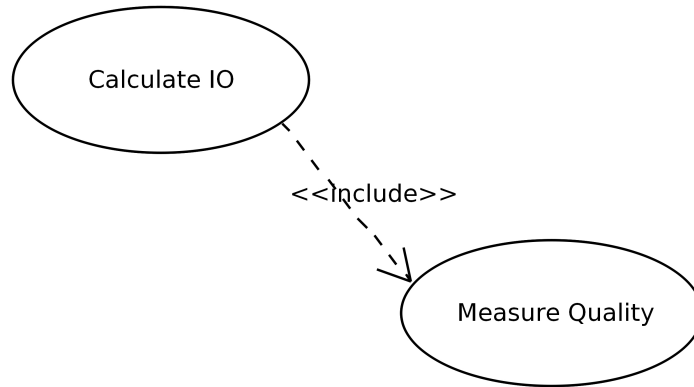
específicos e estereotipados ¹¹. Basicamente, a UML, propõe dois tipos de relacionamentos estereotipados entre casos de uso: *extend* e *include*. No primeiro, o caso de uso utilizará, opcionalmente, a funcionalidade de outro caso de uso, estendendo sua própria funcionalidade - como se dele fosse. Observando a Figura 3 pode-se notar que o caso de uso *Calculate IO* (Calcular Orientação Interior) pode estender suas funcionalidades, abrangendo o caso de uso *Measure Fiducial Marks* (Medição de Marcas Fiduciais).

No estereótipo *include*, o caso de uso incluirá a funcionalidade de outro caso de uso. Na realidade, essa funcionalidade deveria estar dentro do caso de uso, mas por ser utilizada diversas vezes dentro do sistema, o analista deve separá-la em um outro caso de uso, utilizando, desta maneira, o reuso de funcionalidades. A Figura 4 demonstra que o caso de uso *Measure Quality* (Medir Qualidade) é sempre utilizado pelo caso de uso *Calculate IO* (Calcular Orientação Interior).

Relacionamentos definem as relações entre os atores e entre esses e os casos de uso. Um relacionamento pode ser bidirecional ou unidirecional, conforme o caso específico da modelagem. A Figura 5 exemplifica um relacionamento bidirecional entre o caso de uso *Calculate DTM* e o ator *User* (Usuário).

¹¹ Segundo Matos (2002), “(...) as ferramentas de desenho de UML tratam os estereótipos como regras sintáticas rígidas. Sua presença inclusive indica que no código-fonte uma ação deve ser feita (...)”

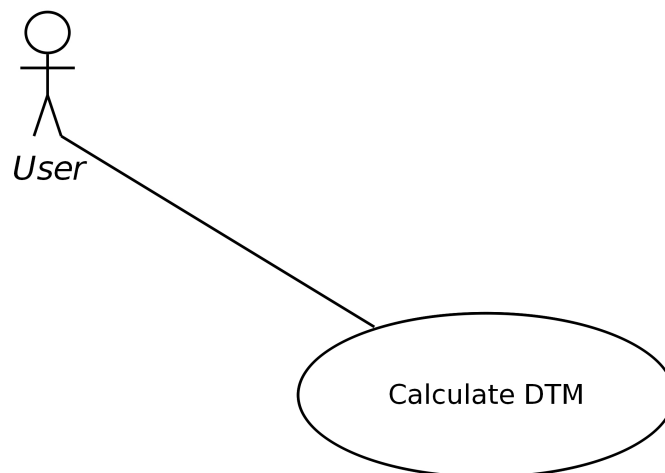
Figura 4 - Exemplo de um relacionamento do tipo *include*



Legenda: Representação do relacionamento *include* entre dois casos de uso.

Fonte: O autor, 2008.

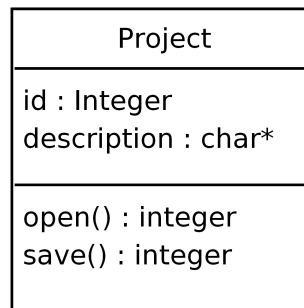
Figura 5 - Símbolo do relacionamento



Legenda: A figura demonstra um relacionamento entre o ator *User* e o caso de uso *Calculate DTM*.

Fonte: O autor, 2008.

Figura 6 - Símbolo de uma classe



Legenda: O símbolo representa a classe *Project*, com seus atributos e responsabilidades.

Fonte: O autor, 2008.

1.2.1.2 Diagrama de Classe

As classes são representações da abstração de entidades do domínio do problema. Uma classe expressa a semântica dos atributos e responsabilidades (métodos) da entidade. A Figura 6 demonstra o símbolo de uma classe para a UML. Pode-se observar que a representação gráfica se dá pelo desenho de um retângulo, dividido em três compartimentos: o nome da classe, os atributos e as responsabilidades.

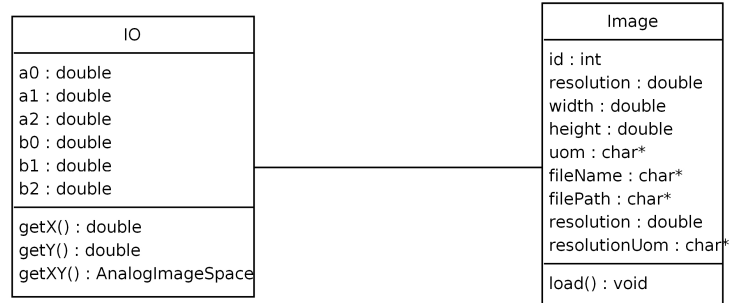
O diagrama de classes é uma ferramenta da UML que representa estaticamente as classes de um sistema, definindo uma modelagem baseada em classes (OMG, 2007), apresentando os relacionamentos existentes entre as classes, criando, assim, relações lógicas entre as entidades. A OMG (2007) descreve os seguintes relacionamentos:

Associação representa uma relação semântica que pode ocorrer entre duas entidades (OMG, 2007).

A Figura 7 exhibe um relacionamento do tipo Associação entre as classes *Image* (Imagem) e *IO* (Orientação Interior).

A OMG (2007) ainda descreve as associações do tipo Agregação/Composição, que representam uma conexão do tipo parte-todo. Na Figura 8 é apresentada uma pequena parte de um diagrama de classes que demonstra as agregações e composições. Os losangos preenchidos formam composições, enquanto que os losangos vazados exemplificam agregações. Todas as relações apresentadas são do tipo parte-todo, entretanto, as composições apresentam uma semântica mais forte que as agregações, representando que a parte não existe sem o todo. Este é o caso das classes *MatricialBlock* (Bloco Matricial) e *TransversalStrip* (Faixa Transversal) que não podem ser instanciadas fora do contexto de um objeto da classe *Block* (Bloco).

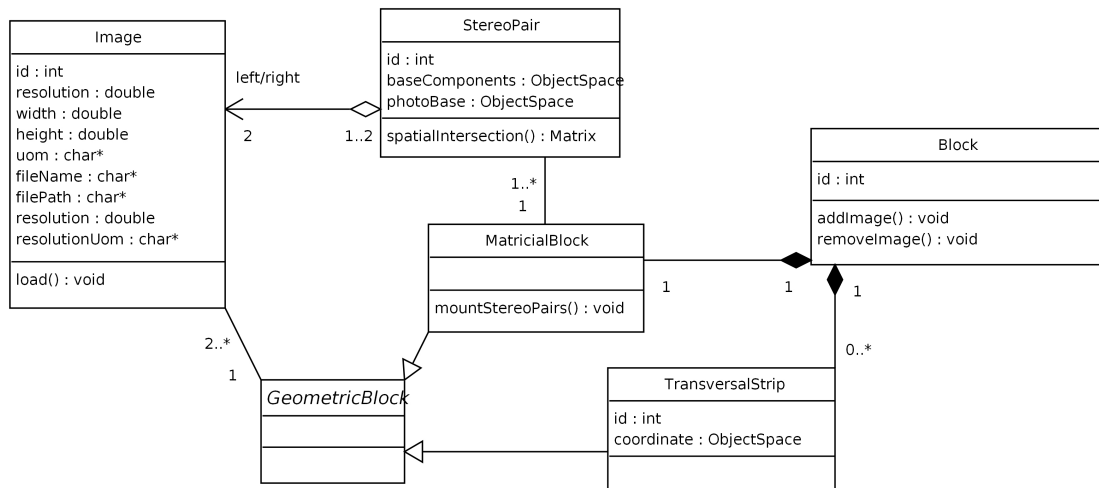
Figura 7 - Associação entre classes



Legenda: A figura representa a associação entre as classes *IO* e *Image*.

Fonte: O autor, 2008.

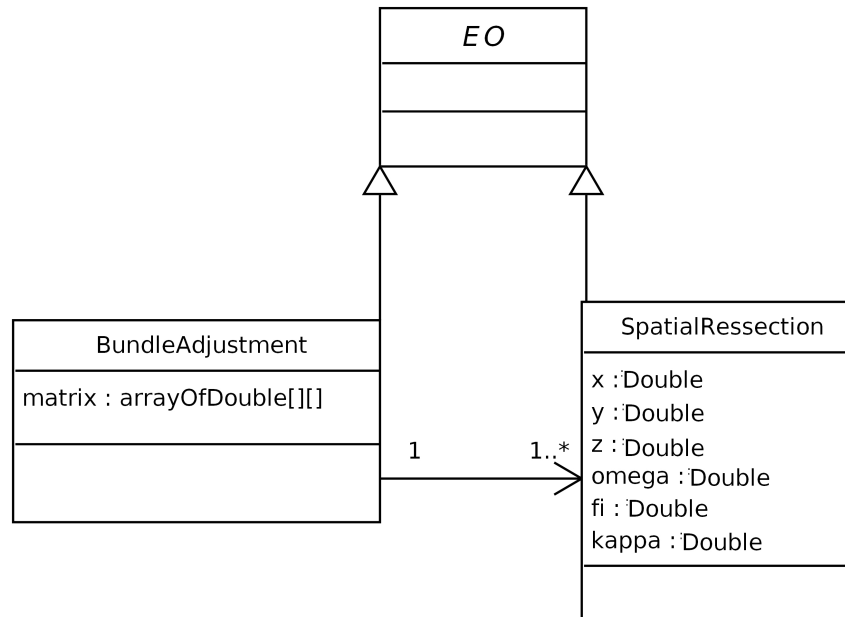
Figura 8 - Classes associadas com Agregação e Composição



Legenda: As classes *Image* e *StereoPair* se relacionam através de agregação, enquanto que as classes *Block*, *MatricialBlock* e *TransversalStrip* se relacionam através de composição.

Fonte: O autor, 2008.

Figura 9 - Generalização de uma classe



Legenda: As classes *BundleAdjustment* e *SpatialResection* são especializações da classe geral *EO*.

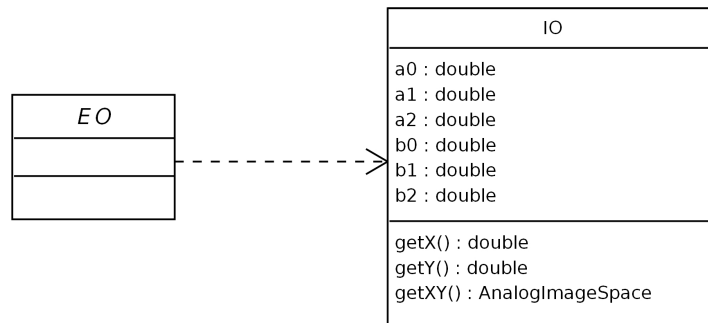
Fonte: O autor, 2008.

Generalização representa uma hierarquia entre classes, onde os atributos e/ou métodos podem ser herdados de uma classe para outra. A Figura 9 exibe dois relacionamentos do tipo generalização, *SpatialResection* (Resseção Espacial) e *BundleAdjustment* (Ajustamento por feixe) são herdados de *EO* (Orientação Exterior).

Dependência representa uma conexão forte entre instâncias de uma classe e outra, de forma que uma é dependente da outra. Além disso, uma modificação na classe que causa dependência pode demandar a alteração da classe dependente. Observando a Figura 10 verifica-se uma ligação por dependência entre as classes, ou seja, a classe *EO* depende de *IO*.

Estereótipo é um mecanismo que permite a extensão da linguagem através da criação de elementos semânticos de acordo com as características do problema. O estereótipo pré-definido *abstract*, na especificação da UML(OMG, 2007), representa uma classe abstrata, tipo de classe que não possui nenhum método implementado, nem possibilita a instanciação de objetos, podendo conter somente atributos e assinaturas de métodos. Wiener e Pinson (2000) ainda mencionam o fato de que classes abstratas “(...) são frequentemente definidas no topo de uma estrutura hierárquica de classes (...)”. Uma classe concreta, por outro lado, é aquela que contém métodos implementados.

Figura 10 - Dependência entre classes



Legenda: A classe *EO* depende da classe *IO*.

Fonte: O autor, 2008.

Ainda está descrito na especificação da UML (OMG, 2007) a classe com o estereótipo denominado *interface*. Wiener e Pinson (2000) descrevem como “ (...) um caso especial de classe abstrata também utilizada para estabelecer características comportamentais que devem ser realizadas por uma classe concreta que implemente a interface (...)”. Na prática, uma classe de interface possui apenas assinaturas de métodos, não podendo conter propriedades e nem implementação. Toda classe de interface necessita de uma classe concreta que a implemente.

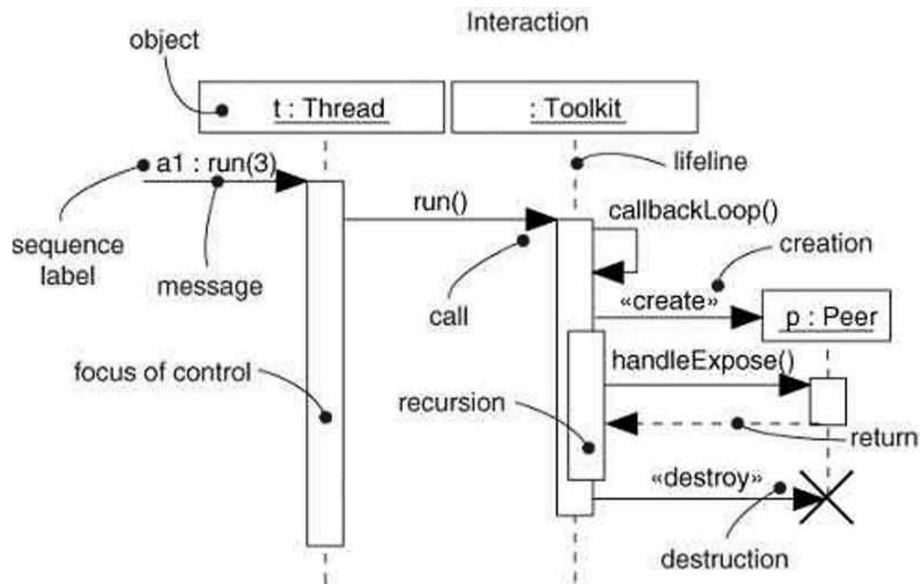
Da mesma forma que o diagrama de caso de uso pode determinar o sucesso de um sistema, o diagrama de classes pode determinar a eficiência desse sistema. Esse diagrama é considerado por seus autores como sendo o diagrama base de toda a UML, em virtude dessa ser baseada fortemente em orientação a objeto (BOOCH G.; RUMBAUGH; JACOBSON, 2006).

1.2.1.3 Diagrama de Seqüência

Dos diagramas até agora apresentados, esse é o único diagrama dinâmico, ou seja, ele representa o comportamento do sistema durante a realização de um caso de uso, exibindo o dinamismo do sistema, com suas interações, entradas e saídas. Esse diagrama é baseado nos dois anteriores e demonstra como as instâncias¹² das classes estão interagindo entre si, com o sistema e o tempo de duração dessa interação. Através do diagrama de

¹² A instância de uma classe, também conhecida como objeto, é a concretização da abstração que uma classe representa.

Figura 11 - Artefatos do Diagrama de Sequência



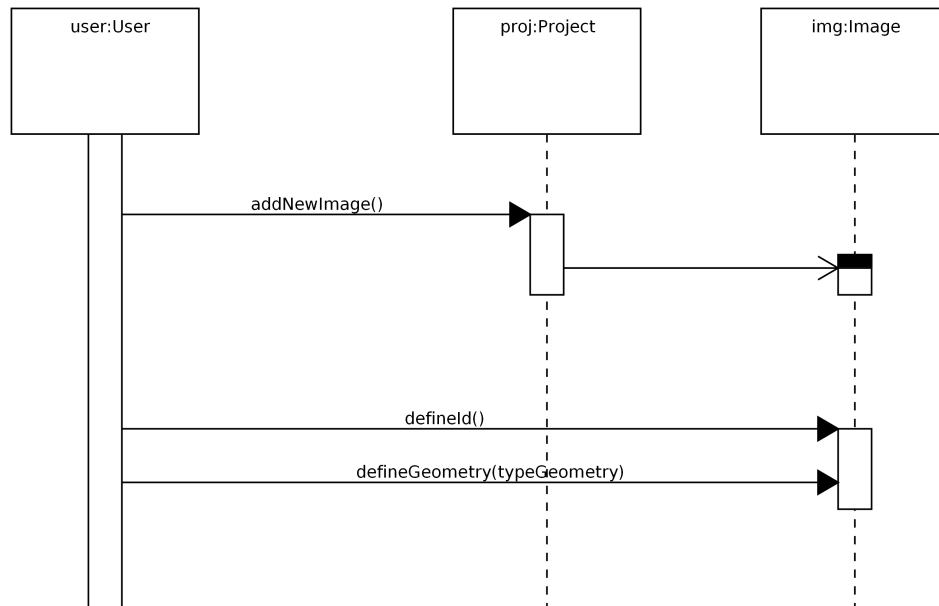
Fonte: O autor, 2008.

seqüência, o analista pode validar os diagramas de classe e casos de uso.

Artefatos são elementos constantes de um diagrama da UML. Para o diagrama de seqüência pode-se ter os artefatos constantes da Figura 11, explicados abaixo.

- Objeto (object) - é a instância de uma classe. No diagrama a sintaxe utilizada exhibe o nome do objeto, dois pontos e o nome da classe instanciada (`t:Thread`);
- Linha do tempo (lifeline) - demonstra o tempo de vida de um objeto, ou seja, o tempo entre sua criação e destruição;
- Mensagem (message) - é a troca de mensagens existentes entre os objetos, normalmente através de métodos.
- Chamada (call) - é quando um método de uma classe realiza uma requisição a outra classe. Uma chamada pode ser assíncrona ou síncrona;
- Criação (creation) - é uma chamada para uma criação de uma instância;
- Destruição (destruction) - é uma chamada para a destruição de uma instância;
- Retorno (return) - é quando a classe requisita responde à classe requisitante;
- Recursão (recursion) - é quando a classe realiza uma chamada a um método próprio;
- Foco do Controle (focus of control) - simboliza o tempo em que o processamento estará ativo no objeto.

Figura 12 - Diagrama de Sequência



Legenda: Uma representação do Diagrama de Sequência que mostra a adição de uma nova imagem ao projeto fotogramétrico.

Fonte: O autor, 2008.

Na Figura 12 é exemplificado um diagrama de seqüência para a inclusão de uma imagem.

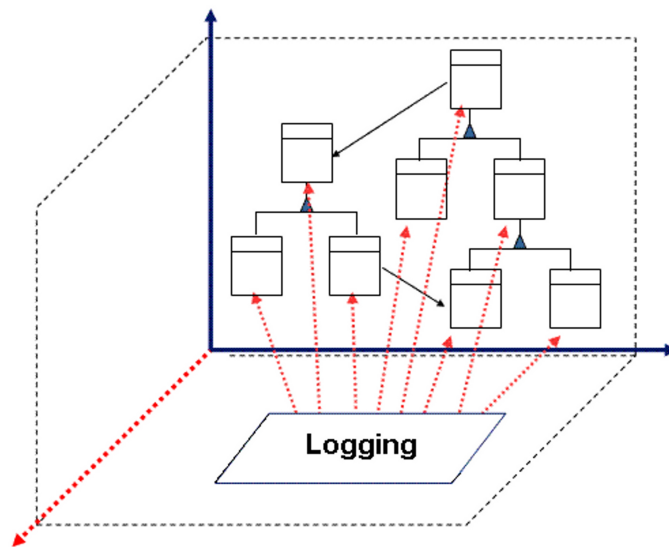
1.3 Orientação a Aspectos

O conceito do paradigma da orientação a aspectos (POA) está fortemente ligado ao conceito de *crosscutting concerns*, termo de origem inglesa que pode ser traduzido por problemas ou interesses transversais.

Todo o sistema ao ser elaborado passa por uma fase chamada de elicitação de requisitos. Nessa fase, o analista realiza todo o levantamento das informações importantes para a construção desse sistema. Por diversas vezes, os requisitos estão ligados ao negócio e, por outras, estão relacionados ao ambiente no qual o sistema será executado. São denominados como requisitos funcionais e não funcionais, respectivamente (SOMMERVILLE, 2005; PRESSMAN, 1995).

Ao se analisar profundamente os requisitos, pode-se notar que alguns itens são, constantemente, repetidos e/ou reutilizados. O paradigma da orientação a objetos provê recursos para que não haja esforço duplicado e para que as classes possam ser utilizadas

Figura 13 - Interesse transversal sem uso de POA



Fonte: O autor, 2008.

em diversas partes do sistema. Entretanto, algumas propriedades transversais não são facilmente modeladas pela orientação a objetos (SEINTURIER; RETAILL, 2005). Essas propriedades são espalhadas ao longo de todo o sistema, dificultando a manutenção posterior.

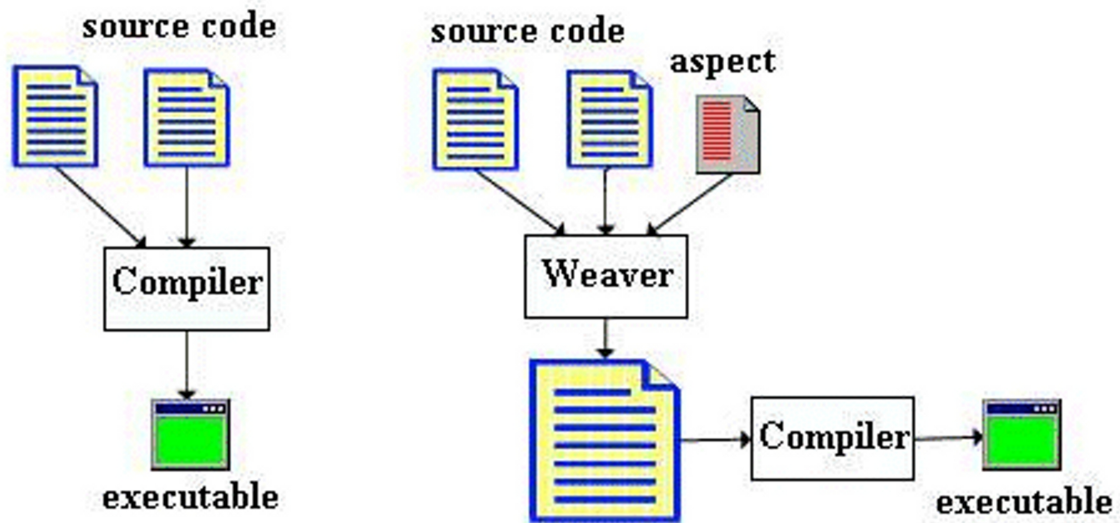
Diversas técnicas como *Multi-dimensional Separation of concerns*, *Adaptive Programming*, *Composition Filters* etc (ARAUJO et al., 2002) foram desenvolvidas para solucionar o problema dos interesses transversais. A orientação a aspectos é uma dessas técnicas que provê meios para a decomposição do sistema por áreas de interesses, através de entidades denominadas aspectos (SEINTURIER; RETAILL, 2005). De acordo com Laddad (2003), os interesses de um sistema podem ser classificados em:

- Interesses do negócio - são os requisitos de negócio propriamente dito;
- Interesses em nível de sistema - são os requisitos periféricos, que atravessam múltiplos módulos, como por exemplo, persistência de dados e *logging*¹³.

Pode-se notar pela Figura 13 que o módulo de *login* “atravessa” todo o sistema, sendo considerado um interesse transversal. A separação de interesses e a modularização é fundamental para se diminuir os seguintes problemas:

¹³ Módulo de um sistema que registra trilhas de auditorias.

Figura 14 - Combinador de Aspectos



Fonte: <http://java-intvw-ques.blogspot.com.br/2013/08/mix-topics.html>

- Código entrelaçado (*code tangling*) - quando um determinado módulo possui chamadas simultâneas a vários interesses transversais, por exemplo, persistência e *logging*;
- Código espalhado (*code scattering*) - quando um interesse transversal é implementado em múltiplos módulos.

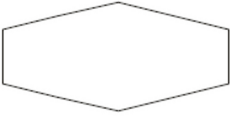
De acordo com Kiczales et al. (1997), uma implementação em aspectos consiste nos seguintes elementos:

- Linguagem de componentes - implementa os interesses do negócio, por exemplo, Java;
- Linguagem de aspecto - deve suportar a implementação de interesses transversais, por exemplo, AspectJ;
- Combinador de aspectos (*aspect weaver*) - realiza a ligação entre as duas linguagens. A Figura 14 exemplifica a utilização do combinador de aspectos, onde se pode observar as duas implementações: na esquerda, sem a utilização de Aspectos; na direita, com a utilização de Aspectos e a função do Combinador de aspectos.

1.3.1 AspectUML

Para Catajy (2006), a AspectUML é “(...) uma extensão para a UML que possibilita a representação dos aspectos e a completa documentação de suas interações com as

Tabela 1 - Modelagem de uma classe de Aspectos

Elemento	Estereótipo	Diagrama	Descrição
Classe	<<aspecto>>		Representa um aspecto

classes do núcleo do sistema (...)". De fato, faltava ao paradigma da orientação a aspectos uma linguagem descritiva de suas funcionalidades, permitindo o analista/projetista documentar o sistema. A AspectUML foi elaborada a partir da extensão da UML através dos mecanismos extensíveis já existentes como: estereótipos (*stereotypes*), rótulos (*tags*) e restrições (*constraints*) (CATAJY, 2006).

1.3.1.1 Estereótipo Aspectos

Os aspectos são estereótipos para os elementos **classe** da UML, podendo também ser abstrato ou concreto, assim como as classes. Um rótulo `«abstrato»` é usado para identificar um aspecto abstrato. O aspecto pode ser ainda rotulado como ativo ou passivo, identificando quando um aspecto afeta ou não um comportamento de uma classe do núcleo do sistema, respectivamente (CATAJY, 2006).

Na tabela 1 estão os elementos que representam, na UML, os aspectos.

A partir dessa definição, pode-se montar os diagramas da UML com a utilização do símbolo apresentado na tabela 1. Na Figura 15 é demonstrada a funcionalidade de *logging* com o uso de aspecto.

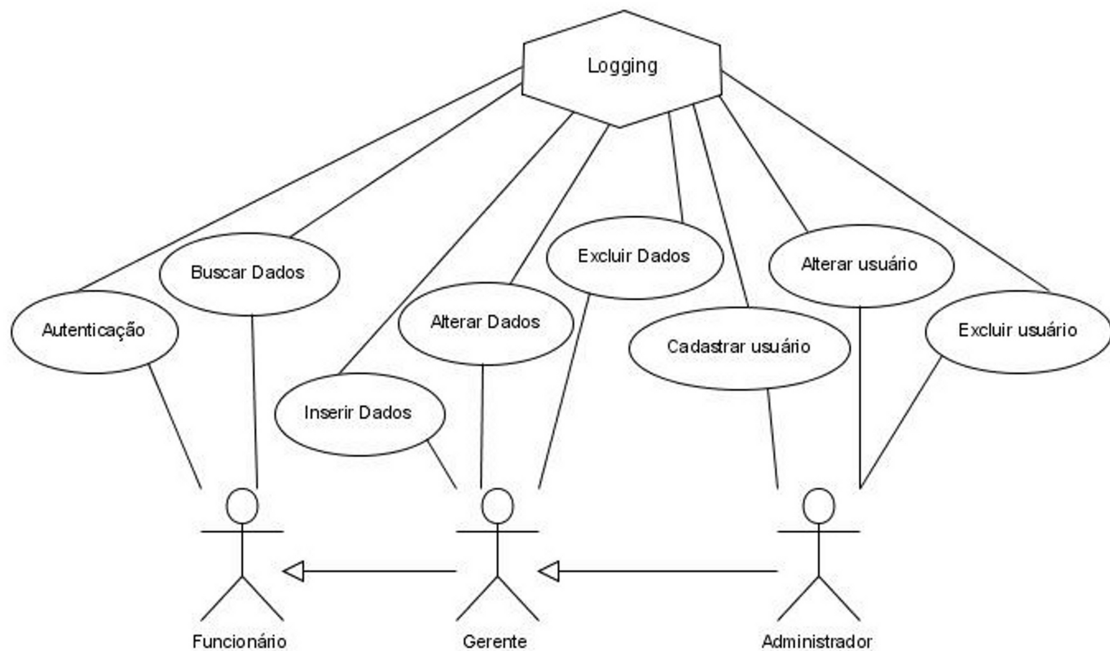
Outros estereótipos foram criados para a completa definição de aspectos utilizando-se a UML. Catajy (2006) descreve esses estereótipos, mostrando exemplos práticos de sua implementação. A partir dessas especificações pode-se modelar todo o sistema (diagrama de caso de uso e diagrama de classes) segundo a perspectiva da POA.

1.4 Padrões de Projeto

Um dos principais conceitos da orientação a objetos é o encapsulamento de uma classe. Para Pressman (1995), o encapsulamento é "(...) um conceito que propõe a criação de objetos de modo a serem autossuficientes procurando ter uma interação mínima com outros objetos na aplicação (...)". Desta forma, a orientação a objetos promove o reuso de objetos dentro de uma aplicação e até mesmo entre aplicações distintas.

Nesse contexto, surgiram os padrões de projeto orientados a objetos, soluções reuti-

Figura 15 - *Logging* com o uso de Aspecto



Fonte: O autor, 2008.

lizáveis em diversas aplicações. Na realidade, o conceito de padrão surgiu com o arquiteto Christopher Alexander (ALEXANDER, 1964; ALEXANDER, 1979) para definir padrões de construções e cidades. As características mencionadas por Christopher formam a base dos padrões de projeto orientados a objetos.

O conceito básico para padrões é simples: são classes ou objetos que se relacionam, criando um núcleo para a solução de um problema, podendo ser utilizado em diversos contextos. Todos os padrões são rigidamente testados e avaliados, portanto, o emprego de padrões de projeto traz como benefício a qualidade de código e, quando bem empregados, facilitam a manutenção posterior. Christopher Alexander afirma: “(...) cada padrão descreve um problema no nosso ambiente e o cerne da sua solução, de tal forma que você possa usar essa solução mais de um milhão de vezes, sem nunca fazê-lo da mesma maneira (...)”.

Em 1995, um grupo de quatro pesquisadores denominados GoF¹⁴ (GAMMA; HELM et al., 2000) tornou célebre a aplicação de padrões ao publicarem um livro contendo 23 padrões reutilizáveis.

¹⁴ GoF é um acrônimo para Gang of Four, que em português significa Gangue dos Quatro

1.4.1 Classificação dos Padrões

Os padrões de projeto são classificados quanto à finalidade e quanto ao escopo. O primeiro demonstra o que o padrão realiza, enquanto que o segundo diz respeito a onde o padrão será aplicado.

Como descrito por Gamma, Helm et al. (2000), os padrões podem ser classificados quanto à finalidade da seguinte forma:

- criação - preocupação com o processo de criação dos objetos;
- estrutural - preocupação com a composição das classes ou objetos;
- comportamental - preocupação com o comportamento e interações entre classes ou objetos.

O segundo critério de classificação é quanto ao escopo, que pode ser objeto ou classe, dependendo de onde o padrão será aplicado primeiramente. Os padrões para classes trabalham com os relacionamentos entre as classes, sendo, portanto, considerados estáticos, ou seja, não podem ser alterados em tempo de execução. É isso que os diferencia dos padrões para objeto, pois nesses, os relacionamentos são definidos no nível de objetos e estão sujeitos a alteração em tempo de execução, permitindo um maior dinamismo entre os objetos (GAMMA; HELM et al., 2000).

As próximas sub-seções descrevem exclusivamente os padrões utilizados na modelagem conceitual da EFDE. Cabe salientar que nas fases que se seguirão, principalmente na fase de projeto, outros padrões podem ser identificados e utilizados pela equipe de desenvolvimento do E-FOTO.

1.4.2 Padrão Singleton

O padrão Singleton é classificado como padrão de criação de objeto. Apesar de simples, esse padrão, implementa uma funcionalidade de grande importância para diversas aplicações. Ele garante que uma determinada classe possua uma e apenas uma instância durante todo o tempo de execução do sistema. Isso faz com que não se tenha duas instâncias de uma mesma classe, competindo na mesma aplicação (GAMMA; HELM et al., 2000).

Esse é um padrão muito utilizado, por exemplo, em classes que controlam conexões com banco de dados, permitindo que apenas uma classe gerencie as conexões.

O padrão se baseia nas seguintes características:

- a classe deverá ter um construtor privado;

Figura 16 - Padrão Singleton

Singleton
- instance : Singleton = null
+ getInstance() : Singleton
- Singleton() : void

Fonte: (GAMMA; HELM et al., 2000)

- a classe deverá ter um método público para disponibilizar a instância;
- caso a instância já exista, o método retorna essa instância, caso contrário, cria uma nova e alimenta um contador de instância.

Desta forma, a classe garante que terá tantas instâncias quanto ela mesmo permitir. Por exemplo, uma classe que controle conexões com o banco de dados pode permitir até 10 conexões, ou uma classe que controle o *logger* de uma aplicação só permitirá uma única instância. A Figura 16 representa a modelagem de uma classe com o padrão Singleton aplicado a ela.

1.4.3 Padrão Builder

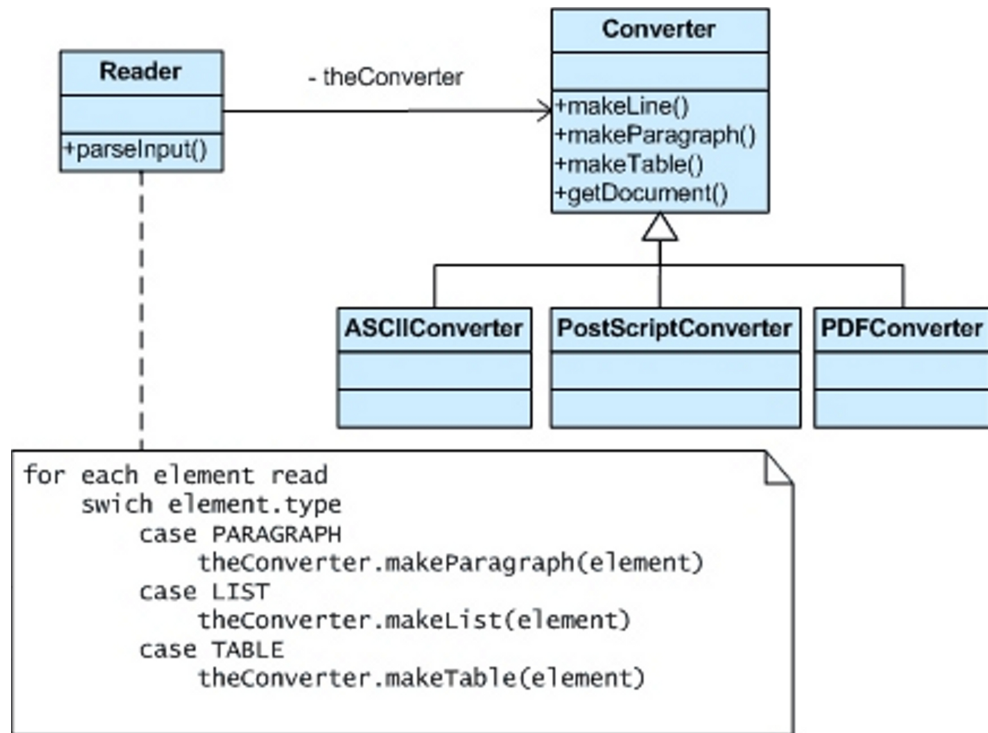
O padrão Builder é classificado como padrão de criação de objeto. Este padrão "(...) separa a construção de um objeto complexo da sua representação de modo que o mesmo processo de construção possa criar diferentes representações (...)"(GAMMA; HELM et al., 2000).

Talvez o exemplo mais prático seja o próprio descrito pelo GoF. Um leitor de arquivos do tipo RTF¹⁵ pode exportar o arquivo para outros formatos como por exemplo texto puro ou HTML¹⁶. Desta forma, o padrão implementa uma interface abstrata (Builder) que possui uma associação com o leitor (Reader). A conversão é realizada pelas classes concretas da interface. (GAMMA; HELM et al., 2000)

¹⁵ RTF é um acrônimo para Rich Text Format, traduzindo para o português, significa Formato de Texto Rico.

¹⁶ HTML é um acrônimo para HypeText Markup Language, que significa Linguagem de Marcação de HiperTexto.

Figura 17 - Padrão Builder



Fonte: (GAMMA; HELM et al., 2000)

1.4.4 Padrão Abstract Factory

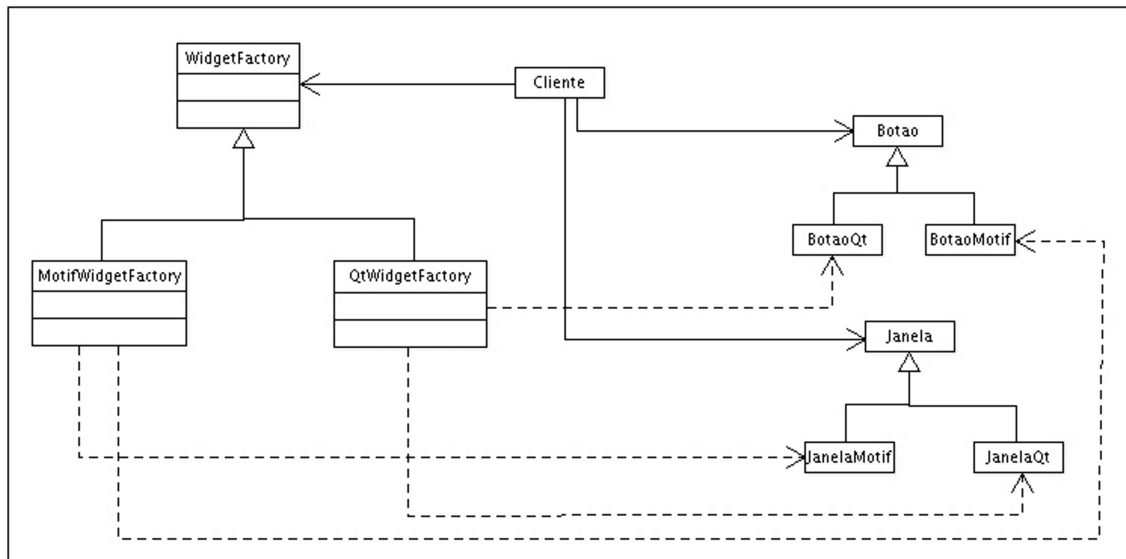
O padrão Abstract Factory é classificado como padrão de criação de objeto. Este padrão, segundo Gamma, Helm et al. (2000), "(..)fornece uma interface para a criação de famílias de objetos relacionados ou dependentes sem especificar suas classes concretas", isto é, o cliente não precisa conhecer uma determinada classe concreta para instanciá-la, ao invés disso, esse padrão utiliza uma classe abstrata que provê acesso às inúmeras classes concretas.

Um bom exemplo para o uso deste padrão são as interfaces que suportam múltiplas interações, também conhecidas como *look-and-feel*. No exemplo da Figura 18, toda vez que o cliente necessitar de um botão, ao invés de solicitar diretamente a classe concreta específica, ele irá solicitar a fábrica (WidgetFactory) um botão que será fornecido de acordo com o tipo de interface escolhida anteriormente (Motif ou Qt).

1.4.5 Padrão Composite

O padrão Composite é classificado como padrão de estrutura de objeto. De acordo com a definição dada por Gamma, Helm et al. (2000), esse padrão permite a criação de

Figura 18 - Padrão Abstract Factory



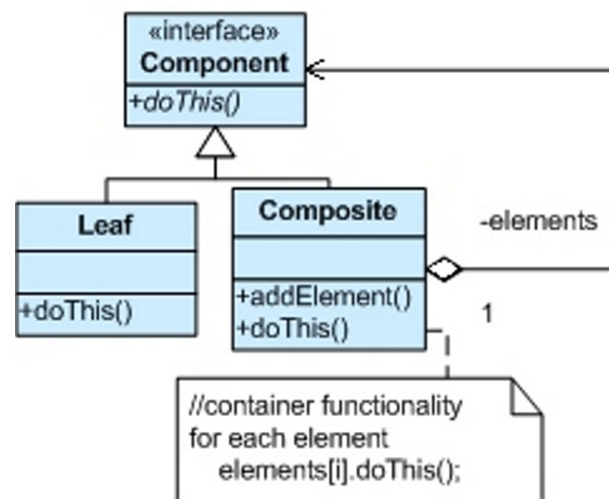
Fonte: (GAMMA; HELM et al., 2000)

estruturas em árvores para representar uma hierarquia do tipo partes-todo. Isso permite que os clientes tratem de maneira uniforme objetos individuais e suas composições, conforme pode ser observado na Figura 19.

O exemplo ilustrado pelo próprio GoF é de uma interface gráfica, onde os elementos individuais como linha, retângulo, texto, etc., são partes de um elemento gráfico. Além desses elementos individuais, o gráfico pode conter ainda composições desses, como se fosse uma figura.

Este capítulo apresentou as técnicas de Engenharia de Software empregadas ao longo desta dissertação. O próximo capítulo é dedicado ao cerne da EFDE, ou seja, a fotogrametria digital.

Figura 19 - Padrão Composite



Fonte: (GAMMA; HELM et al., 2000)

2 FUNDAMENTOS DA FOTOGRAMETRIA

Este capítulo descreve, do ponto de vista da análise orientada a objetos, os fundamentos da fotogrametria digital, conteúdo imprescindível para a compreensão da modelagem da Estação Fotogramétrica Digital e Educacional (EFDE). Para tanto, inicialmente, será apresentado o conceito de fotogrametria, um breve histórico da fotogrametria e o processo fotogramétrico digital. Em seguida, são apresentados os principais detalhes das etapas que compõem o processo fotogramétrico digital. A descrição é realizada sob a ótica da análise orientada a objetos, sendo assim, agrega versões preliminares dos diagramas de classe.

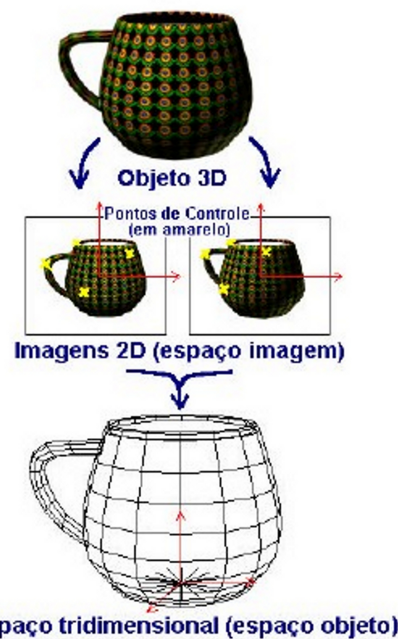
2.1 Conceito de Fotogrametria

De acordo com Informática (2008), fotogrametria significa "(...) medição, numa perspectiva fotográfica, das distâncias e das dimensões reais dos objetos". Uma definição com fortes raízes etimológicas, pois conforme mencionado no E-Book, livro digital sobre fotogrametria (BRITO; COELHO, 2002b): "photon - luz, graphos - escrita, metron - medições". Portanto, a fotogrametria visa à obtenção de medidas confiáveis do mundo real, através de imagens (BRITO; COELHO, 2002b).

Dentre as inúmeras definições existentes para a fotogrametria, uma pode ser destacada por sua simplicidade e abrangência (BRITO; COELHO, 2002b): "(...) ciência e tecnologia de se reconstruir o espaço tridimensional, ou parte do mesmo (espaço objeto) a partir de imagens bidimensionais, advindas da gravação de padrões de ondas eletromagnéticas (espaço imagem)". De acordo com ASPRS (1966), a fotogrametria é "(...) a arte, ciência e tecnologia de obter informações de confiança sobre objetos e do meio ambiente com o uso de processos de registro, medições e interpretações das imagens fotográficas e padrões de energia eletromagnética registrados". A Figura 20 ilustra o conceito de fotogrametria.

Na parte superior da Figura 20, observa-se um objeto tridimensional (xícara), logo a seguir, um par de imagens bidimensionais do objeto obtidas de pontos de vista diferentes e, por fim, na parte de baixo do modelo a reconstrução do espaço tridimensional. As cruzetas definem os pontos de controle utilizados para a reconstrução. Tais pontos criam uma amarração entre as imagens e o espaço objeto.

Figura 20 - Representação esquemática da fotogrametria



Fonte: (BRITO; COELHO, 2002b).

2.2 Breve Histórico

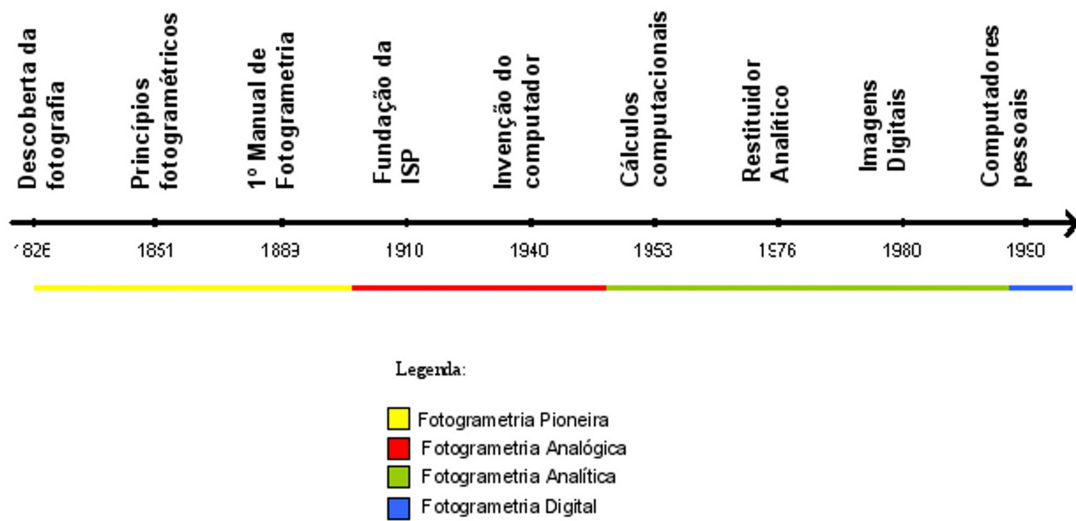
A literatura (BRITO; COELHO, 2002b) costuma dividir a história da fotogrametria em quatro etapas, a saber: fotogrametria pioneira, analógica, analítica e digital. Tal separação é feita em função da tecnologia adotada.

Conforme pode ser observado na Figura 21, o marco inicial da fotogrametria (fotogrametria pioneira) se deu com a descoberta da fotografia, em 1826 (BRITO; COELHO, 2002b). Em pouco tempo, surgiram os primeiros conceitos de fotogrametria, objetivando o aproveitamento da fotografia para a redução de custos em levantamentos topográficos. De acordo com Brito e Coelho (2002b), em 1889 o alemão C. Koppe publicou o primeiro livro teórico sobre fotogrametria.

A primeira grande revolução da fotogrametria (BRITO; COELHO, 2002b) veio com a invenção de um aparelho "estereocomparador" que substituiu os longos cálculos matemáticos por aparelhos óptico-mecânicos. Mas foi em 1911 que o austríaco Theodore Scheimpflug criou o método de retificação de fotografias aéreas. Mais tarde, tais aparelhos foram substituídos por restituidores analógicos ¹⁷ que permitiam visão estereoscópica,

¹⁷ De acordo com Temba (2000), restituição "(...) é o procedimento que pretende obter de fotografias (...) as feições planimétricas e/ou altimétricas(...)"

Figura 21 - Linha Evolucionária da Fotogrametria



Fonte: (BRITO; COELHO, 2002b).

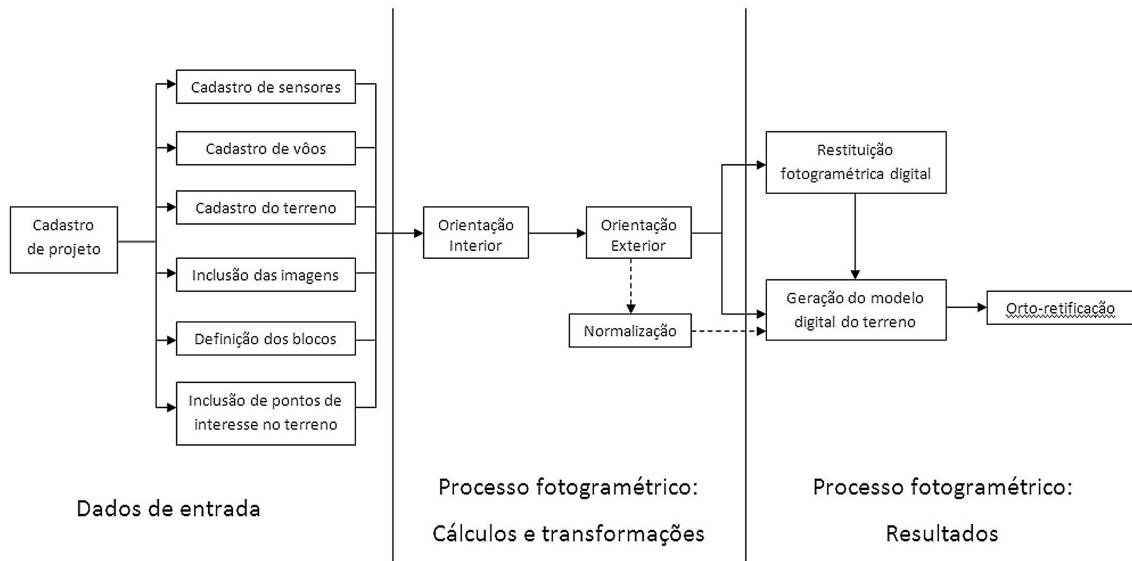
através de pares de imagens (BRITO; COELHO, 2002b). Neste momento, também foi introduzida a técnica de aerotriangulação analógica que permitia o adensamento, em laboratório, dos pontos de campo. Esse momento da fotogrametria foi fundamental para o avanço das técnicas e processos utilizados, nesta fase, foram criadas as câmaras métricas com informações precisas, como as marcas fiduciais (BRITO; COELHO, 2002b).

A partir de 1950, com o uso dos computadores, houve uma nova grande revolução da fotogrametria dando início à fotogrametria analítica. Os processos realizados até então, eram extensamente baseados em cálculos matemáticos complexos e dispendiosos. Com o uso de computadores, novas técnicas foram desenvolvidas como, por exemplo, o tratamento matricial das informações e a solução por mínimos quadrados (BRITO; COELHO, 2002b). Surgem, então, os restituidores analíticos, permitindo a aerotriangulação de blocos de imagens e utilização de câmaras não métricas.

Ao final da década de 1980, com os adventos das imagens digitais e da popularização dos computadores pessoais, surge a fotogrametria digital. Tarefas que eram anteriormente executadas somente em computadores de grande porte puderam ser realizadas em máquinas pessoais. Conseqüentemente, todas as etapas do processo fotogramétrico nos centros de produção catográfica passam a ser realizadas de forma digital.

O processo fotogramétrico digital, tema da seção a seguir, é bastante parecido com o da fotogrametria analítica, diferenciando-se sobretudo na entrada e na saída - imagens de entrada e resultados digitais, como mapas e cartas ortorretificadas (imagens em perspectiva ortogonal), manipulados e produzidos em meio inteiramente digital.

Figura 22 - Etapas do processo fotogramétrico em uma EFD



Fonte: (SILVA, 2008).

2.3 O processo fotogramétrico

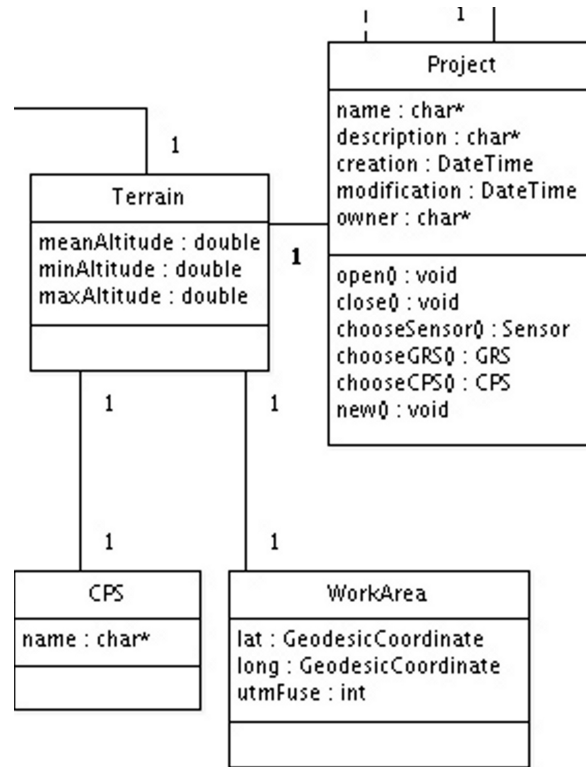
A Figura 22 apresenta as etapas do processo fotogramétrico realizadas numa Estação Fotogramétrica Digital (EFD).

Ao analisar a Figura 22, é possível observar o comportamento cronológico das etapas do processo fotogramétrico. Em primeiro lugar vêm as etapas de entrada de dados - cadastro de projeto, sensores, vôos e dados do terreno, imagens, blocos de imagens e pontos de interesse. Em seguida, são realizadas as etapas de cálculo e transformação sobre as imagens - orientação interior e exterior, normalização. Por fim, são executadas as etapas que produzem os produtos finais da fotogrametria - geração do MNE, restituição espacial 3D e orto-retificação. As seções a seguir apresentam mais detalhadamente as etapas do processo fotogramétrico.

2.3.1 Definição do Projeto

O processo de trabalho numa EFD se inicia com o cadastramento de um projeto fotogramétrico visando à obtenção de dados geoespaciais da área de interesse. Por sua vez, um projeto de mapeamento requer a escolha do sistema geodésico de referência e do sistema de projeção cartográfica. Tais informações serão utilizadas por diversas etapas do processo fotogramétrico. No domínio da EFDE do E-FOTO, estes conceitos são encapsulados pelas classes "Project" (Projeto fotogramétrico), "Terrain" (Terreno) e "Wor-

Figura 23 - Classes da modelagem de Definição do Projeto



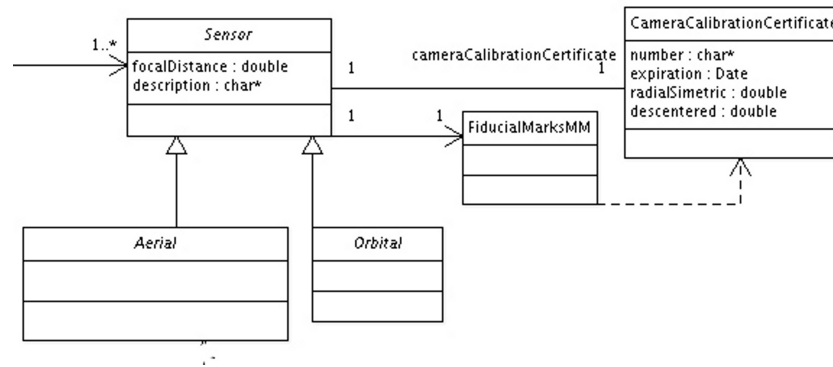
Fonte: O autor.

kArea” (Região de Interesse), ”GRS” (acrônimo do termo técnico em língua inglesa para Sistema Geodésico de Referência) e ”CPS” (acrônimo do termo técnico em língua inglesa para Sistema de Projeção Cartográfica). O fato de dessas classes serem meros repositórios de dados torna o diagrama de classe da definição de projeto, apresentado na Figura 23, de fácil entendimento.

2.3.2 Cadastramento dos Sensores

Projetos fotogramétricos podem valer-se de imagens adquiridas por diferentes tipos de sensores. Segundo Mikhail E.M.; Bethel e McGlone (2001), sensores podem ser classificados de acordo com a plataforma de transporte (aéreos, terrestres ou orbitais), a geometria, o tipo de detector (filme ou CCD), a fonte de energia eletromagnética e as faixas de captura da energia retornada pela cena. O detalhamento dos parâmetros óptico-geométricos dos sensores é fornecido por um documento chamado certificado de

Figura 24 - Classes da modelagem de Sensor



Fonte: O autor.

calibração¹⁸.

A existência de inúmeros tipos de sensores requer uma modelagem abrangente, capaz de representar os detalhes, independentemente do modelo de sensor. Por exemplo, em sensores analógicos métricos há marcas fiduciais que servem de referência para a recomposição da geometria do sensor. A Figura 24 apresenta um diagrama de classes que modela estes conceitos. Podem ser observadas as classes "Sensor", "FiducialMarksMM" (coordenadas no sistema da imagem analógica das marcas fiduciais), "CameraCalibrationCertificate" (Certificado de calibração da câmara). A classe "Sensor" é abstrata, enquanto, as classes abstratas "Aerial" (Aéreo) e "Orbital" são especializações, tipos de sensor em função da plataforma de transporte.

2.3.3 Cadastramento de Vôos Fotogramétricos

Não menos importantes são as informações relativas ao vôo fotogramétrico, o qual, segundo Andrade (2003), é a cobertura fotográfica especialmente projetada para mapear uma região. Informações como limites da região, dados da câmara, velocidade da aeronave, são informações preciosas para o vôo, pois dessas outras informações serão extraídas (aerobase, intervalo entre as exposições, número de faixas, etc.). Uma das informações mais preciosas para o processo fotogramétrico é a altura do vôo que deve ser calculada com base na escala da fotografia. A relação que estabelece a escala média de uma fotografia

¹⁸ O Certificado de Calibração é um documento técnico emitido pelo fornecedor das imagens, que traz informações sobre o sensor utilizado. Informações como distância focal, coeficientes de distorções das lentes e posição em milímetros das marcas fiduciais das imagens, constam desse importante documento.

aérea pode ser dada por Reis T.T; Tommaselli e Ruy (2006):

$$E = \frac{f}{H_v} = \frac{f}{h_v - h_m} \quad (1)$$

onde:

- E : escala da fotografia aérea vertical;
- f : distância focal da câmara (mm);
- H_v : altura do vôo (m);
- h_v : altitude de vôo (m);
- h_m : altitude média do terreno (m).

Por serem importantes para o processo fotogramétrico, as informações do vôo também são armazenadas em classes-repositório.

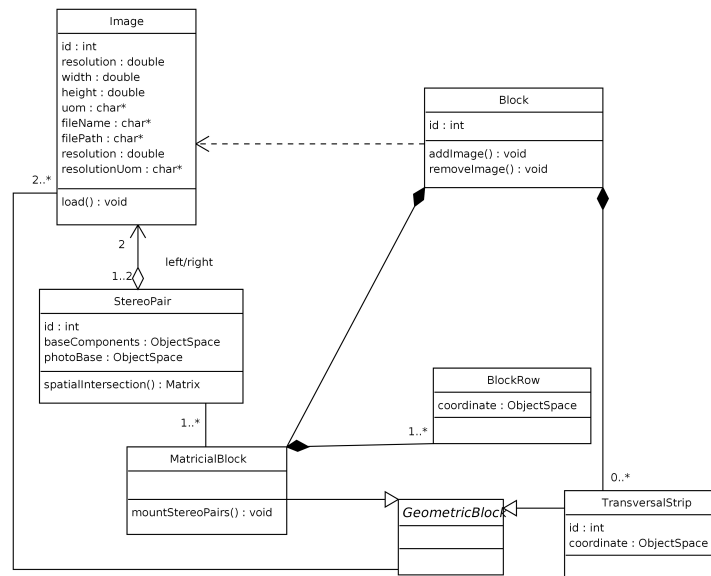
2.3.4 Inclusão de Imagens e Criação do Bloco Fotogramétrico

A EFDE permite somente um bloco por projeto e são necessárias, no mínimo, duas imagens para a formação do bloco. Inicialmente, é requerida a definição da geometria bloco fotogramétrico. Em seguida, as imagens são associadas às respectivas posições do bloco.

A estrutura geométrica do bloco contém as seguintes subpartes: Matricial e Faixa Transversal. Matricial, por sua vez, é formada por um conjunto de faixas horizontais. Além disto, existe a opção de definir faixas transversais, permitindo uma melhor amarração do bloco e, por conseguinte, uma melhor orientação simultânea das imagens. É importante frisar que as imagens que formam o bloco deverão possuir sobreposição, tanto no sentido do vôo (sobreposição longitudinal) quanto no sentido perpendicular à esse (sobreposição lateral ou transversal). Este requisito da sobreposição é fundamental para possibilitar a orientação simultânea das imagens e, através de estereoscopia, a obtenção de medidas precisas no espaço objeto, tarefa realizada em alguns dos processos descritos ao longo deste capítulo.

Observe que, de acordo com a Figura 25, um bloco de imagem na EFDE pode conter objetos das classes *Matricial* (Matricial) e *TransversalStrip* (Faixa Transversal). Blocos matriciais, formados por linhas e colunas de imagens sobrepostas são comumente utilizados em fotogrametria. A classe *TransversalStrip* possibilita uma melhor amarração entre as imagens do bloco e conseqüentemente maior precisão dos cálculos da orientação exterior simultânea.

Figura 25 - Classes da modelagem de Imagem e Bloco de Imagem



Fonte: O autor.

É possível observar, ainda, a criação dos pares de imagens que formarão o efeito de estereoscopia necessário para a reconstrução tri-dimensional do terreno. Também está representado nesse modelo a limitação da quantidade mínima de imagens de um bloco fotogramétrico através do relacionamento entre as classes *GeometricBlock* (Bloco Geométrico) e *Image* (Imagem).

2.3.5 Orientação interior

De acordo com Brito e Coelho (2002b), orientação interior é "(...) a reconstrução do feixe perspectivo, ou seja, o referenciamento da imagem em relação à câmara.". Esta etapa é necessária, pois imagens digitais e câmaras fotogramétricas possuem sistemas de coordenadas distintos. Imagens digitais possuem um sistema de coordenadas discretas, onde cada posição registra um valor de número digital correspondente à intensidade do pixel ¹⁹. As câmaras fotogramétricas, por sua vez, possuem um sistema de coordenadas contínuas, comumente expresso em milímetros. A orientação interior corresponde à

¹⁹ Pixel é uma aglutinação de *Picture element* e corresponde a menor parte da informação de uma imagem (GRAF, 1999).

transformação do sistemas de coordenadas da imagem digital para o sistema da câmara (BRITO; COELHO, 2002b). Tal transformação é expressa por parâmetros de um modelo matemático. Brito e Coelho (2002b) enumeram três modelos matemáticos para esta operação:

1. Transformação Afim Geral - modelo paramétrico com base em seis parâmetros. Trata-se do modelo mais geral e pode ser utilizada em sistemas que apresentem a característica de não-ortogonalidade dos eixos (BRITO; COELHO, 2002b).
2. Transformação Ortogonal ou Afim Ortogonal - é uma variação da transformação afim geral, quando os sistemas são ortogonais, simplificando o modelo matemático empregado (BRITO; COELHO, 2002b).
3. Transformação Isogonal - é um modelo que utiliza apenas quatro parâmetros: uma rotação, um fator de escala e duas translações (D'ALGE, 1987).

A presente discussão, a título de simplificação, se centra na transformação Afim Geral. Leitores interessados nos detalhes dos demais modelos podem se remeter à seguinte referência (BRITO; COELHO, 2002b). O modelo matemático da transformação Afim Geral é expresso pelas equações 2 e 3 a seguir:

$$X = a_0 + a_1 \times \textit{coluna} + a_2 \times \textit{linha} \quad (2)$$

$$Y = b_0 + b_1 \times \textit{coluna} + b_2 \times \textit{linha} \quad (3)$$

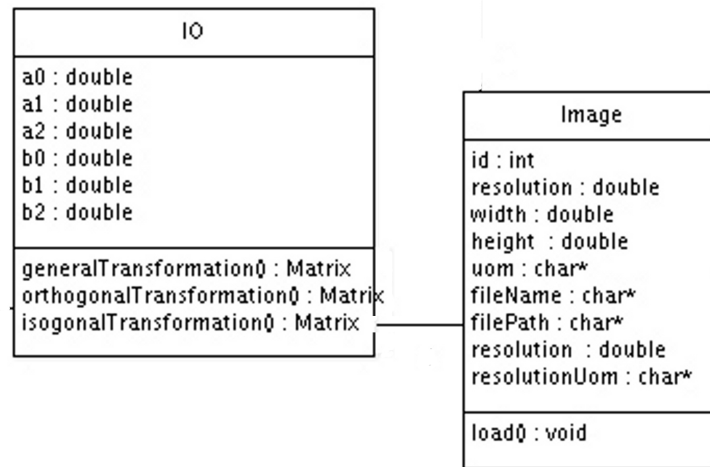
onde:

- X e Y são as coordenadas analógicas no sistema da câmara;
- \textit{coluna} e \textit{linha} são as coordenadas discretas no sistema de imagem digital;
- a_0, a_1, a_2, b_0, b_1 e b_2 são os parâmetros do modelo matemático da Orientação Interior.

A Figura 26 representa um modelo preliminar da estrutura de classes da Orientação Interior, explicitando os parâmetros necessários para a obtenção das coordenadas no espaço imagem. Observe que toda imagem possui sua própria orientação interior.

Se por um lado, para sensores digitais, os parâmetros são previamente definidos, sensores analógicos requerem o cálculo dos parâmetros peculiares a cada imagem. Para que seja possível a determinação dos parâmetros da transformação Afim Geral (BRITO; COELHO, 2002b) são necessárias seis equações linearmente independentes, obtidas a

Figura 26 - Diagrama de Classe preliminar da Orientação Interior



Fonte: O autor.

partir de, pelo menos, três pontos conhecidos tanto no sistema da câmara quanto da imagem digital. Para isso, são utilizadas as marcas fiduciais registradas na imagem, cujas coordenadas no sistema da câmara sejam conhecidas. Essa informação consta no Certificado de Calibração das câmaras fotogramétricas. Como cada imagem tem pelo menos quatro marcas fiduciais e no máximo oito, tem-se no mínimo um par a mais de equações. Deve ser mencionado que o operador, através da interface gráfica, irá fornecer as respectivas coordenadas no sistema de imagem digital. Trata-se, portanto, de um sistema sobredeterminado. Na sua solução, pode ser empregado o método dos mínimos quadrados (GEMAEL, 1994).

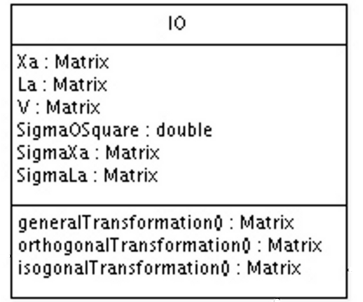
A utilização de uma notação matricial pode melhorar consideravelmente a expressividade do modelo matemático empregado. Em notação matricial o ajustamento por mínimos quadrados do modelo paramétrico da transformação Afim é expresso pela equação 4 a seguir:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 1 & \text{coluna} & \text{linha} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \text{coluna} & \text{linha} \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix} \quad (4)$$

onde:

- Vetor $[X \ Y]$ possui as coordenadas no espaço imagem da câmara;

Figura 27 - Classe da Orientação Interior



Fonte: O autor.

- *coluna* e *linha* são as coordenadas em pixel capturadas pelo operador;
- Vetor $[a_0, a_1, a_2, b_0, b_1$ e $b_2]$ possui os parâmetros calculados pela Orientação Interior.

Em Gemael (1994) pode ser encontrada a formulação para o método paramétrico. A expressão a seguir permite a obtenção dos parâmetros do modelo:

$$X_a = (A^T P A)^{-1} \times (A^T P L_b) \quad (5)$$

onde:

- X_a é o vetor dos parâmetros da Orientação Interior;
- A é a matriz dos valores das coordenadas em pixel das marcas fiduciais;
- L_b é o vetor das coordenadas em milímetros das marcas fiduciais, constantes do certificado de calibração;
- P é matriz-peso. Caso todas as marcas fiduciais tenham desvios-padrão idênticos, P é igual à matriz-identidade.

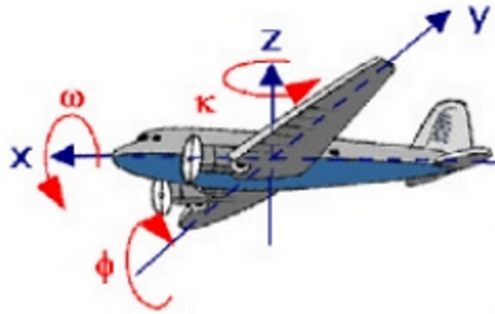
Na modelagem final UML foi feita a opção pela colocação dos atributos da classe de orientação interior no modelo matemático matricial. A Figura 27 apresenta uma versão da modelagem para a classe de orientação interior.

Após o cálculo dos parâmetros, deve-se aferir a qualidade do resultado obtido. Para isso, utiliza-se, entre outras medidas, o vetor de resíduos, que pode ser calculado através da expressão:

$$V = A X_a - L_b \quad (6)$$

onde:

Figura 28 - Esquema da Orientação Exterior



Fonte: (BRITO; COELHO, 2002b)

- V é o vetor de resíduos;
- A é a matriz dos valores das coordenadas em pixel das marcas fiduciais;
- X_a é o vetor dos parâmetros ajustados da Orientação Interior;
- L_b é o vetor das coordenadas em milímetros das marcas fiduciais, constantes do certificado de calibração;

2.3.6 Orientação Exterior

A orientação exterior visa determinar, no sistema de coordenadas do terreno, a posição e os ângulos de atitude da câmara no momento da obtenção da imagem (BRITO; COELHO, 2002b). A Figura 28. ilustra o conceito de ângulos atitude, que são conseqüências dos movimentos da aeronave. Cada ângulo está relacionado a um dos eixos do espaço objeto.

O modelo matemático para o cálculo da orientação exterior é do tipo paramétrico não-linear. As três coordenadas do centro de projeção da câmara, juntamente com o três ângulos de atitude formam um vetor de parâmetros. Com esses seis parâmetros calculados, torna-se possível a determinação das coordenadas de imagem dos pontos do espaço objeto (BRITO; COELHO, 2002b):

- X_0 - posição do centro de perspectiva no eixo X do sistema de coordenadas do espaço objeto;
- Y_0 - posição do centro de perspectiva no eixo Y do sistema de coordenadas do espaço objeto;

- Z_0 - posição do centro de perspectiva no eixo Z do sistema de coordenadas do espaço objeto;
- ângulos de Euler - (ω, ϕ, κ) - rotações sofridas pelo sistema de coordenadas da câmara em relação ao sistema de coordenadas do espaço-objeto.

O modelo matemático para o cálculo desses parâmetros tem sua origem nas chamadas equações de colinearidade (BRITO; COELHO, 2002b):

$$\xi = \xi_0 - c \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (7)$$

$$\eta = \eta_0 - c \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (8)$$

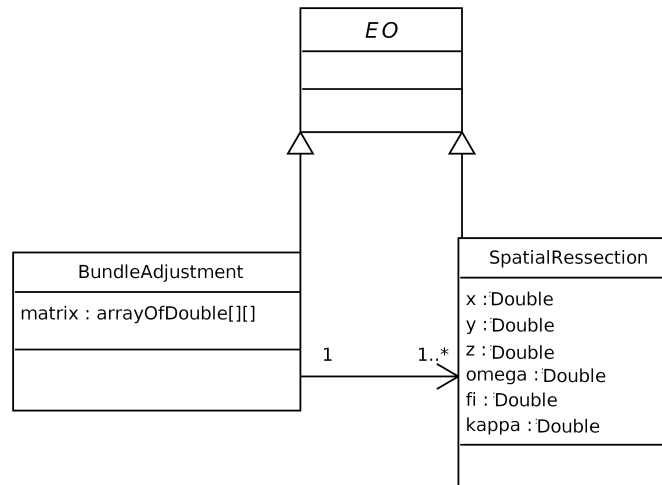
onde:

- ξ e η são as coordenadas no sistema imagem analógico ²⁰;
- ξ_0 e η_0 são as coordenadas do ponto principal (no espaço-imagem analógico) que se encontram no certificado de calibração da câmara;
- X , Y e Z são coordenadas dos pontos de controle no espaço-objeto;
- X_0 , Y_0 e Z_0 são as coordenadas do centro de projeção da câmara no referencial do espaço-objeto;
- c distância focal calibrada da câmara; e
- r_{11} , r_{21} , r_{31} , ..., r_{33} , formam a matriz de rotação R dos ângulos de Euler $(\omega, \phi$ e $\kappa)$.

Para determinar os seis parâmetros, com a solução de ajustamento pelo método dos mínimos quadrados (GEMAEL, 1994), precisa-se de pelo menos quatro pontos não-colineares, cujas coordenadas de terreno sejam conhecidas e que possam ser identificados na imagem (BRITO; COELHO, 2002b). Tais pontos são denominados pontos de controle. Portanto, de posse das coordenadas no espaço-objeto (terreno), das coordenadas no espaço-imagem (fotograma) e da distância focal da câmara (certificado de calibração), podem ser calculados os parâmetros da orientação exterior.

²⁰ ξ e η equivalem a X e Y , respectivamente, nas equações 2.2, 2.3 e 2.4

Figura 29 - Generalização da Orientação Exterior



Fonte: O autor.

Para realizar o cálculo dos parâmetros da orientação exterior, a EFDE pode utilizar dois métodos: resseção espacial e fototriangulação. Desta forma, na modelagem do sistema, a classe Orientação Exterior (OE) foi definida como uma generalização (Figura 29 para as classes de Resseção Espacial (SpatialResection) e Ajustamento por bloco (BundleAdjustment)).

Resseção Espacial

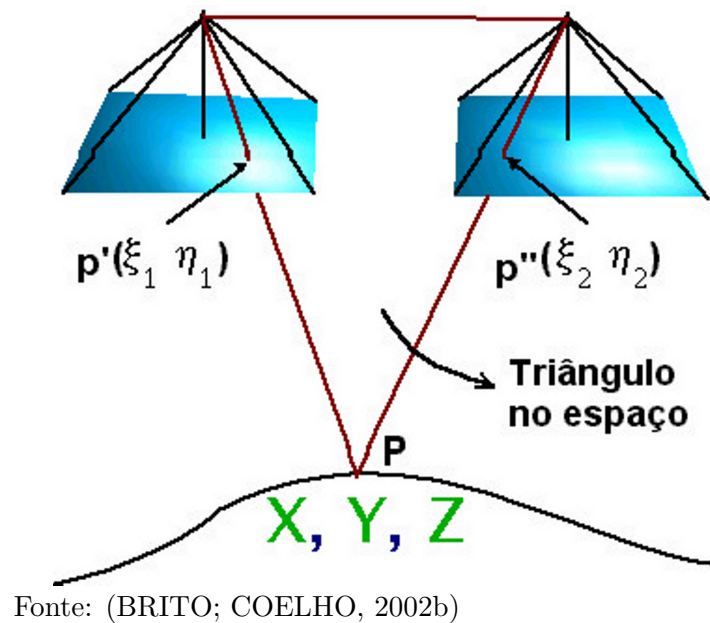
De forma semelhante à orientação interior, o modelo computacional será apresentado na forma matricial:

$$X_a = X_0 - (A^T P A)^{-1} [A^T P (L_0 - L_b)] \quad (9)$$

onde:

- X_a é o vetor a ser calculado (seis parâmetros da orientação exterior);
- X_0 é o vetor com as estimativas iniciais dos seis parâmetros da orientação exterior;
- A é a matriz jacobiana com as equações de colinearidade;
- L_b é o vetor com as coordenadas no espaço-imagem; e
- L_0 é o vetor calculado com bases nas estimativas iniciais das coordenadas no espaço-imagem.

Figura 30 - Interseção espacial



O cálculo da resecção espacial é realizado por um processo iterativo, já que se trata de um modelo não-linear. Tratando-se de um sistema mal condicionado, é preciso realizar uma boa estimativa inicial para os valores de X_0 e L_0 . Assim, a cada iteração, esses valores são recalculados em função da matriz X_a obtidos na iteração anterior. Como critério de parada pode-se ter o valor dos resíduos ou o número de iterações.

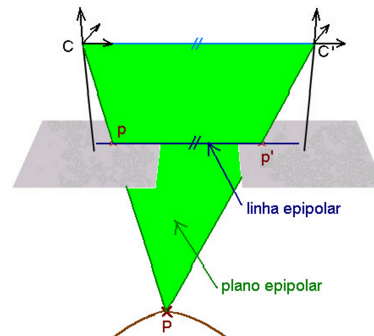
A partir dos parâmetros (X_a), pode-se calcular as coordenadas tridimensionais de qualquer ponto na área de superposição de um par de imagens. Utiliza-se, para isto, o princípio da interseção espacial, ilustrado na Figura 30, que é derivada das equações de colinearidade (7 e 8). Na Figura 30, o ponto P pode ser observado nas imagens da esquerda, $p' = (\xi_1, \eta_1)$, e direita, $p'' = (\xi_2, \eta_2)$. De posse das coordenadas (ξ_1, η_1) e (ξ_2, η_2) , da distância focal e dos parâmetros de orientação (interior e exterior) de ambas imagens, podem ser obtidas as coordenadas de terreno do ponto $P(X_P, Y_P, Z_P)$.

Fototriangulação

De acordo com Brito e Coelho (2002b) a fototriangulação por feixes perspectivos é "(...) a execução de uma resecção espacial simultaneamente à execução da interseção espacial para um conjunto de imagens(...)", ou seja, pode-se, com um único algoritmo matemático, calcular tanto os parâmetros da orientação exterior de todas as imagens de um bloco fotogramétrico, como, também, as coordenadas tridimensionais dos pontos fotogramétricos, no sistema do espaço-objeto.

O cálculo matemático da fototriangulação é iterativo e utiliza-se das mesmas

Figura 31 - Um par estereoscópico normalizado - geometria epipolar



Fonte: (BRITO; COELHO, 2002b)

equações de colinearidade descritas anteriormente. Os parâmetros iniciais de cada passo são os parâmetros calculados no passo anterior, procedimento que é repetido até que "(...) as diferenças a minimizar estejam dentro do limite desejado(...)" (BRITO; COELHO, 2002b).

A fototriangulação requer a escolha de pontos fotogramétricos que, diferentemente dos pontos de controle possuem apenas coordenadas no espaço-imagem. As respectivas coordenadas do espaço-objeto serão calculadas pelo processo de fototriangulação. Esses pontos precisam estar situados na área de sobreposição e precisam ser identificados em pelo menos duas imagens do bloco fotogramétrico. Maiores detalhes sobre fototriangulação podem ser encontrados em Brito e Coelho (2002b) e Silveira (2007).

2.3.7 Normalização

O processo de normalização tem como objetivo "(...) gerar um novo par de imagens digitais que se adapte à chamada geometria epipolar." (BRITO; COELHO, 2002b). Tal processo se faz necessário em virtude dos desvios na rota da aeronave durante a aquisição das imagens. Assim sendo, são obtidos pares não-epipolares (SOKAL et al., 2007).

A normalização equaliza os ângulos de atitude da câmara e elimina os componentes de base ΔY e ΔZ do par de imagens. Em outras palavras, obtém um par com geometria epipolar. A normalização tem como referência o espaço-objeto, ou seja, o terreno e requer o prévio conhecimento dos parâmetros da orientação exterior (SOKAL et al., 2007). A Figura 31 apresenta o conceito de geometria epipolar.

De acordo com Brito e Coelho (2002b), a geometria epipolar se materializa através da existência de um plano epipolar, que por sua vez, é definido pelos dois centros de perspectiva das imagens e mais um ponto no espaço objeto.

É preciso mencionar que, em pares com geometria epipolar, os pontos homólogos ocorrem na mesma linha de ambas imagens do par. Assim sendo, a normalização torna o