ViDA_{ESE}: processo de visualização
exploratória para apoio a estudos experimentais
em verificação, validação e teste de software

Rogério Eduardo Garcia

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: ____ / ____ / 2006

Assinatura: _____

$ViDA_{ESE}$: processo de visualização exploratória para apoio a estudos experimentais em verificação, validação e teste de software

Rogério Eduardo Garcia

Orientadora: *Profa. Dra. Maria Cristina Ferreira de Oliveira*

Tese de Doutorado apresentada ao Instituto de Ciências Matemáticas e de Computação (ICMC-USP), como parte dos requisitos para obtenção do título de Doutor em Ciências de Computação e Matemática Computacional.

ICMC-USP – São Carlos
Agosto/2006

Dedico este trabalho à Caroline,
à Nedyr e ao Sebastião (*in memorian*)

Agradecimentos

À professora Maria Cristina Ferreira de Oliveira e ao professor José Carlos Maldonado, pelo incentivo, apoio, compreensão, direção, parceria e ensinamentos passados ao longo deste trabalho.

Ao Prof. Jeffrey C. Carver (Mississippi State University), pela oportunidade e ensinamentos passados durante o estágio realizado.

Ao Prof. Manoel Mendonça, pela parceria em fases deste trabalho.

À professora Rosane Minghim, à professora Agma Juci Machado Traina e ao professor Leo Pini, membros da Comissão Examinadora do Exame de Qualificação, pela contribuição.

À Caroline, pela compreensão e paciência quanto à minha ausência.

À minha família, em especial aos meus pais – Nedyr e Sebastião (*in memorian*) – pelo apoio incondicional, incentivo e por me ampararem sempre que preciso.

Aos amigos Silvio Tadao Fujisaki (*in memorian*), Milton Hirokazu Shimabukuro, Ronaldo Celso Messias Correia, Antonio Marcos Massao Hachisuca, Reginaldo Ré, Ricardo Menotti e Mário Pazoti, com os quais compartilhei alegrias e superei adversidades, pelo companheirismo.

À Kémelli, com quem tenho aprendido a ter paciência, pelo incentivo, apoio e compreensão.

Ao Thiago – Minhoca – pelo apoio na implementação de parte de uma ferramenta utilizada.

Aos amigos Érika, Bira, Daniela (Unifacs), André, Marcelo, Débora, Otávio, Fabiano, Marco Aurélio, Camila, Haruo e Adenilso, e demais amigos e amigas do Labes, que tornaram muito mais agradáveis os momentos passados dentro e fora do laboratório.

À Profa. Dra. Sandra Fabri, sempre disposta a ajudar, pelas dicas e comentários.

À Érika, pelo trabalho na disponibilização dos dados dos experimentos.

Aos professores do ICMC, em especial à Profa. Rosane Minghim, pela disposição e atenção.

A todos os funcionários do ICMC, em especial à Beth, Laura, Ana Paula e Tatiana, pelo apoio acadêmico.

À FAPESP, pelo apoio financeiro.

Índice

Índice	iv
Lista de Figuras	vii
Lista de Tabelas	x
Resumo	xi
Abstract	xii
1 Introdução	1
1.1 Contextualização	1
1.2 Motivação	3
1.3 Objetivos	4
1.4 Organização	5
2 Experimentação em Engenharia de Software: Trabalhos Relacionados	7
2.1 Estudos Experimentais	9
2.2 O Processo de Experimentação	12
2.3 Papéis da Replicação	17
2.3.1 Arcabouço para Melhoria da Experimentação	21
2.3.2 Meta-Análise e a Análise de Dados Combinados	23
2.3.3 Geração de Hipóteses e Lista de Variáveis	24
2.4 Experimentos Realizados em VV&T	26
2.4.1 Histórico dos Experimentos PBR	28
2.4.2 Resultados Obtidos por Análise Estatística	31
2.5 O Projeto <i>Readers</i>	36
2.6 Considerações Finais	40
3 Visualização de Informação	42
3.1 Conceitos Básicos e o Modelo de Visualização	43
3.2 Visualização em Mineração de Dados	46
3.2.1 Mineração de Dados	46
3.2.2 Mineração Visual de Dados	48
3.3 Técnicas para Visualização Exploratória	50

3.3.1	Técnicas Orientadas a Pixel	51
3.3.2	Técnicas de Projeção Geométrica	53
3.3.3	Técnicas Iconográficas	56
3.3.4	Técnicas Hierárquicas	57
3.4	Técnicas de Interação e Distorção	59
3.4.1	Filtragem Interativa	60
3.4.2	Zoom Interativo	60
3.4.3	Distorção Interativa	61
3.4.4	Seleção e Ligação Interativas	62
3.5	Múltiplas Visualizações Coordenadas	62
3.6	Ferramentas	67
3.6.1	Spotfire Decision Site	67
3.6.2	<i>XmdvTool</i>	68
3.7	Considerações Finais	72
4	Processo de Visualização Estendido para Engenharia de Software Experimental	74
4.1	Visualização em Experimentação: Adequações e Desdobramentos	75
4.2	Abordagem para Meta-Análise Exploratória	80
4.3	Análise Visual de Dados para Engenharia de Software Experimental	82
4.4	Atividades no Espaço dos Dados	87
4.5	Atividades no Espaço Visual	88
4.5.1	Mapeamento Visual	88
4.5.2	Transformações de Visões	100
4.6	Arcabouço para Visualização	101
4.7	Considerações Finais	107
5	Aplicação do Processo $ViDA_{ESE}$: Resultados	109
5.1	Espaço de Dados: Etapas Gerais	110
5.2	Espaço Visual	114
5.2.1	Análise da Experiência dos Participantes	114
5.2.2	Análise de Defeitos por Técnica/Documento	118
5.2.3	Análise de Defeitos Reportados por Documento, Tipo e Página	124
5.3	Geração de Dados: uma abordagem baseada em Algoritmos Genéticos	127
5.3.1	Algoritmos Genéticos	128
5.3.2	Mapeamento do Problema	130
5.3.3	Análise dos Dados Gerados	132
5.4	Considerações Finais	135
6	Conclusões	136
6.1	Contribuições	137
6.2	Limitações	139

6.3 Desenvolvidimentos Futuros	139
Referências Bibliográficas	141

Lista de Figuras

2.1	Processo de Experimentação (Wohlin et al., 1999)	14
2.2	Fase de Definição do Experimento	14
2.3	Fase de Planejamento do Experimento	15
2.4	Fase de Execução do Experimento	16
2.5	Fase de Análise e Interpretação dos Dados Coletados	17
2.6	Replicações de um experimento e o Pacote de Laboratório	20
2.7	Ciclos do FIRE (Mendonça et al., 2006)	21
2.8	Metodologia para o desenvolvimento de lista de variáveis (Carver, 2003)	25
2.9	Família de Técnicas de Leitura (Basili et al., 1996a)	27
2.10	Projeto do Experimento (Basili et al., 1996c)	29
2.11	Formulário de Registro de Defeitos Encontrados (Basili et al., 1998)	32
2.12	Resultados da análise dos times simulados (Basili et al., 1996c)	34
2.13	Resultados condensados de Basili et al. (1996c)	35
2.14	Cobertura de defeitos por perspectiva para o documento ATM (Basili et al., 1996c)	36
2.15	Projeto de replicações da Linha de Pesquisa II do Projeto <i>Readers</i>	37
3.1	Modelo para Visualização de Dados (Card et al., 1999).	44
3.2	Processo de Mineração de Dados (Rezende et al., 2003)	47
3.3	Categorias de VDM segundo Ankerst (Ankerst, 2000)	50
3.4	Distribuição dos atributos visualizados (Keim, 1997).	52
3.5	Arranjo em forma de espiral (Keim, 1997).	52
3.6	Organização em espiral por quadrante (Keim, 1997).	52
3.7	Visualização por <i>pixels</i> de um conjunto de dados aleatório em que cada item tem 9 atributos, usando dois arranjos distintos para a disposição dos itens (Keim, 1997).	53
3.8	Matriz de Gráficos de Dispersão gerada com o conjunto de dados coletados em replicações de experimentos realizadas no âmbito do Projeto <i>Readers</i>	54
3.9	Visualização por Coordenadas Paralelas de sete atributos do Conjunto de Dados coletados em replicações de experimentos realizadas no âmbito do Projeto <i>Readers</i>	55

3.10	Exemplo de Visualização com <i>Stick Figure</i>	57
3.11	Subdivisão do espaço de coordenadas(Wong e Bergeron, 1997)	58
3.12	Exemplo de <i>Treemap</i> apresentando o conjunto de defeitos reportados em repli- cações do Experimento II do Projeto Readers	59
3.13	Controles deslizantes com um (a) e dois cursores (b), e botões de rádio (c) e caixas de verificação (d)	60
3.14	Instantâneo da ferramenta <i>TableLens</i> (Rao e Card, 1994).	61
3.15	Exemplo de “máscara” de distorção (Carpendale et al., 1997)	62
3.16	Seleção de marcadores em: um gráfico Gráfico de Dispersão – (a) e (b) – e em uma representação por Coordenadas Paralelas – (c) e (d).	63
3.17	Esquema de visualização em uso na ferramenta <i>Snap Together Visualization</i>	65
3.18	Visão geral da arquitetura da plataforma InfoVis (Shimabukuro, 2004)	67
3.19	Instantâneo da ferramenta <i>SpotFire</i>	69
3.20	Trecho de um arquivo padrão OKC utilizado pelo sistema <i>XmdvTool</i>	70
3.21	Instantâneo da ferramenta <i>XmdvTool</i> com uma representação em Coordenadas Paralelas	71
3.22	Interface para controle de exibição do agrupamento hierárquico na ferramenta <i>XmdvTool</i> (Fua et al., 1999a).	72
3.23	Duas visualizações exibindo um conjunto de dados em diferentes níveis de detalhe	72
4.1	Perspectivas de uso de Técnicas de Visualização em um processo de experimen- tação	76
4.2	Processo de Experimentação Modificado a partir de Wohlin et al. (1999)	78
4.3	A atividade de Análise considerando meios alternativos (Visualização e Mine- ração) além da Estatística	79
4.4	Abordagem proposta para a meta-análise exploratória	81
4.5	$ViDA_{ESE}$: <i>Visual Data Analysis to Experimental Software Engineering</i>	84
4.6	Dados brutos coletados em um experimento	88
4.7	Distribuição de defeitos reportados na replicação R1, por tipo e documento (cor ênfatisa o tipo de defeito)	89
4.8	Sumarização de defeitos reportados na replicação R1, por tipo e documento (cor identifica o tipo de defeito)	90
4.9	Representação por Coordenadas Paralelas das diferentes Experiências dos par- ticipantes da replicação R1	91
4.10	Hipótese 1 para Análise Confirmatória	92
4.11	Representação por Coordenadas Paralelas obtida pela seqüência intuitiva (R2)	93
4.12	Hipótese 2 para Análise Confirmatória	93
4.13	Representação por Coordenadas Paralelas das hipóteses 1 e 2 na mesma projeção	94
4.14	Representação equivalente à Figura 4.13, com mais um eixo: <i>Domínio da Língua</i>	95
4.15	Representação por Coordenadas Paralelas com seqüência aleatória de atributos	96

4.16	Representação por Coordenadas Paralelas depois de estabelecida uma sequência pela análise visual das correlações e com normalização global	96
4.17	Representação por Coordenadas Paralelas obtida com a ferramenta de Artero e Oliveira (2004)	97
4.18	Arcabouço de Visualização	101
4.19	Instantâneo da ferramenta de suporte à seleção e à estruturação de dados	102
4.20	Estrutura do <i>template</i> proposto em XML	104
4.21	Parte de um <i>template</i> gerado pela ferramenta implementada	105
4.22	Visualização obtida com a aplicação de um <i>template</i> gerado previamente	106
5.1	Modelo da base de dados para análise que mantém dados de todas as replicações	111
5.2	Visualizações por Coordenadas Paralelas de atributos relativos à experiência do conjunto de dados de <i>Fall98</i>	116
5.3	Visualizações por Coordenadas Paralelas de atributos relativos à experiência do conjunto de dados de <i>Fall98</i> e <i>Fall01</i>	117
5.4	Representação hierárquica dos defeitos de reportados em <i>R1</i> e <i>R2</i> obtida com Treemap	119
5.5	Organização hierárquica dos defeitos por documento, técnica e experiência . .	120
5.6	Representação obtida a partir da organização hierárquica exposta na 5.5 com dados de <i>R1</i>	121
5.7	Representação obtida a partir da organização hierárquica exposta na 5.5 com dados de <i>R2</i>	122
5.8	Representação obtida a partir da organização hierárquica exposta na 5.5 com dados de <i>R1</i> e <i>R2</i>	122
5.9	Representação obtida com os dados relativos às replicações <i>R1</i> e <i>R2</i> usando Treemap	123
5.10	Representação hierárquica dos defeitos detectados em <i>R1</i> e <i>R2</i> obtida com técnica Treemap	125
5.11	Gráfico de Dispersão com a distribuição de defeitos por tipo e documento (a cor enfatiza o tipo de defeito)	126
5.12	Esquema de simulação de dados com base em Algoritmos Genéticos	128
5.13	Operação de <i>crossover</i>	129
5.14	Exemplo de operação de mutação	129
5.15	Instantâneo da ferramenta que implementa a abordagem de mutação proposta exibindo defeitos gerados	131
5.16	Distribuição de defeitos por Tipo	133
5.17	Agrupamentos hierárquicos dos dados simulados gerados com a ferramenta XmdvTool – representados em Coordenadas Paralelas Hierárquicas	134

Lista de Tabelas

2.1	Distribuição de Defeitos por Documento (Basili et al., 1996c)	29
2.2	Sumário do Questionário de Entrevista da Replicação 1 (R1) do Experimento II	38
2.3	Sumário de Defeitos reportados na Replicação 1 (R1) do Experimento II para o documento PG	40
4.1	Diretrizes para a análise visual de dados no contexto de experimentação (ESE) .	86
5.1	Lista parcial de atributos relativos à experiência de participantes	112

Resumo

A Engenharia de Software evolui gradativamente para uma disciplina científica baseada em observação, formulação teórica e experimentação. Nesse contexto, estudos experimentais têm sido conduzidos para proporcionar evidências sobre a qualidade e a produtividade de técnicas, ferramentas e métodos de desenvolvimento de software. Porém, resultados obtidos a partir de pesquisas experimentais são úteis somente se os pesquisadores puderem consolidá-las em um contexto significativo de conhecimento. O projeto *Readers: A Collaborative Research to Develop, Validate and Package Reading Techniques for Software Defect Detection* aborda diversos aspectos envolvidos na construção de um corpo significativo de resultados a partir de experimentos controlados. Nesse contexto, sustenta-se que a aplicação de meios alternativos de análise e exploração de dados dos experimentos conduzidos pode colaborar para a evolução do Pacote de Laboratório, e como efeito colateral, contribuir com a formação do corpo de conhecimento almejado. Este projeto propõe mecanismos para integrar técnicas de visualização exploratória à análise de dados de experimentos controlados. Para isso, foram propostas adequações ao processo de experimentação, bem como um processo de visualização para Engenharia de Software Experimental – $ViDA_{ESE}$. Para ilustrar o $ViDA_{ESE}$, ciclos de análises são apresentados, juntamente com as conclusões obtidas em cada um. As análises visuais apóiam o processo de consolidação de conhecimento a partir dos dados e a melhoria do Pacote de Laboratório. Adicionalmente, uma abordagem para simulação de dados experimentais foi proposta, permitindo explorar cenários que apóiem o planejamento de experimento.

Abstract

Software engineering must evolve to a scientific discipline based upon observation, theory formulation and experimentation. In this context, experimental studies have been conducted to provide evidence about the quality and the productivity of software development methods, techniques and tools. However, results obtained from experimental research are useful only if researchers can consolidate them into a significant body of accessible knowledge. The project “Readers: A Collaborative Research to Develop, Validate and Package Reading Techniques for Software Defect Detection” addresses issues involved in the running of effective replications to build a meaningful body of results from controlled experiments. In this context, alternative techniques to analyze and explore data from experimental studies can support evolution of the Lab Package, thus contributing with the construction of a body of knowledge. This work proposes mechanisms to apply exploratory visualization techniques to assist the analysis of data collected in experimental studies. To achieve this goal both the experimental and the visualization process had to be modified, in particular a visual data analysis process for experimental software engineering – $ViDA_{ESE}$ – was proposed. $ViDA_{ESE}$ is illustrated with its application to execute multiple cycles of analyses on data from PRB experiments, conducted under scope of the Readers Project. These analyses are described and the conclusions drawn from them are presented, as well as their contribution to consolidate knowledge from data. Furthermore, an approach for simulating experimental data is proposed, which exploring additional scenarios to support planning new experiments.

Introdução

1.1 Contextualização

A evolução da Engenharia de Software nas últimas décadas, procurando estabelecer técnicas, critérios, métodos e ferramentas para a produção de software, visa a atender uma crescente demanda por qualidade e produtividade, tanto do ponto de vista do processo de produção como do ponto de vista dos produtos gerados. O desenvolvimento de software é um processo calcado em pessoas e conhecimento, com muitas atividades ad-hoc e dependentes da experiência pessoal. Para atingir seus objetivos, equipes de desenvolvimento precisam entender e escolher modelos e técnicas apropriados para apoiar seus projetos.

Nessa perspectiva, estudos experimentais vêm sendo realizados, contribuindo para a formação de uma base de conhecimento sobre Engenharia de Software que pode ajudar nas tomadas de decisão ao longo do desenvolvimento. Várias são as iniciativas em estudos experimentais. O *Center for Empirically-Based Software Engineering*¹ (CeBASE) reúne modelos experimentais para fornecer guias validados de seleção de técnicas e modelos, para recomendar áreas de pesquisa, e para dar apoio ao ensino de Engenharia de Software. O objetivo do CeBASE é conduzir a Engenharia de Software para uma disciplina baseada na Engenharia, na qual os processos de desenvolvimento são selecionados baseados no que se sabe sobre seus efeitos nos

¹<http://www.cebase.org>

produtos, juntamente com a síntese, a derivação, a organização e a disseminação do conhecimento experimental em desenvolvimento de software. Como parte do CeBASE, a *International Software Engineering Research Network*² (ISERN) agrega pesquisadores que acreditam que a pesquisa em Engenharia de Software precisa ser executada em um contexto experimental que permita observar e experimentar as tecnologias em uso, e compreender seus pontos fortes e fracos. As iniciativas motivaram a criação do *Journal of Empirical Software Engineering* (JESE³), um fórum de publicações sobre pesquisas em Experimentação.

Estudos experimentais vêm sendo conduzidos sobre os diversos métodos e técnicas aplicáveis às diversas etapas do processo de desenvolvimento de software (Prechelt e Unger, 1998, Lanubile e Visaggio, 1995, Perry et al., 1998). Um conjunto de atividades tem sido introduzido com o objetivo de melhorar a qualidade do processo e dos produtos de software, dentre as quais, atividades de Verificação, Validação e Teste (VV&T), com diversos estudos realizados (Basili et al., 1986, Basili e Selby, 1987, Basili et al., 1996c,b,a, 1998, Fusaro et al., 1997, Kamsties e Lott, 1995a, Porter et al., 1995, Sova e Smidts, 1996, Wood et al., 1997, Zelkowitz e Wallace, 1997, Hohn, 2003, Maldonado et al., 2006). Resultados de estudos experimentais só são úteis se consolidados em um corpo de conhecimento significativo para a comunidade que atua na área, o que leva a um aspecto relevante em experimentação: a necessidade de replicar estudos experimentais de modo a gerar dados e compor o corpo de conhecimento.

Em uma iniciativa nesse contexto, o Grupo de Engenharia de Software do Instituto de Ciências Matemáticas e de Computação (ICMC-USP), em colaboração com a University of Maryland, Universidade Federal de São Carlos, Universidade Federal do Rio de Janeiro-COPPE e Universidade de Salvador (Unifacs), desenvolveu o projeto *Collaborative Research to Develop, Validate and Package Reading Techniques for Software Defect Detection (NSF-CNPq Readers)*. O Projeto *Readers* tem como principais metas: criar tecnologias para apoiar a detecção de defeitos e desenvolver métodos experimentais que possam contribuir para melhorar essas tecnologias para uso em diferentes ambientes e contextos culturais; desenvolver métodos de avaliação que permitam comparar diversas tecnologias de detecção de defeitos em contextos industriais e acadêmicos; e expandir a base tecnológica de análise de software criando novos pacotes de laboratório, que registram as atividades realizadas na execução de experimentos, para serem

²<http://www.iese.fhg.de/ISERN/>

³<http://www.kluweronline.com/issn/1382-3256>

reutilizados e adaptados por outros pesquisadores, baseando-se em seus próprios ambientes e necessidades culturais (Shull et al., 2002).

1.2 Motivação

As diversas execuções de experimentos – e replicações – geram dados que têm sido analisados estatisticamente. Em especial, as replicações ocorridas no âmbito do projeto *Readers* oferecem condições ideais para investigar a aplicação de estratégias de análise de dados exploratória, alternativas às abordagens estatísticas tradicionais no contexto de Engenharia de Software Experimental. Análise visual foi considerada como uma alternativa viável e promissora, pois os dados acumulados nas múltiplas replicações podem embutir conhecimento “novo” (não antecipado), e técnicas de visualização exploratória permitem aproveitar o conhecimento tácito de experimentadores no processo de busca para identificar tal conhecimento. O próprio arcabouço para experimentação (FIRE – *Framework for Improving the Replication of Experiments*) proposto por Mendonça et al. (2006), que visa a gerenciar o conhecimento e melhorar replicações de experimentos controlados em engenharia de software, provê um cenário dos pontos em que a visualização exploratória pode colaborar com a Engenharia de Software Experimental.

Durante as análises iniciais do corpo de dados coletados nos experimentos foram observadas dificuldades e limitações na instanciação do modelo de processo de visualização proposto por Card et al. (1999), independentemente da adequação das técnicas de visualização empregadas (o que nem é sempre facilmente determinado). A replicação de experimentos é regida pelo Pacote de Laboratório – composto pelos artefatos utilizados e pela descrição dos procedimentos, o pacote de laboratório compõe uma infra-estrutura experimental para dar suporte a futuras replicações descrevendo um experimento em termos específicos, fornecendo material para replicação e estabelecendo um contexto para combinação de resultados de diferentes tipos de tratamentos experimentais –, que deve explicitar as atividades, inclusive de análise de dados, para garantir fidelidade e consistência entre replicações. Tudo isso introduz requisitos ao processo de análise.

O papel de analista pode ser exercido por diferentes pessoas (com diferentes níveis de experiências) ao longo do processo de experimentação. Adicionalmente, o usuário de técnicas de visualização pode assumir diversos papéis no processo de experimentação, o que também introduz necessidades específicas ao processo de exploração visual. Há também atividades en-

envolvidas em processos de análise de dados oriundos de múltiplos experimentos que devem ser consideradas para o emprego efetivo de recursos de visualização exploratória.

Para aplicar recursos de visualização exploratória nesse contexto fez-se necessário propor adequações tanto no processo de experimentação, quanto no processo de visualização proposto por [Card et al. \(1999\)](#), buscando explicitar as atividades envolvidas para usar as técnicas de visualização de modo consistente – considerando os diversos papéis, que podem ser exercidos por diferentes pessoas – e aproveitando o conhecimento dos experimentadores sobre o processo de experimentação.

1.3 Objetivos

Este trabalho de doutorado tem por objetivo propor mecanismos para ajudar pesquisadores em Engenharia de Software Experimental a extrair mais conhecimento da massa de dados coletados e validar conclusões *inter*-experimentos. A proposta é que, por meio de técnicas exploratórias com alto grau de interação do analista (possibilitando usar de seu conhecimento sobre experimentação), seja possível prover diretrizes para a evolução do Pacote de Laboratório, de modo a atingir este objetivo. A evolução do Pacote de Laboratório se dá, por exemplo, pelo aprimoramento da lista de variáveis e das hipóteses avaliadas, como consequência de conhecimento extraído. Adicionalmente, ou até mesmo como consequência da melhora do Pacote de Laboratório, busca-se facilitar a transferência de conhecimento entre os participantes do processo experimental, provendo mecanismos de permitem explicitar conhecimento tácito. Assim, técnicas exploratórias de Visualização de Informação oferecem recursos alternativos à análise de dados estatística, e com a possibilidade de intervenção do analista. Para isso, é preciso adequar os procedimentos atualmente empregados em ambas as áreas: tanto para acomodar a inserção da exploração visual como parte do processo de experimentação, quanto para especificar um processo de visualização que atenda aos requisitos da experimentação. Esses são os principais pontos abordados neste trabalho.

Para atingir o objetivo, foram propostas mudanças no processo de experimentação para inserir abordagens alternativas de análise. Também foi proposto um processo de visualização estendido para ser aplicado à análise de dados de estudos experimentais em Engenharia de Software (*ViDA_{ESE}* – *Visual Data Analysis for Empirical Software Engineering*). As conclusões obtidas com as análises visuais a partir da instanciação do processo *ViDA_{ESE}* são apresentadas,

sugerindo evoluções no Pacote de Laboratório.

Adicionalmente, o conhecimento adquirido a partir das análises visuais sustentou a proposta de uma abordagem para a simulação de dados de estudos experimentais. A simulação de estudos experimentais é prevista na taxonomia estendida por [Travassos e Barros \(2003\)](#), que considera duas possibilidades de simulação: estudos *in virtuo* e *in silico*, ambas baseadas em modelos computacionais. [Travassos e Barros \(2003\)](#) apontam como vantagens dos experimentos simulados (em especial dos experimentos *in virtuo*) o baixo custo – comparado com *in vitro* e *in vivo*, que consistem em realização de estudos em ambientes laboratoriais e reais, respectivamente – e a facilidade de aplicação em situações reais (não laboratoriais). Sua aplicação é indicada em casos em que o objeto de estudo envolve atividades que perduram durante longos períodos, como meses ou anos, durante o projeto de desenvolvimento de software, e conseqüentemente a replicação de experimentos é proibitiva. Nesse contexto foi proposta uma abordagem baseada em Algoritmo Genético que apóia o planejamento de novos estudos, com base na prospecção de cenários a partir de dados reais.

1.4 Organização

Diferentes áreas da Computação estão envolvidas no escopo deste projeto: Visualização de Informação e Engenharia de Software, mais precisamente as atividades de VV&T. A Experimentação é o elemento de ligação entre elas. Assim, para contextualizar a proposta e apresentar os resultados, o presente documento está organizado como segue:

- No Capítulo [2](#) discute-se a Experimentação em Engenharia de Software, direcionada a experimentos realizados no âmbito do projeto *Readers*. O processo de experimentação é apresentado, juntamente com a descrição dos experimentos realizados no projeto *Readers*;
- No Capítulo [3](#) é apresentada uma visão geral da área de Visualização de Informação, com enfoque em Visualização Exploratória. Algumas Técnicas de Interação e Distorção também são apresentadas, devido à importância para a análise visual. Uma breve descrição de Mineração de Dados é apresentada com o objetivo de contextualizar a integração de Visualização a processos de descoberta de conhecimento. Também são apresentadas as ferramentas disponíveis;

- No Capítulo 4 são apresentadas as contribuições deste trabalho, que consistem em adequações no processo de experimentação para acomodar a integração com a análise visual de dados, e o processo de visualização instanciado para Engenharia de Software Experimental – $ViDA_{ESE}$. Foram definidas diretrizes para a instanciação do processo proposto, também apresentadas nesse capítulo;
- No Capítulo 5 é apresentada a instanciação do processo $ViDA_{ESE}$ em ciclos de análise de dados coletados no âmbito de experimentos de VV&T, juntamente com as observações obtidas em cada ciclo, destacando sugestões para a evolução do Pacote de Laboratório. Uma abordagem baseada em Algoritmo Genético para simulação de dados de experimentos também foi proposta e é apresentada. A simulação permite explorar cenários, de modo a apoiar o planejamento de experimentos;
- Por fim, no Capítulo 6 apresentam-se as conclusões desta tese, com um resumo do trabalho efetuado, as contribuições ao estado da arte, as limitações e sugestões para a continuidade desta abordagem em trabalhos futuros.

Experimentação em Engenharia de Software: Trabalhos Relacionados

*E*ngenharia de software é uma área que aborda elementos multidisciplinares que incluem desde aspectos técnicos, como banco de dados e sistemas operacionais, a aspectos sociais (Wohlin et al., 1999). É inegável que seja altamente dependente de habilidades humanas. É importante buscar a evolução da Engenharia de Software com o intuito de estabelecer uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de software, minimizando a dependência de fatores humanos. Nesse contexto, pesquisadores têm dispensado especial atenção à avaliação experimental de técnicas e métodos para Engenharia de Software. O *Center for Empirically-Based Software Engineering*¹ (CeBASE), é uma cooperação entre universidades norte-americanas (University of Maryland/Fraunhofer Center, University of Southern California, University of Nebraska e Mississippi State University), com a participação do ICMC-USP como colaborador. O centro reúne modelos experimentais para fornecer guias avaliados de seleção de técnicas e modelos, para recomendar áreas de pesquisa, e para apoiar o ensino de Engenharia de Software. O CeBASE foi organizado para dar apoio a organizações de software nas respostas a questões chave como: definir qual o modelo de processo de ciclo de vida mais adequado para um determinado projeto

¹<http://www.cebase.org>

particular; estabelecer a proporção apropriada de esforço entre inspeção e teste em um contexto específico; identificar os benefícios, se houver, em comprar componentes de software disponíveis ao invés de desenvolvê-los. O objetivo do CeBASE é conduzir a Engenharia de Software para uma disciplina baseada na Engenharia, na qual os processos de desenvolvimento são selecionados com base no que se sabe sobre seus efeitos nos produtos. Busca-se contribuir com a síntese, a derivação, a organização e a disseminação do conhecimento em desenvolvimento de software. A *International Software Engineering Research Network*² (ISERN) agrega pesquisadores que acreditam que a pesquisa em Engenharia de Software precisa ser executada em um contexto experimental que permita observar e avaliar as tecnologias em uso, e compreender seus pontos fortes e fracos. Pode-se, assim, adaptar as tecnologias às características particulares dos projetos e empacotá-las juntamente com os experimentos para aumentar o reuso em projetos futuros.

A validação experimental pode ser aplicada a diversas atividades do processo de desenvolvimento de software. Estudos demonstram que, para a avaliação, não basta considerar somente a técnica em questão, é preciso considerar também o contexto de sua utilização, por exemplo, o tipo de software e o ambiente de aplicação (Carver, 2003). Estudos experimentais oferecem uma alternativa para avaliar a aplicação de técnicas em diferentes contextos, e a replicação de experimentos, embora não seja trivial, gera um corpo de conhecimento capaz de apoiar análises consistentes das técnicas (Shull et al., 2002).

Neste capítulo é apresentado o contexto do trabalho desenvolvido, focado em experimentos aplicados na avaliação das técnicas e métodos de Verificação, Validação e Teste de software (VV&T) (Basili et al., 1986, Basili e Selby, 1987, Basili et al., 1996c,b,a, 1998, Fusaro et al., 1997, Kamsties e Lott, 1995a, Porter et al., 1995, Sova e Smidts, 1996, Wood et al., 1997, Zelkowitz e Wallace, 1997, Maldonado et al., 2006). Este capítulo encontra-se organizado como segue:

- Na Seção 2.1 é apresentada uma visão geral dos estudos experimentais em Engenharia de Software;
- Na Seção 2.2 é apresentado o processo de experimentação em Engenharia de Software. Suas etapas e atividades envolvidas são detalhadas;

²<http://www.iese.fhg.de/ISERN/>

-
- Na Seção 2.3 é apresentada uma discussão sobre a necessidade de replicar estudos experimentais e os problemas observados na replicação e introduzidos por ela;
 - Na Seção 2.4 é apresentado o experimento base das replicações que originaram os dados utilizados neste trabalho, mostrando sua organização e as conclusões derivadas de análises estatísticas;
 - Na Seção 2.5 são apresentados o projeto *Readers*, os experimentos realizados (que foram utilizados como base para os estudos realizados neste trabalho), suas replicações e a organização dos dados coletados.

2.1 Estudos Experimentais

Segundo Wohlin et al. (1999) os seguintes requisitos são necessários para obter êxito no desenvolvimento de software:

1. Compreensão do processo e produto de software;
2. Definição de processo de desenvolvimento e qualidade do produto;
3. Avaliação de sucessos e falhas;
4. Realimentação de informação para controle do projeto;
5. Aprendizagem com a experiência;
6. Empacotamento e reutilização da experiência relevante.

Estudos experimentais são ferramentas para obter os dados necessários para apoiar as atividades acima. Eles apóiam a busca de resultados objetivos e significativos em relação à compreensão, controle, prognóstico e melhoria nos processos de desenvolvimento de software. Ainda segundo Wohlin et al. (1999), os tipos de estudos experimentais em engenharia de software incluem: **pesquisa de opinião** (*survey*), **estudos de caso**, e **experimentos controlados**.

O primeiro deles – Pesquisa de Opinião – é geralmente uma investigação executada em retrospecto, quando, por exemplo, uma ferramenta ou técnica já está em uso e deseja-se obter dados a respeito (*feedback*). Os meios básicos para obtenção de dados, sejam qualitativos ou

quantitativos, são entrevistas ou questionários aplicados a uma amostra representativa da população a ser estudada. Os resultados são analisados e generalizados para a população da qual se extraiu a amostra. Em uma pesquisa de opinião não há controle da execução ou das medidas. Elas são efetuadas em três situações distintas: quando o objetivo é alcançar a capacidade necessária para afirmar algo sobre uma população – pesquisa descritiva; quando se pretende sustentar explicações sobre uma população – pesquisa explicativa; ou na condição de pré-estudo para uma investigação mais completa – pesquisa de opinião exploratória.

O segundo tipo de estudos experimentais – Estudos de Caso – é usado para monitorar projetos, atividades ou tarefas visando a observar um aspecto ou atributo específico, ou estabelecer relações entre atributos diferentes. O investigador coleciona informações detalhadas sobre uma tarefa durante um período contínuo de tempo, podendo utilizar uma variedade de procedimentos de coleta. Por ser um estudo observacional, o controle nesse tipo de experimento é menor do que em um estudo controlado.

Experimentos controlados são executados manipulando a execução do estudo direta e sistematicamente. Em um experimento, seleciona-se objetos representativos de uma variedade de características e projeta-se a pesquisa para medir mais de um valor para cada característica, os quais serão comparados depois. Tais características são as variáveis medidas no experimento, que dão suporte à avaliação em questão. Experimentos são apropriados para investigar diferentes aspectos, como confirmar teorias, testar concepções de pessoas, explorar relações, avaliar a precisão de modelos, validar medidas, dentre outros. Além disso, podem ser efetuados em ambiente universitário antes de se fazer um estudo em indústria, reduzindo assim o custo e os riscos ([Linkman e Rombach \(1997\)](#) *apud* [Wohlin et al. \(1999\)](#)).

A escolha entre a condução de um estudo de caso ou experimento controlado é influenciada por diversos fatores. O principal fator é o nível de controle necessário para avaliar a questão investigada. Quando se tem um nível de controle alto sobre as variáveis que podem afetar a veracidade das hipóteses, então deve-se considerar a execução de um experimento. Se tal controle não é possível, é preferível um estudo de caso. Mesmo sendo possível controlar as variáveis, deve-se considerar o nível de dificuldade, custo e risco envolvidos. Outro aspecto a ser considerado é a viabilidade de replicar a situação básica que está sendo investigada. Se a replicação não é possível, ou se seu custo é proibitivo, então um experimento controlado pode

não ser recomendado ([Kitchenham et al., 1997](#)).

Ao analisar dados de experimentos, é necessário identificar suas características, como níveis de controle alcançados e influência de fatores externos. O uso de uma taxonomia pode auxiliar a generalização de resultados experimentais. Inicialmente, a taxonomia para estudos experimentais considerava estudos *in vivo* e *in vitro* ([Wohlin et al., 1999](#)).

Estudos experimentais *in vivo* envolvem pessoas em seu ambiente, e no caso de Engenharia de Software, trata-se de experimentos executados em organizações de desenvolvimento de software durante todo o processo de desenvolvimento e sob circunstâncias reais. Estudos de caso para a monitoração de projetos, que coletam dados ao longo do desenvolvimento, são exemplos dessa categoria, dada a característica essencialmente observacional. A dificuldade de isolar e controlar influências externas dificulta a análise e validação de hipóteses. Portanto, o tratamento de dados obtidos de modo observacional deve considerar a inexistência de controles, sendo adequados métodos de análise estatística multivariada, como regressão linear e análise de componentes principais (PCA) ([Wohlin et al., 1999](#)).

De forma geral, experimentos *in vitro* são executados em laboratórios, o que permite um alto grau de controle. Quando não é possível executar a associação aleatória de participantes a diferentes tratamentos, tem-se a *quasi-experimentação* ([Wohlin et al., 1999](#)). Em Engenharia de Software, experimentos dessa classe são normalmente executados em universidades e entre grupos selecionados de organizações de desenvolvimento de software ([Travassos e Barros, 2003](#)). O objetivo é variar uma ou mais variáveis e controlar todas as demais para medir o efeito da variação, aplicando métodos de inferência estatística para obter a significância das variações. Os conjuntos de dados utilizados como estudo de caso neste projeto de doutorado foram obtidos em estudos desta categoria.

A taxonomia inicial foi estendida por [Travassos e Barros \(2003\)](#), incluindo duas outras categorias – *in virtuo* e *in silico* – para acomodar estudos experimentais realizados com o apoio de modelos computacionais que os influenciam diretamente. Experimentos *in virtuo* envolvem a interação entre os participantes e modelos computacionais da realidade. Nestes experimentos o comportamento do ambiente com o qual os participantes interagem é modelado e representado por programas de computadores. O comportamento do modelo em um dado instante é descrito por um conjunto de valores de variáveis, ou conjunto de predicados, e o participante interage

com uma simulação que reproduz o comportamento do sistema – ambiente. Os autores apontam como principais vantagens dos experimentos *in virtuo* o baixo custo – comparado com *in vitro* e *in vivo* – e a facilidade de aplicação em situações reais (não laboratoriais). Sua aplicação é indicada em casos em que o objeto de estudo envolve atividades que perduram durante longos períodos, como meses ou anos, durante o projeto de desenvolvimento de software, e consequentemente a replicação de experimentos é proibitiva. Outra vantagem é a possibilidade de ter a coleta de dados parcial ou totalmente automatizada pelo programa que simula o modelo. Vale ressaltar que em vários experimentos a coleta de dados livre de erros pode ser uma tarefa difícil e limitante da validade. Entretanto, esta classe de experimentos possui uma forte restrição: a qualidade do modelo computacional e, indiretamente, a sua implementação. A qualidade de um modelo está relacionada sua à capacidade de representar fielmente o comportamento de elementos do mundo real. Além disso, o treinamento dos participantes quanto ao uso dos recursos computacionais que implementam o modelo, é outro importante fator de risco (Travassos e Barros, 2003).

Travassos e Barros apontam que o termo *in silico* pode caracterizar estudos executados completamente por meio de modelos computacionais. Ou seja, ambiente, objetos e participantes têm seus comportamentos representados por modelos que simulam características relevantes em estudo. Pode-se utilizar esta classe para designar ambientes de experimentação compostos de modelos numéricos que não requerem interação humana – uma vez selecionados os dados de entrada e iniciada a simulação, o modelo é executado e um conjunto de dados de saída é gerado para análise. Em relação a restrições de validade, deve-se considerar a dificuldade de obter um modelo de comportamento do participante. Diversas variáveis podem influenciar tal comportamento, como experiência, seriedade, aspectos culturais, dentre outros, e sua construção requer profundo conhecimento sobre as variáveis envolvidas.

2.2 O Processo de Experimentação

A avaliação da relação causa-efeito pode ser efetuada pela verificação de uma hipótese por meio de um experimento controlado, que busca avaliar a relação entre variáveis de entrada de um processo – as quais podem ser classificadas como **independentes** ou **dependentes**. Na realização de um experimento, manipula-se, de modo controlado, as variáveis independentes com o objetivo de observar o efeito das mudanças sobre as variáveis dependentes. Busca-se, portanto,

isolar fatores que podem interferir nas relações de causa-efeito, por meio da definição de um conjunto de variáveis independentes que mapeiam possíveis relações, no âmbito do objeto de avaliação. A realização de um experimento requer uma sequência de passos pré-estabelecida, definida em um processo capaz de guiar sua condução de modo a assegurar o controle preciso e sistemático. O processo de experimentação para Engenharia de Software Experimental proposto por [Wohlin et al. \(1999\)](#) é composto pelas seguintes fases: **Definição** – Define-se o experimento em termos de problema, objetivo e metas; **Planejamento** – Determina-se o projeto do experimento e a instrumentação e avalia-se possíveis ameaças para validação dos resultados; **Operação** – Executa-se o experimento coletando medidas; **Análise e Interpretação** – As medidas coletadas são analisadas e avaliadas; e **Apresentação e Empacotamento** – Os resultados são apresentados e as informações do experimento são disponibilizadas para uso futuro, por exemplo, para apoiar replicações. Na Figura 2.1 é apresentada uma visão geral do processo. Nela, os retângulos representam as fases – com as respectivas atividades envolvidas – e retângulos com cantos arredondados representam os produtos de cada fase. [Amaral et al. \(2003\)](#) propuseram que a atividade de *Empacotamento* seja executada durante o processo de experimentação e não como etapa final, aumentando a fidelidade do *Pacote de Laboratório* à condução do mesmo. O *Pacote de Laboratório* tem por objetivo manter: (i) a documentação de como um experimento controlado foi preparado, conduzido e analisado, de forma que se possa discutir os vários aspectos de experimentação e (ii) a explicação do experimento, para que o mesmo possa ser compreendido e repetido por outros pesquisadores. As exposições apresentadas neste trabalho adotam a proposta de Amaral et al., com as atividades de cada etapa incorporando as tarefas de empacotamento.

Na primeira etapa – Definição –, a partir do reconhecimento de uma necessidade, uma sentença estabelece a fundamentação do experimento declarando o problema a ser considerado. Deve-se estabelecer: o Objeto do Estudo (produtos, processos, recursos, etc), o Propósito (a intenção do experimento), o Foco (efetividade, custo, confiabilidade, etc), Perspectiva (desenvolvedor, cliente, pesquisador, etc) e Contexto (ambiente do experimento). Como pode ser observado na Figura 2.2 – seguindo a notação usada por Amaral et al. –, o reconhecimento de uma motivação para o experimento pode surgir de observações do mundo real ou a partir de análises de experimentos previamente conduzidos. Vale ressaltar que o resultado final desta fase

Figura 2.1: Processo de Experimentação (Wohlin et al., 1999)

consiste no início dos documentos apontados na figura, ou seja, os mesmos não encontram-se prontos nesta fase do processo.

Figura 2.2: Fase de Definição do Experimento

Em seguida, na fase de Planejamento, deve-se selecionar o Contexto de Experimentação considerando custos e riscos envolvidos, o que, em geral, leva à escolha de ambientes educa-

cionais ou de projetos em paralelo a projetos reais. Deve-se, também, formular as Hipóteses a serem testadas e selecionar as variáveis independentes e dependentes, juntamente com as respectivas escalas e medidas associadas. Tendo selecionado os Participantes, de forma representativa para a população e adequada ao experimento, pode-se então realizar o Projeto do Experimento, definindo o plano para a sua condução e estabelecendo a submissão dos participantes aos tratamentos estabelecidos. Métodos de análise estatística a serem aplicados aos dados devem ser estabelecidos, considerando as medidas e escalas adotadas. Os Instrumentos necessários à execução do experimento devem ser criados: artefatos de software, diretrizes para os participantes e questionários. Por exemplo, o questionário para coleta de dados relativos ao perfil de cada participante é preparado considerando as hipóteses que se deseja verificar e o mesmo se aplica a outros documentos de apoio. Vale ressaltar que todas as atividades desta fase são executadas à luz dos resultados da fase anterior, como mostra o *workflow* da Figura 2.3. Como última atividade deve ser efetuada uma análise de adequação do conjunto de dados definido para o experimento (que serão coletados durante a Execução), com relação ao contexto, às hipóteses estabelecidas, às variáveis, aos participantes, ao projeto e aos instrumentos, deve ser efetuada e, se preciso, esse processo pode ser repetido buscando a completitude necessária.

Figura 2.3: Fase de Planejamento do Experimento

A fase de Operação, apresentada na Figura 2.4, consiste na execução do experimento propriamente dita. Nessa fase, documentos gerados na *Definição* e no *Planejamento* – Projeto

Figura 2.4: Fase de Execução do Experimento

Experimental, Procedimento do Experimento, Artefatos do Experimento e de Software e Glossário – são utilizados para a condução do experimento. No final dessa fase observa-se um ponto de controle, que permite o retorno às etapas anteriores, encerrando a execução apenas se os dados coletados forem aprovados na validação. A idéia é não permitir, sem o devido tratamento, dados incompletos, ou mesmo inexistentes, fornecidos pelos participantes.

A etapa de *Análise e Interpretação* consiste em classificar os dados coletados e aplicar técnicas estatísticas, como medidas de dispersão e análise de variância (ANOVA). As atividades que integram essa etapa são apresentadas na Figura 2.5. O objetivo de testar uma hipótese é verificar se é possível rejeitá-la, ou não, com base em amostra de alguma distribuição estatística, considerando alguma significância. Como os resultados dependem diretamente do conjunto de dados de entrada, pode ser interessante restringir o conjunto de dados e avaliar um subconjunto adequado à análise. Os dados podem ser selecionados considerando tanto a execução do experimento quanto o dado colhido. No primeiro caso, o objetivo é descartar os dados não confiáveis, e no segundo, o descarte pode ser justificado caso se verifiquem discrepâncias. Também pode ser vantajoso excluir dados redundantes. As técnicas estatísticas indicadas por Wohlin e

colaboradores têm sido aplicadas em análises confirmatórias e, embora permitam a verificação quantitativa das hipóteses propostas nos experimentos, introduzem algumas dificuldades e limitações. Dentre elas pode-se citar a dificuldade de combinar em uma análise dados de tipos distintos – por exemplo, nominal e ordinal –, uma vez que diferentes técnicas são indicadas para cada um dos tipos de dados (Wohlin et al., 1999); dificuldade de interpretação dos resultados obtidos, pois é necessário conhecimento do critério de seleção dos dados e da técnica aplicada; dificuldade em consolidar o conhecimento a partir dos resultados; dificuldade, e até inviabilidade, de conduzir testes de hipóteses baseados em permutação de indivíduos (Basili et al., 1996c). Uma vez analisados e interpretados os dados do experimento, deve-se registrar os resultados, as conclusões e possíveis recomendações para uso futuro.

Figura 2.5: Fase de Análise e Interpretação dos Dados Coletados

2.3 Papéis da Replicação

Segundo Basili et al. (1999), a formação do corpo de conhecimento em Engenharia de Software requer a execução de famílias de experimentos e adoção de um conjunto de princípios unificados que permita combinar e generalizar resultados. A replicação de experimentos em diferentes ambientes culturais permite não só confirmar, ou não, hipóteses sob condições diferentes das inicialmente propostas, como também formular novos questionamentos. Replicar um experi-

mento é de extrema importância, pois fornece uma estimativa de erro experimental capaz de embasar uma avaliação das diferenças observadas em uma variável independente. Ou seja, a replicação permite avaliar melhor o grau de segurança dos resultados do experimento. Além disso, a replicação permite estimar o efeito principal de qualquer fator experimental (Kitchingham et al., 1997). Miller (2005) também aponta três problemas potenciais com experimentos isolados: baixa força estatística; amplo número de potenciais influências sobre variáveis sob tratamento; e a verificação do processo e produtos do estudo.

Ainda segundo Basili et al. (1999), existem três tipos de replicações, cujos objetivos variam. São eles:

1. Replicações em que nenhuma hipótese de pesquisa varia

As variáveis dependentes ou independentes do experimento original não variam. Podem ser:

- (a) **Replicações estritas:** repetem o experimento original tão precisamente quanto possível. São necessárias para aumentar a confiança na validade da conclusão do experimento.
- (b) **Replicações que variam a execução:** Variam a maneira na qual o experimento é executado. Aumentam a confiança em resultados experimentais testando as mesmas hipóteses, mas alterando detalhes para tratar determinadas ameaças internas à validade.

2. Replicações em que as hipóteses de pesquisa variam

Variam atributos do processo, produto e modelos de contexto, mas permanecem no mesmo nível de detalhamento que o experimento original. Podem ser:

- (a) **Replicações que variam as variáveis independentes:** investigam que aspectos do processo são importantes como propriedades intrínsecas, variando o processo e examinando os resultados.
- (b) **Replicações que variam as variáveis dependentes:** podem variar os modos nos quais a efetividade está sendo medida, para entender em que dimensões de uma tarefa um processo trará melhores resultados.

- (c) **Replicações que variam as variáveis de contexto:** variação de contexto no ambiente no qual a solução é avaliada. Podem identificar fatores ambientais importantes que afetam os resultados do processo sob investigação e assim, ajudar a entender sua validade externa.

3. Replicações que estendem a teoria

Ajudam a determinar os limites de efetividade de um processo, fazendo grandes mudanças no processo, produto e/ou modelo de contexto, para verificar se os princípios básicos ainda permanecem.

Replicações de experimentos são essenciais não só para obter resultados comparativos ao experimento original, mas também para gerar e facilitar pesquisas experimentais colaborativas, transferir experiência na execução de experimentos e replicações, obter volume de dados para meta-análise e melhorar os artefatos do pacote experimental (Shull et al., 2002, Maldonado et al., 2006). A necessidade de conduzir um mesmo estudo em diferentes contextos para formar um corpo de conhecimento confiável introduz complicadores adicionais no processo de experimentação. Um aspecto fundamental para a condução de replicações é o uso do conhecimento nelas adquirido é um registro preciso do procedimento de condução do experimento, o qual é registrado no chamado Pacote de Laboratório, representado na Figura 2.6. Na figura as Regras referem-se à condução do experimento, como distribuição de participantes em equipes e documentos, e tempo de treinamento; os Documentos referem-se aos documentos utilizados, tanto para treinamento quanto para avaliação; Artefatos de Software são os artefatos a serem utilizados, previamente preparados; e as Técnicas referem-se não só à técnica propriamente dita, mas também ao treinamento a ser aplicado na técnica. Outra dificuldade encontrada diz respeito à habilidade de compartilhar conhecimento entre replicadores, que é fundamental para a execução de replicações e sua melhoria. Segundo Shull et al. (2002), ao replicar um experimento, além de verificar hipóteses, os experimentadores devem: facilitar a colaboração entre parceiros; transferir conhecimento sobre a execução do experimento e replicações; conduzir meta-análises; e melhorar e empacotar artefatos experimentais para futuras replicações. Além disso, há a possibilidade de combinar dados de vários experimentos para a condução de análise exploratória. Nesse contexto, faltam diretrizes que apoiem o registro e manutenção de informações explícitas e no nível de granularidade adequado.

Figura 2.6: Replicações de um experimento e o Pacote de Laboratório

Também é necessário estabelecer critérios e diretrizes para a análise de dados oriundos de replicações distintas. O mesmo vale para dados coletados em diferentes estudos, mas que apresentam características mínimas que permitam a sua junção, dependendo do objetivo da análise. A inexistência de diretrizes para o empacotamento e de mecanismos que facilitem a colaboração, e as dificuldades na transferência de conhecimento e na condução de análises exploratórias, dentre outros, são problemas a serem tratados na experimentação em Engenharia de Software. Observa-se a ausência de iniciativas de uso de Visualização nesse contexto. A Visualização de Informação tem potencial para contribuir na Engenharia de Software Experimental em diversos níveis de aprendizado. O primeiro deles consiste no uso de técnicas de visualização para apoiar a análise de dados de um experimento, de modo complementar às tradicionais técnicas estatísticas. O segundo ponto de contribuição é na análise exploratoria, atividade fundamental para criar um corpo de conhecimento sobre o tema em estudo, em que novas hipóteses podem ser levantadas e novas diretrizes estabelecidas para a realização de experimentos, contribuindo para a evolução do Pacote de Laboratório. À medida em que o experimento e o Pacote de Laboratório evoluem, a análise dos dados coletados nas múltiplas replicações exige cuidados específicos que dependem da uniformidade entre as replicações. Além disso, é necessário estabelecer critérios para a evolução de experimentos que tenham como foco a qualidade da experimentação e que explorem aspectos complementares de estudos e abordagens, como exposto a seguir.

2.3.1 Arcabouço para Melhoria da Experimentação

Mendonça et al. (2006) propuseram um arcabouço para o processo de experimentação (FIRE – *Framework for Improving the Replication of Experiments*), apresentado na Figura 2.7, que visa a gerenciar o conhecimento e melhorar replicações de experimentos controlados em engenharia de software.

Figura 2.7: Ciclos do FIRE (Mendonça et al., 2006)

O ciclo de *Execução do Experimento* (ciclo menor) corresponde à execução do experimento em estudo. O ciclo de *Aprendizado Intra-Grupo* (ciclo do meio) retrata o aprendizado, empacotamento e planejamento de novos experimentos por um mesmo grupo de pesquisadores, visando a construir um corpo de conhecimento local. As atividades desse ciclo consistem em:

- Etapa 1: Definir metas do experimento;
- Etapa 2: Planejar o experimento, identificar participantes e obter/adaptar artefatos: preparação do experimento;
- Executar experimento (ciclo de Execução do Experimento);

- Etapa 3: Executar meta-análise: analisar resultados experimentais em conjunto com outros dados experimentais disponíveis;
- Etapa 4: Criar/Evoluir pacote de laboratório reutilizáveis, armazenando experiência e aprendida.

O ciclo de *Aprendizado Inter-Grupo* (ciclo maior) tem como foco a padronização do Pacote de Laboratório, a evolução do pacote de experimentação e o compartilhamento de conhecimento entre os grupos de pesquisa, visando a construir um amplo corpo de conhecimento sobre um tema baseado em replicações executadas por diferentes grupos de pesquisa. As atividades chave são:

- Etapa 1: Planejamento e coordenação das atividades experimentais entre os diferentes grupos;
- Etapa 2: Compreensão dos pacotes de laboratório e dos resultados experimentais disponibilizados por outros grupos de pesquisa;
- Execução de uma série de replicações e evolução do pacote de laboratório local (ciclo de *Aprendizado Intra-grupo*);
- Etapa 3: Compartilhamento e consolidação do aprendizado com outros grupos;
- Etapa 4: Harmonização do conhecimento: uniformizar o conhecimento adquirido pelos grupos;
- Etapa 5: Criação/Evolução do corpo de conhecimento (repositório de conhecimento).

A externalização e internalização do conhecimento na forma de pacotes de laboratório e a “materialização” do conhecimento tácito – que não está inserido nos pacotes – são os pontos críticos do FIRE. Os autores apontam a ausência de diretrizes que apoiem o registro e manutenção de informações explícitas, e referenciam o Pacote de Laboratório como o corpo de informação explícita sobre um experimento para atingir a uniformidade necessária entre replicações. Consideram essencial o contato entre replicadores e o projetista de um experimento para a transferência de conhecimento tácito, já que não é trivial a transformação de conhecimento tácito em explícito e vice-versa. Entretanto, o FIRE não aborda como contornar esses problemas.

Nesse contexto, vários pontos apresentados na Figura 2.7, como a *Analisar e integrar Dados* e *Criar/Evoluir Corpo de Conhecimento*, precisam ser tratados. Pode-se observar também que os ciclos e suas atividades reconhecem a natureza evolutiva da aquisição de conhecimento, tanto sobre o objeto em foco (o estudo propriamente dito) como sobre o procedimento da experimentação. Entretanto, faltam na literatura da área mecanismos que apóiem mais adequadamente a evolução do conhecimento.

2.3.2 Meta-Análise e a Análise de Dados Combinados

Como discutido, um único estudo envolve vários parâmetros, alguns controlados, outros completamente não delimitados – ou seja, podem até ser conhecidos, mas não se tem noção de seu alcance e influência. A realização de replicações é um mecanismo propício para tentar identificar, isolar, e/ou entender os efeitos de tais parâmetros. Segundo Miller (2005), apesar da comparação de resultados de experimentos ter o objetivo de construir um corpo de conhecimento a respeito das hipóteses em questão, um objetivo secundário deve ser considerado: mensurar a validade da abordagem original do experimento. Considerando o objetivo principal, caso as conclusões obtidas em diferentes replicações sejam concordantes – ditos suportivos – as abordagens são mutuamente validadas. Neste caso, pode-se agregar os resultados e expandir o corpo de conhecimento sobre as hipóteses. Entretanto, caso as conclusões dos experimentos sejam diferentes, é necessário buscar uma explicação para tal variação.

A meta-análise tem como objetivo o entendimento de estudos e suas replicações (Miller, 2005). Tradicionalmente, ela tem como foco as hipóteses dos experimentos cujos resultados serão comparados, sendo necessário combinar **resultados** de estudos que tratam de hipóteses similares. Por outro lado, Miller enfatiza que os experimentos a serem combinados devem ser independentes para permitir a generalização das conclusões, ou seja, para garantir significância estatística os estudos devem ser estatisticamente independentes. Tal fato impõe restrições à aplicação de procedimentos meta-analíticos tradicionais, principalmente com relação às replicações de experimentos que usam o mesmo projeto experimental e os mesmos materiais.

Entretanto, nos casos em que as hipóteses não são conhecidas *a priori* ou nos casos em que se conhece as hipóteses, mas estas têm fraco apoio estatístico, a abordagem de processos meta-analíticos tradicionais pode não ser efetiva. Por exemplo, se pesquisadores estiverem interessados em explorar conjuntos de dados coletados em um ou mais experimentos, ou repli-

cações, com o objetivo de descobrir relações não antecipadas e identificar novas hipóteses, uma abordagem dirigida por hipóteses prévias não será efetiva. Neste caso, é interessante fazer uma análise exploratória, que permita explorar cenários.

A exploração de tais cenários, por não ser limitada às hipóteses conhecidas, sugere uma análise exploratória dos dados coletados nos experimentos, tanto de modo isolado como combinado. A combinação de dados de múltiplas replicações não só possibilita o aumento do potencial estatístico das observações (quando garantida a validade estatística), mas principalmente aumenta o volume de dados disponível para análise, o que facilita o uso de estratégias alternativas como as técnicas de visualização introduzidas no Capítulo 3. Entretanto, as variações entre replicações e experimentos tornam-se obstáculos à combinação natural e direta de dados de diferentes replicações, introduzindo um requisito fundamental tanto para a combinação quanto para a condução da análise exploratória propriamente dita: a combinação dos dados.

2.3.3 Geração de Hipóteses e Lista de Variáveis

O primeiro passo em um experimento é a própria sentença que o define, como apresentado na Seção 2.2. Tal definição é motivada por alguma necessidade observada. Em geral, pesquisadores têm como motivação a verificação de suas técnicas, com o intuito de obter indícios de vantagens e desvantagens. Tão importante quanto a definição das hipóteses é a definição de uma lista de variáveis que apoiem a investigação das hipóteses. Carver (2003) aponta que são raros os pesquisadores que investigam não apenas suas técnicas, mas também o contexto de sua aplicação, talvez pela dificuldade de obter uma lista de variáveis que descrevam o contexto. Para facilitar o estabelecimento de hipóteses e a verificação do experimento, antes da Execução, é possível separar as variáveis em duas classes distintas: variáveis de foco e variáveis de contexto. As *variáveis de foco* são aquelas diretamente relacionadas às perguntas que o experimento tenta responder, e são obtidas a partir de hipóteses do tipo *a técnica de revisão A é melhor do que a técnica B para revelar defeitos*. As *variáveis de contexto* descrevem o contexto do experimento, definindo as condições em que as respostas obtidas são válidas. Em casos em que as hipóteses são mais restritivas, como *a técnica de revisão A é melhor do que a técnica B para revelar defeitos em documentos de requisitos*, considera-se o foco mais restrito.

Carver (2003) propôs um processo de identificação de variáveis e hipóteses baseado no conhecimento obtido com revisão de literatura³, experiência prévia de especialista da área sob investigação e dados experimentais, como exposto na Figura 2.8. Observa-se que o ponto de partida é a revisão da literatura e a experiência prévia, que apóiam o estabelecimento de hipóteses e a lista inicial de variáveis. Vale ressaltar que a revisão de literatura, para tal fim, não tem como foco apenas estudos em Engenharia de Software Experimental, mas também outras em áreas, como Sociologia e Psicologia, já que questões importantes como o modo que as pessoas aprendem, podem influenciar os resultados. Se o treinamento é um fator importante que pode influenciar o resultado, sua definição clara e suficiente se faz necessária.

Figura 2.8: Metodologia para o desenvolvimento de lista de variáveis (Carver, 2003)

O segundo ponto a ser considerado é o uso de dados de estudos já realizados para apoiar o refinamento da lista de variáveis e hipóteses, indicado pelo ciclo à direita na Figura 2.8. Neste caso, deve-se considerar o conjunto de experimentos que exploraram questões relacionadas ao experimento em planejamento e analisar os dados neles coletados. Durante essa análise, pode-se

³Embora o autor não tenha identificado o procedimento para tal revisão, indica-se aqui o emprego de Revisão Sistemática.

concluir que as hipóteses inicialmente estabelecidas podem ser “melhoradas”, considerando as variáveis de outros experimentos: (i) algumas variáveis não têm influência na hipótese e, portanto, podem ser retiradas; (ii) variáveis podem ser incluídas, restringindo o escopo da hipótese.

No ciclo da esquerda da Figura 2.8 é apresentada a possibilidade de refinar hipóteses pela condução de experimentos com o intuito específico de avaliar hipóteses existentes. Neste caso, diante de um pequeno conjunto de hipóteses previamente postulado, prepara-se um experimento para verificá-las, e conseqüentemente, há a possibilidade de utilizar os dados obtidos para refinar tanto a lista de variáveis quanto a lista das hipóteses estabelecidas.

No processo proposto, o passo *Refinar Variáveis e Hipóteses* corresponde a agregar os dados para a análise, considerando os resultados estatísticos, o que pode levar a situações como: a) os dados geram resultados concordantes com a lista de variáveis existente; b) dados são contrários à lista de variáveis existente; c) itens da lista de variáveis podem permanecer sem suporte; d) os dados podem levar à adição de novos itens à lista. Seja qual for o caso, as análises têm como foco o apoio estatístico, ou seja, o objetivo é a análise estatística das hipóteses existentes, e não uma exploração não direcionada dos dados. Entretanto, o apoio de um processo exploratório não direcionado pode aumentar a eficácia da análise e, conseqüentemente, do refinamento da lista de variáveis e hipóteses. Pode-se observar que tanto a replicação (como exposto nesta seção), quanto a análise exploratória (usando dados combinados) também podem contribuir no processo de refinamento da lista de variáveis e hipóteses.

2.4 Experimentos Realizados em VV&T

Estudos baseados em experimentos controlados em Engenharia de Software têm sido realizados por diversos pesquisadores. [Hetzl \(1976\)](#) executou um experimento controlado com o objetivo de comparar três técnicas de detecção de defeitos, no que diz respeito à efetividade em revelar falhas. [Myers \(1978\)](#) realizou um experimento parecido, observando, porém, cada participante uma única vez aplicando uma técnica.

[Basili e Selby \(1987\)](#) executaram um experimento controlado com o mesmo objetivo, comparando efetividade e eficiência em revelar falhas de três técnicas de detecção de defeitos. Esse experimento foi conduzido por três vezes, sendo as duas primeiras execuções realizadas com estudantes, e a terceira contou com a participação de desenvolvedores profissionais. [Kamsties e Lott \(1995b\)](#) estenderam o projeto e as técnicas originalmente utilizadas por [Basili e Selby](#)

(1987), incorporando uma atividade para isolar defeitos após a observação de falhas. Wood et al. (1997) replicaram o experimento de Kamsties e Lott (1995b). Os artefatos de software usados nos experimentos citados foram códigos fonte.

Para fornecer uma base tecnológica à leitura de artefatos de software, Basili et al. (1996a) apresentam técnicas específicas de leitura, compostas de um conjunto concreto de instruções dadas ao leitor indicando como ler ou o que procurar em um artefato. Seus esforços focalizam o desenvolvimento de famílias de técnicas da leitura, baseadas na avaliação experimental. Cada família de técnicas de leitura pode ser instanciada para uso em contextos diferentes, devendo, conseqüentemente, ser avaliadas no contexto apropriado. A taxonomia hierárquica de famílias de leitura é mostrada na Figura 2.9. A parte superior da árvore (acima da linha horizontal tracejada) modela os problemas que podem ser tratados pelas técnicas de leitura. Na parte inferior são apresentadas as possíveis soluções para cada caso.

Figura 2.9: Família de Técnicas de Leitura (Basili et al., 1996a)

Nesse contexto, para avaliar a técnica de leitura baseada em perspectiva – *Perspective-Based Reading* (PBR) – Basili et al. (1996c) executaram um experimento que deu origem a uma série de replicações e que foi utilizado como referência no Projeto *Readers*, cujo histórico é apresentado a seguir.

2.4.1 Histórico dos Experimentos PBR

Basili et al. (1996c) executaram um experimento controlado com o objetivo de avaliar a efetividade da técnica de leitura *Perspective Based Reading* (PBR) na detecção de defeitos, quando aplicada a documentos de especificação de requisitos. *Perspective Based Reading* (PBR) é uma técnica de leitura cuja motivação está relacionada diretamente com as responsabilidades de revisão. Enquanto as técnicas de leitura normalmente não definem responsabilidades específicas para os revisores, a técnica PBR requer que o revisor crie uma abstração de um produto de software para, então, responder questões baseadas na análise da abstração considerando papéis diferentes. Esses papéis definem pontos de vista diferentes que o revisor pode assumir, ou seja, perspectivas diferentes. Um revisor pode assumir a perspectiva do desenvolvedor, do testador ou do usuário final do sistema (*designer*, *tester* ou *user*). Para cada uma das perspectivas existe um cenário definido, com questões e atividades que estabelecem o que fazer e como ler um documento (Shull et al., 2000). As questões e atividades habilitam o descobrimento de defeitos em um artefato de software.

O experimento teve como participantes desenvolvedores profissionais de software do *Software Engineering Laboratory* (SEL), da *National Aeronautics and Space Administration / Goddard Space Flight Center* (NASA/GSFC). Os participantes leram dois tipos de documentos, um genérico e outro do domínio específico da NASA, usando duas técnicas de leitura: PBR e uma técnica usualmente aplicada pelos participantes. O experimento destinou-se a responder às seguintes questões:

1. Se fossem dadas apenas as regras PBR aos grupos de inspeção, como em uma sessão de inspeção, um número maior de defeitos seria detectado do que seria detectado caso os grupos de revisão utilizassem técnicas convencionais?
2. Se os participantes revisassem um documento utilizando PBR, um número diferente de defeitos seria detectado do que aqueles detectados caso os participantes utilizassem técnicas convencionais?
3. A experiência do participante em uma das perspectivas (Projeto, Teste, Uso) tem influência sobre o seu desempenho?

Os documentos para a realização do experimento foram preparados introduzindo defeitos comuns ao seu ambiente, sendo que dois documentos de domínio específico da NASA foram preparados. Derivados de um conjunto de documentos de especificação de requisitos já existente, especificamente de aplicações de dinâmicas de voo, foram identificados como NASA_A e NASA_B. Também foram preparados dois documentos de domínio genérico para serem utilizados em outros contextos. São eles: *Parking Garage Control System* (PG) e *Automatic Teller Machine* (ATM). Outros dois documentos foram elaborados para treinamento, um genérico (*Video Rental System*) e um específico (*NASA sample*). Na Tabela 2.1 é apresentada a distribuição de defeitos semeados por documento.

Tabela 2.1: Distribuição de Defeitos por Documento (Basili et al., 1996c)

Domínio	Documento	Páginas	Defeitos
Genérico	Automatic teller machine (ATM)	17	29
	Parking garage control system (PG)	16	27
	Video rental system	14	16
Específico	Flight dynamics (NASA_A)	27	15
	Flight dynamics (NASA_B)	27	15
	NASA sample	9	6

A distribuição dos documentos entre participantes foi feita de forma que cada participante revisasse todos os documentos apenas uma vez. Tal distribuição possibilita a comparação entre as técnicas sem deturpar o resultado pelo prévio conhecimento adquirido por intermédio de múltiplas aplicações de técnicas a um mesmo documento. Os indivíduos foram divididos em dois grupos – Grupo 1 e Grupo 2 – e cada grupo subdividido em 3 subgrupos vinculados às perspectivas do PBR (*Designer*, *Tester* e *User*). A organização é exposta na Figura 2.10.

	Grupo 1			Grupo 2			
	Designer	Tester	User	Designer	Tester	User	
Técnica Usual	Treinamento			Treinamento			Primeiro Dia
	NASA_A			NASA_B			
	Treinamento			Treinamento			
	ATM			PG			
Técnica PBR	Teoria sobre a técnica PBR						Segundo Dia
	Treinamento			Treinamento			
	PG			ATM			
	Treinamento			Treinamento			
	NASA_B			NASA_A			

Figura 2.10: Projeto do Experimento (Basili et al., 1996c)

No primeiro dos dois dias de duração do experimento, os participantes utilizaram a sua

técnica convencional para revisar os documentos, tanto os de domínio específico (NASA_A e NASA_B) quanto os de domínio genérico (ATM e PG). No segundo dia, participantes separados de acordo com os subgrupos foram treinados em uma das perspectivas da técnica PBR, antes de revisar os documentos de domínio genérico e os de domínio específico. A composição dos grupos por documento garantiu que o grupo que utilizou o documento NASA_A aplicando sua técnica usual utilizou NASA_B para aplicar uma perspectiva do PBR, e vice-versa. O mesmo tratamento foi dado para os documentos genéricos. Antes de iniciar cada sessão de revisão, os indivíduos foram instruídos sobre como proceder utilizando determinada técnica.

Um estudo piloto foi executado em Novembro de 1994, tendo sido observados os seguintes problemas ([Basili et al., 1996c](#)):

1. Tentou-se alocar os participantes para cada perspectiva de acordo com suas experiências. Entretanto, a maioria deles era desenvolvedor de software, inviabilizando uma divisão adequada. Decidiu-se que as separações seriam randômicas em futuras replicações;
2. Os documentos de domínio específico da NASA foram considerados longos para análise em uma única sessão. Decidiu-se torná-los menores em futuras replicações;
3. Cada participante teve até 3 horas para revisar cada documento. Entretanto, apenas um participante levou mais de 2 horas, sendo considerado um limite de 2 horas por documento efetivo em futuras replicações;
4. Foram efetuadas sessões de treinamento apenas para documentos genéricos, mas os participantes indicaram que treinamento sobre documentos de domínio específicos seriam adequadas. Conseqüentemente, em futuras replicações, a inserção de sessões de treinamento antes de cada revisão deveria ser considerada, tornando necessário o documento de treinamento do domínio específico;
5. Foi permitido que os participantes trabalhassem em suas próprias salas, ficando sujeitos a interrupções, chamadas telefônicas e outras eventualidades. Acredita-se que a validade dos resultados seria ampliada se fossem utilizados ambientes uniformes em futuras replicações;
6. Algumas inconsistências foram descobertas nas especificações de requisitos. As especificações consistem de uma descrição geral do problema seguida de uma série de especifi-

cações precisas, algumas das quais estavam intencionalmente incorretas. Decidiu-se que, em futuras replicações, a descrição geral da especificação deve estar correta para que os participantes pudessem ter uma base adequada para tomar decisões.

Vale destacar que, entre os problemas relatados, percebe-se uma potencial contradição entre os itens 2 e 3: em (2) não se sabe porque os documentos específicos foram considerados longos para uma análise apropriada, já que o tempo alocado foi mais que suficiente (3).

Devido aos problemas relacionados acima uma nova execução do experimento foi realizada em Junho de 1995, com 14 participantes. Entretanto, um deles não estava familiarizado com aplicações de dinâmica de vôo, sendo considerados apenas 13 participantes na análise dos documentos específicos – NASA_A e NASA_B. Os resultados obtidos são expostos na Seção [2.4.2](#).

2.4.2 Resultados Obtidos por Análise Estatística

Para a análise estatística, os defeitos registrados em formulários específicos devem ser preparados e planilhados. coleta dos dados em formulários com layout previamente estabelecido (como mostra a Figura [2.11](#)), mas com possibilidade de escrita em texto livre, permite o registro de defeitos sem formato padrão, tornando necessária uma pré-análise, que consiste em classificar cada defeito reportado em defeito real, falso positivo ou descartá-lo.

Observa-se na Figura [2.11](#) a presença de uma coluna na qual os participantes devem classificar cada defeito encontrado de acordo com a taxonomia estabelecida. Tais classes não são definitivas, pois podem ser alteradas para adequar a taxonomia ao ambiente e/ou domínio de aplicação. Por exemplo, [Shull et al. \(2000\)](#) apresentam uma taxonomia para defeitos em documentos de requisitos, descrita como segue:

- *Ambiguous Information*: múltiplas interpretações causadas pelo uso de termos diferentes para a mesma situação ou múltiplos significados de um termo em um contexto particular;
- *Extraneous Information*: Informação que é fornecida, mas não é necessária ou utilizada;
- *Inconsistent Information*: Duas declarações contraditórias;
- *Incorrect Fact*: Alguma sentença contida na especificação de requisitos afirma um fato que não pode ser verdade sobre as condições especificadas;

Execution - Form E4Reviewer ID: _____ Date: ____/____/____Page _____ of _____

Document name: _____

Defects			
# Defect	Page	Class	Description

Figura 2.11: Formulário de Registro de Defeitos Encontrados (Basili et al., 1998)

- *Missing Information*: Informação necessária sobre o sistema não incluída no documento de especificação de requisitos.
- *Miscellaneous Defects*: outros defeitos, como inserir um requisito em uma seção errada, por exemplo.

Como observado no experimento inicial de Basili et al., as replicações de experimentos trazem lições que podem influenciar futuras replicações. Dessa forma, pode-se considerar que está sendo realizado um experimento diferente, ao invés de uma simples replicação, dependendo do grau de alteração devido a lições aprendidas e conseqüentes alterações no pacote de laboratório que guia a replicação. Isso, inclusive, introduz restrições à análise dos dados, mas não a inviabiliza, desde que tomados os devidos cuidados.

A partir das planilhas que sumarizam os dados coletados, os resultados das análises, como efetividade e eficiência, são expressos quantitativamente em percentual e, em geral, uma análise de variância (ANOVA) é aplicada. A análise estatística varia de acordo com as hipóteses a serem verificadas. Por exemplo, [Basili et al. \(1996c\)](#) analisaram a taxa de cobertura que poderia resultar de uma equipe de revisão aplicando PBR, simulando equipes compostas de um revisor

de cada perspectiva. Realizaram também análises por participante e quanto à perspectiva, como apresentado a seguir.

2.4.2.1 Análise por Equipe

Considerando que as perspectivas do PBR (*Designer*, *Tester* e *User*) são específicas, supõe-se que a união das perspectivas ofereça uma maior cobertura do documento em análise. Há, assim, a necessidade de combinar as perspectivas dentro de um mesmo grupo de revisão. Basili et al. utilizaram os dados coletados no experimento da NASA para analisar taxa de cobertura obtida pelos grupos de revisão. Os resultados dos grupos foram simulados a partir da união dos resultados individuais dos seus integrantes, depois da conclusão do experimento, sem que tenha ocorrido qualquer interação entre os integrantes durante o experimento (já que o interesse não era verificar aspectos como a integração entre os integrantes do grupo).

Os revisores do Grupo 1 aplicaram sua técnica usual ao *Documento A* e PBR ao *Documento B*, sendo *Documento A* e *Documento B* representantes dos dois documentos de qualquer um dos domínios. Pôde-se, assim, gerar o conjunto de todas as equipes possíveis dos sujeitos que não aplicaram PBR para o *Documento A* e o conjunto de todas as equipes possíveis de revisores que aplicaram PBR para o *Documento B*, e examinou-se a cobertura de defeitos de equipes de cada técnica examinando a taxa média da detecção de defeitos de cada grupo. Isso garante que os resultados sejam independentes de qualquer escolha arbitrária de membros da equipe, mas como os pontos de dados para todas as equipes possíveis não são independentes (por exemplo, cada revisor aparece múltiplas vezes nessa lista de todas as equipes possíveis), não é possível efetuar testes estatísticos nesses valores médios. Chamou-se então essas médias de A_{1USUAL} e B_{1PBR} . Foram efetuados os mesmos cálculos para o Grupo 2, em que os revisores aplicaram sua técnica usual ao *Documento B* e PBR ao *Documento A*, a fim de obter as médias A_{2PBR} e B_{2USUAL} . O teste estatístico

$$(A_{2PBR} - A_{1USUAL}) + (B_{1PBR} - B_{2USUAL})$$

provê uma medida de como todas as possíveis equipes PBR teriam executado em relação a todas as equipes que aplicaram a técnica usual, para cada documento. Na Figura 2.12 são apresentadas as médias do PBR comparadas às médias da técnica “Usual”.

Observa-se na figura que, na primeira execução do experimento (Estudo Piloto), o grupo PBR obteve resultados melhores, embora isso não tenha sido significativo nos documentos de

Figura 2.12: Resultados da análise dos times simulados (Basili et al., 1996c)

domínio específico. Na segunda execução do experimento, os grupos que utilizaram PBR tiveram melhor desempenho em ambas classes de documentos. A razão para essa melhora observada entre a primeira e a segunda execução do experimento pode ser atribuída à reformulação dos documentos de domínio específico e às sessões de treinamento que antecederam cada sessão de revisão dos documentos.

2.4.2.2 Análise por Participante

Embora o objetivo principal do experimento tenha sido investigar os efeitos da técnica PBR em equipes de revisão, comparou-se também o desempenho individual dos participantes na detecção de defeitos aplicando a técnica PBR com o obtido aplicando outras técnicas, observando o percentual de defeitos detectados por cada indivíduo em relação ao número total de defeitos existentes no documento inspecionado.

Basili et al. analisaram separadamente o desempenho individual, em função da técnica aplicada (Usual e PBR) e do domínio do documento. Em muitos casos, os participantes que aplicaram uma perspectiva da técnica PBR descobriram o mesmo número de defeitos que participantes utilizando técnicas usuais. No entanto, os participantes que aplicaram PBR tiveram um desempenho significativamente melhor nos documentos de domínio genérico na segunda

execução do experimento, conforme pode ser observado na Figura 2.13. Este resultado foi, de certa forma inesperado, uma vez que os benefícios reais da técnica PBR são esperados para grupos de revisão, pelo fato de combinarem as perspectivas. Os resultados mostram que, sob certas circunstâncias, o uso de técnicas sistemáticas pode melhorar, também em termos de percentuais individuais, a cobertura do documento.

Figura 2.13: Resultados condensados de Basili et al. (1996c)

2.4.2.3 Análise das perspectivas

Outro objeto de interesse na análise consistiu em determinar se indivíduos utilizando PBR descobriram classes mais amplas de defeitos e se os defeitos detectados com as diferentes perspectivas eram ortogonais (isto é, as perspectivas não se sobrepuseram em relação ao conjunto de defeitos que ajudaram a descobrir). Um exame da cobertura de defeitos de cada perspectiva e de como as perspectivas se sobrepuseram indicou que houve um número considerável de defeitos descobertos por uma das perspectivas em documentos de domínio genérico, diferentemente dos documentos específicos, nos quais observou-se maior grau de sobreposição entre as perspectivas.

Na Figura 2.14 são apresentados os números para um documento de domínio genérico (ATM) a título de ilustração. Observa-se que participantes utilizando a técnica PBR na perspec-

tiva do projetista encontraram 11 defeitos na segunda execução do experimento, sendo que dois deles não foram detectados por nenhuma outra perspectiva, três foram detectados pela perspectiva de testador, um pela perspectiva do usuário e cinco defeitos foram detectados por pelo menos um participante de cada uma das três perspectivas.

ATM

Figura 2.14: Cobertura de defeitos por perspectiva para o documento ATM (Basili et al., 1996c)

2.5 O Projeto *Readers*

O Grupo de Engenharia de Software do Instituto de Ciências Matemáticas e de Computação (ICMC–USP), em colaboração com a University of Maryland, Universidade Federal de São Carlos, Universidade Federal do Rio de Janeiro-COPPE e Universidade de Salvador (Unifacs), desenvolveu o projeto *Collaborative Research to Develop, Validate and Package Reading Techniques for Software Defect Detection (NSF-CNPq Readers)*. As principais metas desse projeto são:

- Criar tecnologias para apoiar a detecção de defeitos em diferentes tipos de documentos de software e desenvolver métodos experimentais que possam contribuir para melhorar essas tecnologias para uso em diferentes ambientes e contextos culturais;
- Desenvolver métodos de avaliação que permitam comparar diversas tecnologias de detecção de defeitos em contextos industriais e acadêmicos;
- Expandir a base tecnológica de análise de software criando novos pacotes de laboratório que possam ser reutilizados e adaptados por outros pesquisadores, baseando-se em seus próprios ambientes e necessidades culturais.

O sucesso no estabelecimento e evolução de processos de software requer mais do que a consideração de aspectos técnicos. Por exemplo, as características da organização, do pessoal técnico e do domínio de aplicação podem afetar a qualidade e a produtividade do processo e, conseqüentemente, do produto. Além de conhecer os métodos e processos de Engenharia de Software, é fundamental saber como avaliá-los para propiciar uma base sólida para a evolução do processo. A disponibilidade de uma base de dados histórica abrangendo as diversas perspectivas e fatores que afetam o desenvolvimento de software é um elemento indispensável nesse cenário. Estudos experimentais fornecem subsídios para a avaliação e constituição de tal base. No âmbito do Projeto *Readers*, duas linhas de investigação vêm sendo abordadas, tendo sido conduzidos experimentos específicos para cada linha:

- Linha de Pesquisa I: Consiste em aplicar e comparar diferentes técnicas de validação na atividade de detecção de defeitos utilizando como artefato o código-fonte. As seguintes técnicas têm sido utilizadas: Leitura de código; Teste Funcional; Teste Estrutural – fluxo de controle e fluxo de dados – e Teste de Mutação.
- Linha de Pesquisa II: Consiste em aplicar e comparar as técnicas de leitura *Perspective-Based Reading* (PBR) e *Checklist* visando a detectar defeitos em especificações. Utiliza como base o experimento de [Basili et al. \(1996a\)](#), descrito na Seção 2.4, sem usar os documentos de domínio específico e adotando *Checklist* como técnica de comparação. O projeto experimental adaptado é apresentado na Figura 2.15.

	Grupo 1			Grupo 2			
	<i>Designer</i>	<i>Tester</i>	<i>User</i>	<i>Designer</i>	<i>Tester</i>	<i>User</i>	
Técnica Checklist	Treinamento sobre a técnica <i>Checklist</i>						Primeiro Dia
	ATM			PG			
Técnica PBR	Treinamento sobre a técnica PBR						Segundo Dia
	PG			ATM			

Figura 2.15: Projeto de replicações da Linha de Pesquisa II do Projeto *Readers*

Nos experimentos na Linha de Pesquisa II (chamados Experimento II), o pacote de laboratório é constituído de:

- Procedimentos para o Replicador: descreve os passos a serem executados pelo responsável pelo experimento durante a replicação, estipulando tempos para as atividades;

- Formulários Iniciais: consiste dos formulários de consentimento, da lista de participantes e de um questionário de conhecimento;
- Material de Treinamento: consiste do material teórico e prático referente às técnicas de leituras *Perspective-based Reading* (PBR) e *Checklist*;
- Material de Instrução: consiste dos passos a serem seguidos pelo participante para a aplicação do experimento e dos formulários de coleta de dados;
- Especificações: documento de Requisitos;
- Formulário de Avaliação;
- Entrevista.

Tabela 2.2: Sumário do Questionário de Entrevista da Replicação 1 (R1) do Experimento II

Participante	Inglês	Experiência Em				Documento de Requisitos	
		Gerência	Desenvolvimento	Teste	Análise	Usando	Escrevendo
S01	Médio	0	60	36	24	24	24
S02	Médio	0	24	1	4	4	4
S03	Médio	0	24	1	2	3	3
S04	Avançado	0	24	2	0	3	2
S05	Médio	0	24	1	0	6	6
S06	Médio	0	24	5	5	5	5
S07	Avançado	0	24	4	4	4	0
S08	Avançado	2	0	2	0	1	1
S09	Especialista	0	4	1	2	3	3
S10	Avançado	0	36	3	2	3	1
S11	Avançado	0	24	1	6	6	6
S12	Avançado	12	12	1	6	6	3
S13	Avançado	0	4	1	1	4	1
S14	Médio	0	3	1	2	3	1
S15	Avançado	0	0	0	0	0	0
S16	Avançado	0	4	0	4	12	12
S17	Médio	0	4	0	4	4	6
S18	Avançado	0	0	0	0	6	6

As diversas replicações do experimento original de [Basili et al. \(1996c\)](#) resultaram em aprendizado e experiência adquiridos, que motivaram alterações nos pacotes de laboratório utilizados inicialmente. Até o momento, no ICMC-USP, foram realizadas quatro replicações do experimento. Os dados coletados são sumarizados em planilhas, como ilustrado nas Tabelas 2.2 e 2.3, que mostram atributos coletados na entrevista dos participantes e os defeitos reportados, respectivamente.

O Pacote de Laboratório evoluiu ao longo das múltiplas replicações: o documento PG, inicialmente preparado com 27 defeitos introduzidos, foi utilizado com 32 defeitos classificados quanto ao tipo; o formulário para delinear o perfil dos participantes também evoluiu, passando a coletar mais detalhes; sua estrutura também mudou – por exemplo, a experiência prévia na atividade de teste, inicialmente coletada em tempo (meses), passou a ser coletada separadamente para *Experiência em Teste*, *Experiência em Particionamento de Equivalência*, *Experiência em teste baseado em Requisitos/Use-Case*, e em categorias que mapeiam diferentes situações: *Nenhuma*, *Estudado em Classe ou em Livro*, *Praticado em Projeto Escolar*, *Em um projeto na Indústria*, ou *Em Múltiplos projetos na Indústria*.

Para a análise dos dados têm sido utilizadas as mesmas técnicas apresentadas na Seção 2.4.2. Pode-se observar na estrutura da Tabela 2.3 a organização dos grupos: a linha inferior da tabela mostra os quatro grupos e os sujeitos alocados a cada um. Pode-se observar também a “evolução” do pacote, já que o documento PG, inicialmente preparado com 27 defeitos (ver Tabela 2.1), é aqui apresentado com 32 defeitos, juntamente com as respectivas classificações quanto ao tipo. Os tipos de defeitos são classificados de acordo com a taxonomia: informação ambígua (A); informação inconsistente (II); fato incorreto (IF); informação estranha (E); outros defeitos (MD); e omissão (O). O número apresentado em cada célula da tabela denota a quantidade de defeitos reportados pelo participante, associados ao defeito existente no documento.

Quatro métricas foram usadas para avaliar quantitativamente os resultados obtidos:

- Defeitos Encontrados: o número de defeitos encontrados usando uma técnica específica;
- Ocorrência de Defeitos: quantas vezes cada defeito foi observado pelos participantes;
- Efetividade: percentual de defeitos encontrados por participante em cada documento. Os percentuais são calculados pela expressão $(x * 100)/y$, sendo x o número de defeitos observados por participante e y o número de defeitos presentes no documento;
- Eficiência: número de defeitos encontrados por participante por hora. É calculada usando a razão $x/(y/60)$, sendo x o número de defeitos encontrados por participante e y o tempo despendido na revisão do documento. A divisão por 60 ajusta a unidade do tempo coletado para minutos.

Tabela 2.3: Sumário de Defeitos reportados na Replicação 1 (R1) do Experimento II para o documento PG

Defeitos		Identificação dos Participantes																		
Tipo	Id	9	10	12	7	14	18	2	5	11	1	3	4	6	8	13	15	16	17	
II	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	
IF	2	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	
A	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
II	4	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	
O	5	0	0	0	1	0	1	1	1	1	1	3	0	0	1	0	1	0	1	
O	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
II	7	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
A	8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
O	9	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	
IF	10	1	0	1	1	1	0	1	0	1	0	1	0	1	0	0	0	0	0	
II	11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	
A	12	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
II	13	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	
II	14	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	
O	15	0	1	1	2	1	1	1	1	0	0	0	0	0	0	0	0	0	1	
O	16	0	1	1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	
II	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
II	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
II	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
O	20	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	
II	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
E	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
MD	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
O	24	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	
MD	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
O	26	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
O	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
O	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
O	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MD	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
O	31	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
A	32	1	2	1	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	
		PBR – D			PBR – T			PBR – U			Checklist									
		Técnicas Aplicadas																		

Uma massa considerável de dados é coletada em cada replicação, cujos registros informam a técnica aplicada, condições de aplicação, dados dos testadores, recursos utilizados e resultados obtidos (quanto à eficiência e à eficácia). Os documentos a serem avaliados, quer sejam código fonte ou especificação, também são disponibilizados e, conseqüentemente, manuseados pelos testadores.

2.6 Considerações Finais

Neste capítulo foi apresentado o foco deste projeto de doutorado – Engenharia de Software Experimental – tendo sido destacados diversos pontos em aberto que podem ser apoiados por

estratégias alternativas para a análise de dados. Pode-se mencionar especificamente: apoio à exploração não direcionada de dados coletados nos múltiplos experimentos; meta-análise dos dados de diversas replicações; formulação de listas de variáveis e hipóteses; e outras atividades para a melhoria da experimentação.

Como parte de um processo sistemático guiado pelo Pacote de Laboratório, a exploração dos dados coletados em experimentos isolados, apesar de ser não direcionada, deve ser incluída no Pacote de Laboratório e aplicada em um processo sistemático. Neste caso, a busca não direcionada deve ser guiada por um processo que dê diretrizes para estimular a busca por novas descobertas. A sistematização é essencialmente importante para a meta-análise, pois deve considerar desde a combinação dos dados oriundos de diferentes estudos experimentais, até o mapeamento visual, dando suporte à análise exploratória. A abordagem escolhida para a exploração de dados – no caso, a análise visual – deve considerar os requisitos da Engenharia de Software Experimental. Entretanto, para definir a abordagem para análise, faz-se necessário considerar conceitos e técnicas relacionados à Visualização de Informação, que são apresentados no próximo capítulo.

Visualização de Informação

Cada replicação de um experimento gera um conjunto de dados que, até então, tem sido analisado por métodos estatísticos. Com as replicações, um corpo de conhecimento vem sendo formado, caracterizando um contexto ideal para aplicar métodos alternativos de análise, como a análise apoiada por visualização. Além disso, há o interesse em buscar o conhecimento adicional embutido nos dados coletados. A visualização como alternativa para a exploração dos dados é uma proposta que merece atenção, apoiando diferentes abordagens de análise de dados e descoberta de conhecimento. O objetivo deste capítulo é apresentar uma visão geral da área de Visualização de Informação, bem como um modelo do processo de visualização e a terminologia utilizada. São apresentadas também algumas técnicas clássicas de Visualização de Informação. O capítulo encontra-se organizado como segue:

- Na Seção 3.1 é apresentada uma definição de Visualização de Informação e um modelo do processo de geração de representações visuais;
- Na Seção 3.2 uma breve introdução à Mineração de Dados é apresentada, com o objetivo de contextualizar a introdução de Visualização no processo de Mineração de Dados propriamente dito; e
- Na Seção 3.3 é apresentado um conjunto de técnicas de visualização capazes de apoiar a exploração de dados, suas classes e características;

-
- As Técnicas de Interação e Distorção, essenciais à visualização exploratória, são brevemente apresentadas na Seção 3.4;
 - Na Seção 3.5 é apresentada a estratégia de múltiplas visualizações coordenadas;
 - Na Seção 3.6 ferramentas para Visualização de Informação aplicadas à Exploração Visual dos dados no âmbito deste projeto são apresentadas.

3.1 Conceitos Básicos e o Modelo de Visualização

Artefatos visuais para a representação de informação vêm sendo utilizados há muito tempo em áreas em que sua aplicação é natural e necessária, como em cartografia, por exemplo. Entretanto, o computador e as facilidades por ele oferecidas permitem a criação de tais artefatos em ambientes virtuais e interativos. Dessa forma, *visualização* pode ser definida como uso interativo de representações visuais de dados para amplificar a cognição (Card et al., 1999), sendo que, nesse contexto, entende-se por cognição a aquisição ou uso de conhecimento por uma pessoa.

Informações não associadas a fenômenos físicos, tais como dados financeiros, coleções de documentos e informações organizacionais e abstratas, podem beneficiar-se de metáforas visuais, ainda que tais tipos de dados não tenham um mapeamento espacial direto. Além do problema de como representar propriedades visíveis de objetos de interesse há o problema de mapear abstrações não-espaciais em formas visuais efetivas. A Visualização de Informação aborda, então, o uso interativo de representações visuais de dados **abstratos** (Card et al., 1999).

Card et al. (1999) apresentam um modelo do processo de mapeamento visual mostrado na Figura 3.1. O fluxo indica uma série de transformações efetuadas sobre um conjunto de dados, sendo que cada seta pode indicar múltiplas transformações encadeadas, e as setas para as próprias atividades indicam ajustes das transformações. Nesse processo, a Transformação de Dados indica a organização de dados brutos em tabelas, juntamente com a descrição das relações e metadados, devendo-se tratar os dados quanto aos valores incorretos e aos valores ausentes. A etapa de Mapeamento Visual transforma os dados das tabelas em modelos visuais que combinam substratos espaciais, marcas e propriedades gráficas, ou seja, atributos dos dados são mapeados em posição, cor, transparência, dentre outras propriedades. Por último,

as Transformações Visuais criam as visões das estruturas visuais pela especificação de parâmetros gráficos como posição, escala e recorte (*clipping*). A interação do usuário controla os parâmetros dessas transformações.

Figura 3.1: Modelo para Visualização de Dados (Card et al., 1999).

A organização em tabela implica em elementos de dados com a mesma estrutura e na ausência de relações hierárquicas nos dados. O número de atributos de cada elemento do conjunto define a dimensionalidade do conjunto, sendo que conjuntos cujos elementos possuem um, dois ou três atributos são ditos unidimensionais, bidimensionais e tridimensionais, respectivamente. Porém, em Visualização de Informação o foco é, em geral, na representação de conjuntos com dimensionalidade superior a três, ditos multidimensionais.

Dados multidimensionais podem ser definidos como um conjunto de elementos E , sendo que o i -ésimo elemento e_i consiste em um vetor com n atributos $(x_{i1}, x_{i2}, \dots, x_{in})$. Cada atributo (dimensão) pode ser independente ou interdependente com um ou mais atributos. Os atributos podem ser de natureza discreta ou contínua, ou representar valores simbólicos (nominais). Além disso, podem também ter uma escala associada, definida de acordo com a existência ou não de um relacionamento, de uma distância métrica (intervalo), e de um zero absoluto (origem). Os dados multidimensionais podem ser organizados em *flat-files*, nos quais cada linha representa um elemento, ou registro, cujos atributos são, geralmente, separados por um símbolo especial (como vírgula, ponto e vírgula, ou outro qualquer). Relações implícitas entre os registros de um *flat-file* podem ser representadas em estruturas auxiliares, influenciando a estratégia utilizada para a visualização: um caso em que a relação entre os registros reflete diretamente na

representação gráfica é a hierarquia, naturalmente representada em uma árvore, por exemplo.

Quando visualizado, cada atributo dos dados multidimensionais pode ser representado por alguma entidade gráfica, sendo que o tipo (discreto, contínuo, nominal) ou a escala podem ser mudados para facilitar a exposição. Em tais situações, cuidados devem ser tomados para que múltiplos atributos, com características diferentes, não sejam apresentados com mesma propriedade gráfica percebida (tipo ou escala), podendo induzir falsas interpretações. Além disso, o mapeamento deve considerar a técnica e o tipo de dados do atributo, pois certas propriedades gráficas não são adequadas para mapear certos tipos de dados (Bertin, 1981, Card e MacKinlay, 1997, Card et al., 1999, Ware, 2000), como *tamanho* em projeções $2^{1/2}D$.

O modelo de processo proposto por Card et al. (1999) é genérico e abstrato. Por não especificar, por exemplo, as atividades a serem efetuadas em cada etapa e tampouco considerar características específicas dos dados, sua instanciação no contexto de experimentação permite que uma mesma tarefa de análise seja conduzida de diferentes maneiras. Essa característica dificulta, ou até mesmo impossibilita, a aplicação consistente e uniforme do processo de visualização pelos envolvidos no processo de experimentação. Na replicação de experimentos há uma dificuldade muito grande em captar o conhecimento tácito existente quando uma definição é vaga, o que leva à necessidade de explicitar as atividades em cada etapa no Pacote de Laboratório.

Na Figura 3.1 observa-se que o processo é centrado no usuário, que realiza tarefas conforme seus objetivos. Cabe ao usuário determinar as tarefas, que podem ser subdivididas segundo dois aspectos: tarefas para gerar visualizações adequadas, o que requer um especialista em visualização; e tarefas para usar as visualizações com o objetivo de aprender sobre os dados, já que o modelo pode ser instanciado em diferentes contextos, o que requer um especialista no domínio dos. Observa-se que há dois papéis para o usuário representado no modelo e que tais tarefas não são claramente definidas (e nem poderiam, já que depende do usuário definir seus objetivos). Independente do papel, tal conjunto de tarefas deve gerar uma representação visual capaz de explicitar conhecimento, instanciado para revelar informações úteis, ajudando o analista a perceber propriedades e relações, aumentando a sua cognição. O conhecimento a ser explicitado é delimitado, pois tem como foco um objetivo específico, como a investigação de uma hipótese ou a identificação e prospecção de relações não antecipadas.

Quanto ao uso, a visualização pode ser aplicada nos seguintes contextos que apresentam características e requisitos distintos: Apresentação de Informação, Apoio à Decisão, Análise Confirmatória e Análise Exploratória. Para a *Apresentação de Informação*, deve-se escolher uma forma de exibição dos dados que evidencie fatos já detectados. Já o *Apoio à Decisão* permite a exploração de modelos de dados e do espaço de soluções resultante de um processo de busca. A *Análise Confirmatória* tem como meta a confirmação de uma hipótese previamente estabelecida, direcionando uma busca que pode corroborar ou não tal hipótese. Diferentemente, a *Análise Exploratória* não parte de hipóteses prévias, mas sim dos dados e, em uma busca iterativa e geralmente não direcionada, tenta identificar tendências, padrões e correlações. No escopo deste trabalho, a visualização pode ser aplicada para auxiliar, em especial, três atividades: Apresentação de Informação, Análise Confirmatória e Análise Exploratória. A primeira delas tem potencial uso na sumarização de resultados por categorias de interesse. Por serem bem conhecidas, técnicas de visualização para apresentação – como gráficos em Barra, Pizza e Linha – não são aqui explanadas. As outras atividades, tanto de confirmação quanto de exploração, podem ser beneficiadas pelo uso de técnicas interativas de visualização multidimensional, em conjunto com técnicas de Apresentação.

3.2 Visualização em Mineração de Dados

Esta seção apresenta a Mineração de Dados tanto como uma área que pode se apoiar na Visualização de Informação em atividades inerentes ao processo, quanto como uma área que pode adotar a Visualização de Informação como elemento chave de seu processo, dando origem à Mineração Visual de Dados. Assim, uma visão geral de Mineração de Dados é apresentada, juntamente com o uso de Visualização de Informação em ambos os casos.

3.2.1 Mineração de Dados

A Mineração de Dados pode ser definida como o processo de pesquisa e análise de dados com o objetivo de encontrar informações implícitas. A participação do ser humano no processo de mineração é de extrema importância. Por ser dotado de um sistema de percepção capaz de tratar diferentes tipos de estímulos de modo flexível, o ser humano é capaz de analisar eventos complexos com rapidez, e de reconhecer propriedades não usuais desconsiderando as bem conhecidas. Além disso, é capaz obter conclusões a partir de descrições vagas e informações imprecisas,

utilizando seu conhecimento prévio acerca de determinado assunto (*background*) (Keim, 1996).

A descoberta de conhecimento em bases de dados (KDD – *Knowledge Discovery in Databases*), segundo Fayyad (Fayyad et al., 1996a,c), tem por objetivo identificar padrões e modelos válidos, novos e potencialmente úteis às organizações. De acordo com Fayyad, Piatetsky-Shapiro e Smyth (Fayyad et al., 1996a,b), o processo de KDD é interativo e iterativo, e se refere ao processo global de descoberta de conhecimento útil a partir de dados, sendo que a Mineração de Dados é apenas uma dessas etapas. Rezende et al. (2003) agrupa o processo em um ciclo com três grandes etapas: Pré-Processamento, Extração de Padrões e Pós-Processamento, como ilustrado na Figura 3.2. Adicionalmente, os autores identificam uma fase anterior ao processo de Mineração de Dados, que trata da identificação do problema a ser abordado, e uma fase posterior, que se refere à utilização do conhecimento adquirido. Tais etapas, apresentadas esquematicamente na Figura 3.2, são descritas a seguir:

Figura 3.2: Processo de Mineração de Dados (Rezende et al., 2003)

- **Identificação do Problema:** Para identificar os objetivos da mineração de dados, é necessário analisar o problema. Pessoas com conhecimento do domínio são fundamentais nesta etapa;
- **Pré-Processamento:** os dados podem conter ruído, estar incompletos ou em formato e tamanho (quantidade de instâncias) inadequado, sendo necessário tratá-los antes do pro-

cessamento. Quando os dados provêm de fontes diferentes ou estão em formatos diferentes, é necessário unificá-los para que sejam interpretados corretamente. A limpeza da base de dados também é importante, pois dados incompletos ou incorretos podem induzir ao erro. A escolha de um subconjunto representativo também está envolvida no pré-processamento, ou seja, a diminuição do número de instâncias e a eliminação de atributos pouco relevantes devem ser feitos nesta etapa;

- **Extração de Padrões:** é a principal etapa do processo de descoberta de conhecimento, pois é nela que acontece a mineração propriamente dita. Para isso é necessário identificar o tipo de modelo desejado – descritivo ou preditivo –, escolher a tarefa apropriada e o algoritmo a ser aplicado;
- **Pós-Processamento:** descobertos os padrões é necessário analisá-los, e algumas vezes interpretá-los para utilizá-los na tomada de decisão pelo usuário final. A escolha/filtragem dos padrões mais relevantes também faz parte desta etapa, assim como a avaliação e julgamento do conhecimento obtido;
- **Utilização do Conhecimento:** a última etapa do processo consiste em utilizar o conhecimento obtido em relatórios, sistemas inteligentes ou em atividades de tomada de decisão.

Para [Keim et al. \(1994\)](#), duas características importantes de ferramentas de Mineração de Dados são a interatividade e o desempenho (estabilidade). Eles consideram que o ser humano deve ter mais controle sobre os processos de Mineração, com computadores, que são adequados para processar pesquisas e cálculos numéricos, auxiliando a inteligência, a criatividade e as habilidades de percepção. Uma estratégia para “inserir” o ser humano no processo é o uso de Visualização ([Wong, 1999](#), [Keim, 2002](#)).

3.2.2 Mineração Visual de Dados

[Keim \(2002\)](#) aponta que para a mineração de dados ser efetiva, é importante incluir o ser humano no processo de exploração e combinar a sua flexibilidade, criatividade, percepção visual e conhecimento geral com a alta capacidade de armazenamento e processamento dos computadores. [Ankerst \(2000\)](#) define Mineração Visual de Dados (VDM) como um passo no processo de extração de conhecimento (KDD), não necessariamente automático, que utiliza a visualização como um canal de comunicação entre computador e usuário para apoiar a identificação de

padrões novos e interpretáveis. Posicionando a Visualização em diferentes fases do processo (Mineração de Dados e Avaliação), como apresentado na Figura 3.3, Ankerst (2000) identifica três categorias de VDM:

- Visualização dos dados: nessa categoria o usuário tem total controle sobre a busca no espaço de busca, interagindo com a representação dos dados brutos. O problema dessa abordagem é que a exibição de todos os itens de dados geralmente sobrecarrega visualmente o resultado (3.3(a));
- Visualização dos resultados de uma mineração: oferece ao usuário um apoio à interpretação dos modelos extraídos. Nessa abordagem é possível obter uma descrição condensada ou um modelo do conjunto de dados. Vale ressaltar que a determinação da técnica de mineração mais adequada, sem um conhecimento prévio sobre os dados, não é uma tarefa trivial, e que novas tentativas com diferentes algoritmos de mineração podem ser necessárias (3.3(b));
- Visualização de um processo de mineração: permite direcionar a busca apresentando resultados intermediários, fornecendo conhecimento sobre o domínio, por exemplo, para instanciar um arcabouço genérico desenvolvido para diferentes aplicações, com a intervenção do usuário. Esta abordagem dá ao usuário uma maior compreensão do processo de mineração sendo executado, do seu espaço de busca e das características dos dados manipulados a cada etapa, permitindo um maior controle sobre o processo como um todo 3.3(c).

O emprego de Visualização no processo de Mineração de Dados introduz algumas possibilidades, tais como: percepção de propriedades não antecipadas, essencial para o descobrimento de novos padrões; percepção simultânea das características dos dados em grande e pequena escala; apoio a processos de formação de hipóteses; apoio a tarefas de pré-processamento dos dados, como detecção de problemas, limpeza, seleção, dentre outras.

Para atingir seu objetivo, um processo de Mineração de Dados precisa aplicar algoritmos de busca de padrões (*pattern-matching*), por exemplo. A presença de padrões em uma coleção de dados pode ser facilmente observada quando estes são organizados e apresentados de

Figura 3.3: Categorias de VDM segundo Ankerst (Ankerst, 2000)

modo adequado utilizando uma representação visual. Em modo textual, como em tabelas, por exemplo, nem sempre fica nítida a relação e a semelhança (por padrão) entre dados.

Wong (1999) identifica duas categorias de uso de Visualização no processo de Mineração: Fracamente e Fortemente Acoplada. Na primeira delas a Visualização é “intercalada” com estratégias analíticas de mineração, sendo utilizada para apoiar atividades do processo de KDD, como pré-processamento e interpretação de resultados, por exemplo. Na segunda, a Visualização é “integrada” a estratégias analíticas de mineração, dando ao usuário maior controle e entendimento sobre o processo analítico, apoiando a tomada de decisões necessárias e permitindo a criação de representações visuais do espaço de busca, situação identificada na Figura 3.3(c). A visualização integrada à mineração (com interferência humana a partir da interpretação de resultados visuais), permite maior controle sobre a execução do algoritmo, já que é possível usar o conhecimento do domínio direcionar o processo.

3.3 Técnicas para Visualização Exploratória

Técnicas de visualização de informação multidimensional podem oferecer um apoio importante a processos de extração de informação, pois exploram a habilidade humana de detectar padrões, mesmo em cenários complexos. Tais técnicas permitem a visualização (em geral, em uma exibição bidimensional) de conjuntos de dados muito grandes ou de alta dimensionalidade. A

entrada para as técnicas de visualização de informação orientadas a dados multidimensionais aparece, em geral, na forma de tabelas de dados, como discutido na Seção 3.1. Muitas técnicas foram propostas para explorar visualmente tabelas de dados (Keim, 1996, Wong e Bergeron, 1997, Hoffman, 1999, Card et al., 1999, Eick, 2000, Keim, 2000, de Oliveira e Levkowitz, 2003, Chen, 2004).

Keim e Kriegel (1996) apresentam uma visão geral de técnicas multidimensionais, dividindo-as nas seguintes categorias: **orientadas a pixel**, **geométricas**, **iconográficas**, e **hierárquicas**. Posteriormente, Keim (2002) estendeu a classificação destacando que uma taxonomia deve considerar também os tipos de dado e as técnicas de interação, que podem ser combinados com a técnica de maneira ortogonal. As categorias identificadas por Keim são apresentadas a seguir.

3.3.1 Técnicas Orientadas a Pixel

Na categoria das **técnicas orientadas a pixel**, atributos de dados são mapeados em *pixels*, os quais são coloridos segundo o valor representado. O conjunto de valores de cada atributo é apresentado em uma janela individual de forma que, para um conjunto de dados com m atributos, a tela é particionada em m janelas (Figura 3.4).

Um aspecto relevante é o arranjo espacial dos *pixels* (ou, mais precisamente, dos elementos de dados) na tela, que deve apresentar boas propriedades de agrupamento de elementos semelhantes, e manter algum significado semântico. Na Figura 3.5 é apresentada uma organização em espiral, embora outras organizações sejam possíveis, como a apresentada na Figura 3.6, na qual i e j representam dois atributos quaisquer cujos valores separam a área de apresentação em quadrantes. Correlação e dependência funcional entre atributos podem ser detectadas pela análise de regiões correspondentes em diferentes janelas.

Keim e colaboradores têm tido um papel importante no desenvolvimento e aplicação de técnicas nessa categoria (Keim et al., 1994, Keim, 1997, 1996, Ankerst et al., 1996, Keim, 2000, 2002, Sips et al., 2006). Tais técnicas têm uma dependência direta em relação à resolução da tela: quanto maior a dimensionalidade do conjunto de dados, menor o número de elementos que podem ser exibidos simultaneamente. Na Figura 3.7 é apresentada uma ilustração da técnica.

As técnicas orientadas a *pixel* podem ser classificadas como independentes ou dependentes da consulta. Em técnicas independentes da consulta a organização dos *pixels* em sub-janelas

Figura 3.4: Distribuição dos atributos visualizados (Keim, 1997).

Figura 3.5: Arranjo em forma de espiral (Keim, 1997).

Figura 3.6: Organização em espiral por quadrante (Keim, 1997).

é fixa, independentemente dos valores dos atributos, e o mapeamento de uma cor ao *pixel* é baseado no valor do atributo associado. Em técnicas dependentes da consulta, o posicionamento de um elemento de interesse é tomado como base e distâncias dos valores do atributo exibido ao atributo do elemento de interesse são calculadas. Neste caso, o mapeamento de uma cor para um *pixel* é baseado na distância computada para cada atributo, e *pixels* em cada sub-janela são organizados de acordo com suas distâncias ao elemento de consulta.

Uma vantagem dessas técnicas é maximizar o espaço útil de tela, já que cada atributo é mapeado em um único *pixel* - conseqüentemente, é possível representar conjuntos de dados grandes, em comparação com outras técnicas multidimensionais. Entretanto, é possível des-

Figura 3.7: Visualização por *pixels* de um conjunto de dados aleatório em que cada item tem 9 atributos, usando dois arranjos distintos para a disposição dos itens (Keim, 1997).

tacar algumas desvantagens, como dificuldade de interpretação, pouco significado semântico e necessidade de treinamento. Aliás, o treinamento é necessário não só para interpretar o posicionamento – que depende do arranjo –, mas também o mapeamento de cor, que deve ser cuidadosamente escolhido. Um bom arranjo e um bom mapeamento de cor facilitam a descoberta de agrupamentos e correlações entre os atributos. Para a visualização geo-espacial o arranjo não é um problema, mas a escalabilidade sim, e pode ser necessário associar múltiplos dados para um mesmo pixel, dependendo da resolução do dispositivo de exibição (Sips et al., 2006). Vale ressaltar que a escalabilidade é um problema geral, para diversas categorias de técnicas de visualização. Há também a limitação de um *pixel* representar um único atributo em um mapeamento direto, o que não existe em técnicas de projeção, expostas a seguir.

3.3.2 Técnicas de Projeção Geométrica

Técnicas de projeção geométrica buscam encontrar projeções “interessantes” dos dados multidimensionais no espaço bidimensional. Os tradicionais Gráficos de Dispersão (*Scatter plots*) (Cleveland, 1993) usados em Estatística são um exemplo típico, em que dois atributos dos dados são projetados ao longo dos eixos x e y de um sistema de coordenadas cartesianas. Matrizes de Gráficos de Dispersão oferecem uma extensão para exibir dados multidimensionais. A estratégia consiste em exibir simultaneamente múltiplas projeções bidimensionais dos diferentes pares de atributos. Na Figura 3.8 é apresentado um gráfico de dispersão gerado com dados obtidos

em uma replicação realizada no âmbito do Projeto *Readers*.

Figura 3.8: Matriz de Gráficos de Dispersão gerada com o conjunto de dados coletados em replicações de experimentos realizadas no âmbito do Projeto *Readers*

Considerando que existem inúmeras maneiras de projetar dados multidimensionais em uma tela bidimensional, foi proposto um algoritmo denominado *Grand Tour* (Asimov, 1985, Wegman e Luo, 1996), para encontrar tais projeções automaticamente. Essa abordagem utiliza animação para exibir a seqüência de todas as projeções bidimensionais, em uma certa ordem. Essa solução é muito lenta e computacionalmente cara no caso de dados de alta dimensão. Uma alternativa é *Projection Pursuit* (Crawford e Fall, 1990), que tenta identificar e exibir apenas as projeções potencialmente mais interessantes, identificadas por um índice que caracteriza a projeção, sendo que tal índice é submetido a uma busca heurística que usa densidade normalizada padrão (Nason, 1996).

Outra técnica bastante conhecida é denominada Coordenadas Paralelas (Inselberg e Dimsdale, 1990, Inselberg, 1997), na qual um espaço de dimensão k é mapeado para um espaço bidimensional usando k eixos equidistantes e paralelos a um dos eixos principais. Cada eixo representa um atributo cujo intervalo de valores é mapeado linearmente sobre o eixo. Cada registro é exibido como uma poli-linha que intercepta cada eixo no ponto correspondente ao

valor do atributo associado, normalizado em cada eixo. A técnica permite identificar características como distribuição dos dados e dependências funcionais (Wegman e Luo, 1996, Inselberg, 1997), bem como densidades marginais, estruturas não lineares e agrupamentos, entre outros. Na Figura 3.9 é apresentado o instantâneo de uma representação por Coordenadas Paralelas dos mesmos atributos apresentados na Figura 3.8.

Figura 3.9: Visualização por Coordenadas Paralelas de sete atributos do Conjunto de Dados coletados em replicações de experimentos realizadas no âmbito do Projeto *Readers*

Técnicas similares têm sido propostas, como o Gráfico Estrela (*star graph*), no qual n eixos são arranjados em uma disposição radial, e elementos (que podem ser interpretados como pontos do espaço n -dimensional) são representados por curvas que interceptam cada eixo na posição correspondente aos valores dos atributos associados (Sobol e Klein, 1989, Parsaye e Chignell, 1993) e *Radial Coordinate Visualization, RadViz* (Hoffman, 1999) e *Viz3D* (Artero e Oliveira, 2004), que adotam uma disposição radial de eixos para posicionar os elementos.

Para técnicas desta categoria, a ordem de exibição dos atributos (seja em eixos ou janelas) afeta a qualidade da visualização resultante – algumas escolhas podem evidenciar padrões, outras não. Artero e Oliveira (2004) integraram uma abordagem analítica baseada em similaridade

de atributos para determinar uma “boa” sequência de apresentação. Para grandes volumes de dados, as técnicas de projeção geométrica também apresentam baixa escalabilidade e resultam em apresentações cuja interpretação é prejudicada (Keim e Kriegel, 1996, Lopez et al., 2001). Uma possível abordagem para reduzir o problema considera a densidade no espaço de dados projetado em Coordenadas Paralelas para identificar e realçar visualmente a presença de grupos (identificados por regiões de alta densidade nos dados) (Artero et al., 2004). Para os conjuntos de dados tratados neste trabalho, o uso dessas abordagens não é fundamental, já que o baixo número de registros permite contornar as dificuldades por meio de interações do usuário, por exemplo, destacando por cor uma seleção de elementos de interesse. Vale ressaltar que técnicas de projeção geométrica, como Coordenadas Paralelas, resultam em apresentações visuais capazes de exibir algumas dezenas de atributos, o que a torna adequada para as análises exploratórias realizadas no âmbito deste projeto.

3.3.3 Técnicas Iconográficas

Nas **técnicas iconográficas**, um ícone é utilizado para mapear cada elemento multidimensional e, para isso, atributos dos dados são usados como parâmetros que definem propriedades visuais dos ícones. A apresentação de uma quantidade considerável de elementos gráficos – resulta em um efeito visual de textura criada a partir dos dados.

Diferentes ícones podem ser usados, como as ‘estrelas’ nos Ícones Estrela (*Star Glyphs*) (Ward, 1994), e a configuração de arestas no *stick figures* (Pickett e Grinstein, 1988). No primeiro, as “estrelas” são geradas variando-se o comprimento de segmentos de reta arranjados radialmente de acordo com os valores dos atributos a eles associados. Nos *stick figures*, as duas dimensões da tela são usadas para mapear dois atributos, com os demais sendo mapeados em ângulos e/ou comprimentos das arestas. A vantagem desse ícone é que, ao exibir um volume de dados, variações qualitativas podem ser detectadas facilmente, como pode ser observado na Figura 3.10¹, que apresenta dados sobre um censo: Sexo, Educação, Ocupação e Estado Civil definem o ícone (3.10(a)) e Renda e Idade mapeados para os eixos X e Y, respectivamente, (3.10(b)).

Nas técnicas iconográficas o número de atributos que pode ser apresentado é limitado pelos atributos visuais perceptíveis do ícone adotado e a quantidade de elementos que podem ser

¹Obtida de <http://ivpr.cs.uml.edu/index.html>

Figura 3.10: Exemplo de Visualização com *Stick Figure*

visualizados simultaneamente é bastante limitada. Essas técnicas são adequadas para a detecção de padrões, exceções (*outliers*) e variações bruscas nos dados, mas são menos efetivas para comparações, por exemplo. Outros problemas que dificultam a sua aplicação são: a dificuldade em elaborar um ícone significativo (um metáfora visual que revele padrões, por exemplo); a dificuldade em aumentar a dimensionalidade de um mapeamento, pois implica em modificar o ícone.

3.3.4 Técnicas Hierárquicas

Técnicas hierárquicas subdividem o espaço de dados em sub-espacos hierárquicos para exibição. A idéia é mapear dados não-hierárquicos k -dimensionais em sub-espacos bidimensionais, aninhados, mantendo a relação entre eles, sendo que o arranjo dos atributos é importante para a representação dos dados. Exemplos representativos dessa estratégia são *Worlds-within-Worlds* (Beshers e Feiner, 1993) e Aninhamento Dimensional (*Dimension Stacking*) (LeBlanc et al., 1990).

Na técnica *Worlds-within-Worlds* (Feiner e Beshers, 1990) as dimensões visuais são aninhadas com três atributos mostrados em cada nível, para gerar uma hierarquia interativa das exposições. Os três eixos mais externos são representados por três eixos ortogonais, como mostrado na Figura 3.11. É escolhido, então, um ponto que define uma posição no espaço e um novo

conjunto de eixos é posicionado. Isso continua até que todos os atributos estejam definidos. No nível mais interno o atributo final pode ser exibido como uma superfície 3.11. A saída é uma exposição tridimensional dos mundos virtuais dentro dos quais o usuário pode explorar seus dados. Essa técnica dinâmica de visualização dá um novo significado aos gráficos de manipulação direta. Os diferentes mapeamentos gerados pelas combinações dos atributos associados aos eixos dão origem a diferentes visualizações. O processo direto da manipulação é, de fato, uma consulta de recuperação dos dados. Assim, os usuários têm que saber o que estão procurando, já que nem toda a informação é visível na exposição inicial. Esse processo interativo tende a ser difícil e tedioso devido à grande quantidade de combinações possíveis de mapeamentos para os atributos (Wong e Bergeron, 1997).

Figura 3.11: Subdivisão do espaço de coordenadas(Wong e Bergeron, 1997)

Outro exemplo de técnica hierárquica é o *Treemap* (Shneiderman, 1992). Esta técnica, ao contrário da anterior, pressupõe que o espaço dos dados é hierárquico, e mapeia a hierarquia para a representação gráfica. Ela é, portanto, particularmente adequada para a visualização de estruturas de árvores, mapeando diretamente uma árvore pela subdivisão recursiva do espaço de representação correspondente, uma janela ou uma região planar – os nodos folha são mapeados em um espaço proporcional ao valor do atributo escolhido. Um mapa de cores pode representar outros atributos dos nodos folha, e rótulos podem ser exibidos. A representação de dados não hierárquicos é possível, desde que seja definida uma hierarquia para a exibição. Na Figura 3.12 é apresentado o mapeamento dos defeitos reportados em um experimento do Projeto *Readers* organizado hierarquicamente por documento, tipo de defeito e as “folhas” são os defeitos. O tamanho dos retângulos indica o número de vezes que o defeito foi reportado, o número apresentado internamente em cada retângulo identifica o defeito, e as cores indicam as replicações do experimento, conforme indicado na legenda.

Figura 3.12: Exemplo de *Treemap* apresentando o conjunto de defeitos reportados em replicações do Experimento II do Projeto Readers

Uma vantagem dessa categoria de técnicas é a representação de hierarquias presentes no conjunto de dados diretamente no espaço de apresentação. Entretanto, quando a relação hierárquica não é inerente aos, pode ser difícil definir uma organização hierárquica adequada. Com organização hierárquica inerente ou não, pode-se destacar outros dois problemas: a apresentação de vários níveis de hierarquia; e dificuldade de visualizar áreas que representam valores muito pequenos, ao utilizar a área segmentada para representar atributos cujos valores têm alta amplitude.

3.4 Técnicas de Interação e Distorção

Para a visualização exploratória, recursos de interação constituem ferramentas essenciais, permitindo ao analista selecionar diferentes aspectos do conjunto de dados, ou da representação visual, alterando dinamicamente a visualização. As técnicas de distorção permitem exibir regiões de interesse da visualização com maior detalhe, enquanto as demais permanecem inalteradas. As seguir técnicas de interação e distorção são apresentadas.

3.4.1 Filtragem Interativa

Técnicas de Filtragem possibilitam ao analista selecionar interativamente um subconjunto de interesse do conjunto de dados. A seleção pode ser feita diretamente nos dados, ou pela especificação de propriedades específicas. Os filtros são operações que podem ser aplicadas aos dados selecionados, ou seja, os dados submetidos à visualização.

Nesta categoria de técnica estão as consultas dinâmicas (*dynamic queries*) (Shneiderman, 1994), representadas visualmente por meio de controles deslizantes (*Sliders*) e botões de seleção. No primeiro caso, os controles deslizantes são constituídos, basicamente, por uma barra graduada, que indica os valores possíveis, e um ou dois cursores que o usuário utiliza para delimitar a faixa de valores de interesse, como ilustrado na Figura 3.13 (a) e (b), respectivamente. Já no segundo caso, a consulta pode ser feita por meio de botões e caixas de verificação (ver Figura 3.13 (c) e (d)), para habilitar e desabilitar valores de interesse. Em ambos os casos, Eick (1994) sugere a inclusão de algumas informações sobre os dados no próprio controle.

Figura 3.13: Controles deslizantes com um (a) e dois cursores (b), e botões de rádio (c) e caixas de verificação (d)

3.4.2 Zoom Interativo

O *zoom* interativo permite que o usuário tenha uma visão geral dos dados, juntamente com visualizações detalhadas de áreas de interesse. A técnica consiste em apresentar o conjunto de dados em diferentes níveis de detalhe: em uma resolução baixa os marcadores gráficos utilizados são constituídos, por exemplo, por um único pixel; em uma resolução intermediária por uma figura; e em uma resolução alta por figuras e/ou rótulos contendo a descrição do objeto.

TableLens (Rao e Card, 1994) é uma técnica que utiliza este recurso para permitir que muitos registros sejam apresentados na tela. Cada valor de atributo é representado por uma barra horizontal cujo comprimento é determinado pelo valor do atributo. O número de colunas é

dado pelo número de atributos, sendo que a técnica permite que o usuário ordene as colunas, facilitando a identificação de correlações entre os dados. Quando o usuário deseja obter mais informações sobre algum elemento específico, basta selecionar o marcador para ter os seus valores apresentados. Na Figura 3.14 é apresentado o *TableLens* aplicado a um conjunto de dados sobre jogadores de *Baseball* da liga norte-americana.

Figura 3.14: Instantâneo da ferramenta *TableLens* (Rao e Card, 1994).

3.4.3 Distorção Interativa

A distorção interativa considera que o conjunto de dados sob análise possui um sub-conjunto de interesse que precisa ser apresentado em detalhe. Dessa forma, a representação visual do sub-conjunto alvo é apresentada com variações no tamanho dos elementos gráficos. Essa técnica permite a visualização do sub-conjunto de interesse sem perder o relacionamento com o conjunto como um todo. Isso é particularmente útil quando existem relacionamentos entre os objetos, como ocorre em árvores ou redes.

Para obter tal efeito pode-se aplicar uma projeção perspectiva sobre uma esfera aos objetos gráficos, ou outras projeções, sobre cilindros, por exemplo (Carpendale et al., 1997, Leung e Apperley, 1994). Na Figura 3.15 é mostrado um exemplo de distorção: na região central os objetos são aumentados, e nas bordas mantém um tamanho que permite observar as relações

entre os elementos.

Figura 3.15: Exemplo de “máscara” de distorção (Carpendale et al., 1997)

3.4.4 Seleção e Ligação Interativas

Em visualização multidimensional a operação de seleção (*brushing*) se refere ao processo de selecionar um subconjunto de registros apontando diretamente os seus elementos gráficos, por exemplo, com o *mouse*. A maneira de selecionar os elementos depende da visualização. No caso dos Gráficos de Dispersão é comum selecionar um único elemento gráfico (ver Figura 3.16(a)) apontando-o diretamente, ou delimitar uma região para selecionar múltiplos marcadores (ver Figura 3.16 (b)). Já em Coordenadas Paralelas, o procedimento é similar para selecionar um único elemento (Figura 3.16 (c)), porém pode-se selecionar uma região sobre um eixo estabelecendo um intervalo de interesse como condição na seleção, ou seja, os registros cujo valor do atributo pertence ao intervalo demarcado são selecionados (Figura 3.16(d)). A seleção pode ser feita de forma cumulativa em ambos os casos, estabelecendo múltiplas seleções simultâneas.

A ligação (*linking*) permite que objetos selecionados em uma visualização sejam também destacados em outra relacionada. Tal recurso requer que diferentes representações visuais estejam integradas, o que é discutido a seguir.

3.5 Múltiplas Visualizações Coordenadas

A possibilidade de relacionar e combinar visualizações diferentes é um recurso importante (Keim, 2002). **Técnicas híbridas** integram múltiplas estratégias de visualização em uma ou várias janelas. Todas as técnicas apóiam-se em um extenso conjunto de estratégias de interação para apoiar processos dinâmicos de exploração. Múltiplas visualizações de um mesmo conjunto

Figura 3.16: Seleção de marcadores em: um gráfico Gráfico de Dispersão – (a) e (b) – e em uma representação por Coordenadas Paralelas – (c) e (d).

de dados, ou de conjuntos distintos relacionados, permitem observá-los sob várias perspectivas, bem como explorar os pontos fortes e minimizar o efeito de possíveis pontos fracos das técnicas envolvidas (Keim e Kriegel, 1996, Eick, 2000). Nesse caso, é interessante que as múltiplas visualizações estejam coordenadas durante as etapas exploratórias de forma que ações sobre uma delas sejam refletidas nas demais (North e Shneiderman, 2000). Nesse contexto, as interações ocorrem em duas modalidades: intravisualização e intervisualizações. Nesta última, as visualizações são ditas “fortemente acopladas”, com significado distinto do caso de integração entre técnicas de Visualização e algoritmos de mineração (apresentado na Seção 3.2).

O conceito de visualizações coordenadas é explorado no ambiente Snap- Together Visualization (STV), ou simplesmente Snap, uma solução que viabiliza a coordenação entre múltiplas ferramentas de visualização. Essa solução é composta por um modelo de coordenação, uma interface com o usuário e uma arquitetura de sistema. O ambiente implementado viabiliza, de forma genérica, o uso coordenado de múltiplas ferramentas de visualização, permitindo ao usuário criar cenários personalizados integrando ferramentas independentes. A meta é oferecer ao usuário flexibilidade para acesso a diferentes conjuntos de dados, para a escolha de visualizações adequadas aos dados, e para a configuração dinâmica de coordenações, de acordo com as tarefas a serem executadas. Ferramentas de visualização são integradas ao ambiente por uma API, e interagem por meio de mecanismos de comunicação entre processos (North e Shnei-

derman, 2000). Os dados visualizados estão armazenados em tabelas ou são provenientes de consultas a bases de dados relacionais.

O modelo conceitual do Snap para coordenação de visualizações é baseado no esquema de dados relacional. As visualizações disponibilizadas no STV utilizam como entrada tabelas de dados ou resultados de consultas a um banco de dados relacional, mapeando cada tabela/consulta para a representação visual escolhida pelo usuário. A coordenação entre duas visualizações também é configurada pelo usuário, com base nos relacionamentos entre as tabelas, os quais são definidos por chaves primária e estrangeira.

A estratégia de coordenação segue o modelo apresentado por North e Shneiderman, que formaliza esquema de múltiplas visualizações coordenadas com base nos seguintes elementos:

- A definição de visualizações individuais que exibem elementos de um conjunto de dados. Uma coordenação entre duas visualizações estabelece uma relação entre os elementos por elas exibidos;
- Uma linguagem para especificar a coordenação entre duas visualizações, com as seguintes sintaxe e semântica:
 - $((visA, açãoA, itensA)(visB, açãoB, itensB))$: Uma ação do tipo *açãoA* em uma visualização *visA* envolvendo os itens *itensA* provoca uma ação correspondente *açãoB* na visualização *visB* envolvendo os itens *itensB*. O caminho inverso de *visB* para *visA* também é válido, pois a coordenação tem propriedade comutativa;
- Uma taxonomia para os tipos de ações:
 - *Seleção*: O usuário seleciona elementos de interesse, que são destacados. Opcionalmente, para ressaltar os elementos selecionados, os demais podem ser apagados total ou parcialmente;
 - *Navegação*: O usuário navega em uma visualização para focar em elementos de interesse ou exibir esses itens. Exemplos de ações de navegação são: abrir arquivo, carregar dados, ampliar/reduzir, deslizar, particionar, rotacionar, e outras.

A interface do ambiente, apresentada na Figura 3.17, tem duas regiões principais: uma a extrema esquerda (relativamente pequena) na qual é representado o modelo da base de dados

e as técnicas que compõem a visualização, juntamente com o modelo de coordenação; e outra à direita na qual são apresentadas as visualizações propriamente ditas. Na Figura 3.17 é apresentado um esquema e o cenário resultante de sua aplicação: à esquerda, na área de apresentação do esquema (abaixo da aba *Database*), o ícone do gráfico de linha (exibindo o *Profile*) foi estendido para mostrar os atributos mapeados na representação (no caso, *English*, *Manager*, *Developer*, *Tester*, *Analyst*, *Using_Req*, *Writing_Req* e *Subject*, embora seja possível visualizar apenas os três primeiros), e a lista completa no eixo x do gráfico correspondente, logo à direita na área das visualizações; os ícones encontram-se conectados no esquema, indicando que uma ênfase (H-Highlight) na representação por Coordenadas Paralelas deve disparar uma Seleção (S-Selection) na representação por Gráfico de Dispersão (*Scatter plot*) e um carregamento (L-Load) na representação por Gráfico de Linha. Ainda no esquema pode-se observar no Gráfico de Dispersão os dados sobre os defeitos (defeitos mapeados no eixo x e número de vezes que o defeito foi reportado no eixo y) e na representação por Coordenadas Paralelas os registros de detecção de defeitos de cada participante. No exemplo, analisando o defeito 2 do documento

Figura 3.17: Esquema de visualização em uso na ferramenta *Snap Together Visualization*

PG (defeito mais reportado, como observado no Gráfico de Dispersão), e sua indicação pelo participante de número 15 (linha em amarelo), observa-se na representação por Coordenadas Paralelas que esse participante não reportou esse defeito ao aplicar a técnica *Checklist* (linha intercepta o eixo *Report* rotulado em *False*). A exploração interativa dessas representações permite observar que esse foi o único participante que não reportou o defeito de número 2 do documento PG aplicando *Checklist*, o que pode ser consequência da sua pouca experiência em diversas atividades (coletadas no experimento), observada no Gráfico de Linha.

Uma vantagem da ferramenta *Snap-Together* é que ela permite derivar esquemas de visualização a partir do modelo de dados, integrando naturalmente a geração de representações visuais ao processo de análise. Dessa forma, pode-se facilitar a escolha e uso de técnicas visuais, mapeamentos e coordenações, podendo inclusive oferecer algumas visualizações “pré-definidas” para validar as hipóteses de um experimento. O esquema de visualização poderia, por exemplo, integrar o pacote de laboratório que guia a replicação de experimentos. Entretanto, limitações observadas na versão utilizada da ferramenta representam uma barreira à sua utilização imediata.

Outras iniciativas existem para prover soluções de visualização extensíveis e adaptáveis a diferentes domínios de aplicação. No âmbito do ICMC, [Shimabukuro \(2004\)](#) propôs um modelo de arquitetura de software cuja finalidade é estabelecer um arcabouço para nortear o desenvolvimento e a implementação de uma plataforma computacional extensível, que permita a geração de ambientes personalizados de apoio ao processo de descoberta de conhecimento. A plataforma deve permitir que o usuário configure seu próprio ambiente de visualização a partir de módulos disponíveis, e estabeleça suas próprias coordenações entre visualizações de sua escolha. Um requisito fundamental para garantir o uso integrado e coordenado de múltiplas visualizações é que cada módulo seja capaz de trocar mensagens e informações com os demais. Para isso foi definido um protocolo de comunicação que implementa um conjunto de métodos predefinidos que garante a funcionalidade de coordenação.

A partir do modelo foi desenvolvida a plataforma InfoVis ([Shimabukuro, 2004](#)), que integra múltiplas técnicas de visualização e cuja arquitetura é organizada em dois núcleos: um núcleo básico, que agrega as funcionalidades necessárias ao gerenciamento do ambiente, e um núcleo funcional, que agrega os componentes de visualização e de acesso aos dados ([de Oliveira e](#)

Shimabukuro, 2004, Shimabukuro, 2004, Shimabukuro et al., 2004, Moreno, 2006). Uma visão geral da arquitetura da plataforma é apresentada na Figura 3.18. Uma versão inicial do ambiente que implementa essa plataforma está disponível, mas ainda em processo de desenvolvimento.

Os gerenciadores se intercomunicam diretamente.

Os componentes funcionais se comunicam através do Núcleo Básico, situação indicada pelas setas tracejadas.

Figura 3.18: Visão geral da arquitetura da plataforma InfoVis (Shimabukuro, 2004)

3.6 Ferramentas

Técnicas de Visualização de Informação e de Mineração Visual de Dados são apoiadas por ferramentas, comerciais ou não. Algumas ferramentas que implementam técnicas de visualização consideradas potencialmente úteis na exploração visual dos dados alvo deste estudo foram exploradas, sendo o principal critério de escolha a disponibilidade para fins acadêmicos e o conjunto de técnicas oferecidas pela ferramenta. Tais ferramentas são apresentadas a seguir.

De modo geral, as ferramentas de visualização tratam os dados já organizados em tabelas e, algumas vezes, adotam formatos de dados específicos. Assim, lembrando o modelo de Visualização proposto por Card et al. (1999) (Figura 3.1), as ferramentas consideram que a primeira etapa, de Transformação dos Dados, extrapola suas funcionalidades, podendo oferecer algum controle sobre as consultas às tabelas, mas apenas a partir de dados já pré-formatados.

3.6.1 Spotfire Decision Site

A ferramenta *SpotFire Decision Site*, desenvolvida inicialmente por Ahlberg e Wistrand (1995) em um contexto acadêmico, é hoje uma ferramenta comercial que agrega um conjunto amplo

de recursos que inclui diversas técnicas de visualização e interação, e permite que o usuário execute consultas interativas sobre representações gráficas. A discussão a seguir refere-se à versão 6.0.0 e ao seu uso na exploração dos dados coletados em replicações do projeto *Readers*.

A ferramenta possibilita a manipulação de dados mantidos em arquivos texto no padrão CSV, em planilhas do Microsoft Excel, bem como em bases de dados mantidas por sistemas gerenciadores como Oracle, Sybase e MS SQL Server. A seleção dos dados de interesse pode ser efetuada por comandos SQL, que podem ser salvos para posterior reuso.

Na Figura 3.19 uma representação gráfica (A) entre treze criadas é apresentada: na grande região central há um gráfico de linhas, chamado *Profile Chart*². As poli-linhas que representam cada elemento do conjunto de dados interceptam os eixos dos respectivos atributos normalizados localmente. Cada eixo é mapeado em uma escala de zero a cem, em percentual (podendo utilizar uma única escala normalizada para todos os eixos). Dentre as linhas apresentadas, percebe-se que há um conjunto em maior espessura (B) em tom azulado, o que indica elementos gráficos selecionados na interface para exibição na representação textual, na parte inferior da figura (C). Na última linha da figura pode-se observar que a base de dados exibida contém 1242 registros, sendo que apenas 64 foram selecionados para apresentação, dentre estes, 6 estão selecionados e são listados abaixo da palheta de gráficos gerados.

No lado direito da Figura 3.19 (D) encontram-se os controles interativos de filtragem (consultas dinâmicas). Tais controles são gerados a partir de uma verificação dos dados a serem representados e seus conteúdos são utilizados como parâmetros para a criação da interface. As diversas técnicas implementadas nessa ferramenta, com os recursos de filtragem e coordenação entre visualizações, constituem um ambiente que facilita a abordagem interativa de exploração visual, mas o custo pode ser um obstáculo, principalmente se considerado os recursos disponíveis para pesquisas³.

3.6.2 *XmdvTool*

XmdvTool (Ward, 1994) é software de domínio público para visualização exploratória de dados multivariados. A versão utilizada no presente trabalho (5.0) disponibiliza quatro técnicas para apresentação de dados planares (*flat data*): Gráficos de Dispersão, Ícones Estrela, Coordenadas

²Nome atribuído a uma representação idêntica a Coordenadas Palalelas, com variação na escala de valores

³A licença utilizada neste trabalho foi adquirida com recurso de outro projeto

Figura 3.19: Instantâneo da ferramenta *SpotFire*

Paralelas e Aninhamento Dimensional. O *XmdvTool* utiliza como fonte de dados arquivos no formato OKC. Esse formato inclui um bloco de metadados, no início do arquivo, com o número de atributos, o número de registros, os nomes dos atributos, além de informações como os valores mínimo e máximo de cada atributo e a cardinalidade para o Aninhamento Dimensional, que facilita a construção dos gráficos. Após o cabeçalho, seguem os registros, sendo que cada linha contém os valores dos atributos de cada registro, separados por um espaço (ver Figura 3.20). Além disso, por incorporar representações hierárquicas, o *XmdvTool* utiliza dois outros arquivos que descrevem as relações hierárquicas em *flat files*, gerados a partir do arquivo padrão OKC.

Na Figura 3.21 um instantâneo do *XmdvTool* com uma representação em Coordenadas Paralelas é apresentado. Pode-se observar que o fundo possui uma região de cor azul que delimita o espaço de registros selecionados (*brushed*), destacados em vermelho. Em cada eixo (atributo) há um ponto mínimo e um ponto máximo que determinam o intervalo selecionado, definindo a região em azul. O mesmo processo de *brushing*, adequadamente adaptado, está disponível para as outras representações visuais.

```

1  18 698
2  Class
3  Type
4  Analyst
5  Developer
6  English
7  Experience
8  Manager
9  Tester
10 Using_Req
11 Writing_Req
12 Replication
13 Subject
14 Report_Doc
15 Technique
16 Persp
17 Report
18 Report_Error
19 List_Defect
20 1.000000 2.000000 3
21 1.000000 6.000000 3
22 0.000000 24.000000 3
23 0.000000 60.000000 3
24 1.000000 3.000000 3
25 0.000000 2.500000 3
26 0.000000 12.000000 3
27 0.000000 36.000000 3
28 0.000000 24.000000 3
29 0.000000 24.000000 3
30 1.000000 1.000000 3
31 1.000000 18.000000 3
32 1.000000 1.000000 3
33 1.000000 2.000000 3
34 1.000000 4.000000 3
35 0.000000 2.000000 3
36 1.000000 37.000000 3
37 0.000000 19.000000 3
38 1.000000          2.000000      24.000000      60.000000      ...
39 2.000000          1.000000      24.000000      60.000000      ...
40 2.000000          1.000000      24.000000      60.000000      ...

```

Figura 3.20: Trecho de um arquivo padrão OKC utilizado pelo sistema *XmdvTool*

Fua et al. (Fua et al., 1999a,b, 2000) apresentam uma abordagem de visualização interessante que integra mineração para permitir a visualização por Coordenadas Paralelas de conjuntos de dados muito grandes. Eles aplicam um algoritmo para gerar um agrupamento hierárquico dos dados (MinMax Tree) (Doshi et al., 2003). Assim, obtêm uma hierarquia de agrupamentos (*clusters*) representada por uma árvore cujos nodos folhas representam os registros individuais, e cuja raiz representa o agrupamento global que inclui todo o volume. O usuário pode selecionar diferentes níveis de detalhe e abstração para visualizar os dados, navegando por uma representação dessa árvore. Um esquema de codificação por cores baseado na proximidade computada entre os agrupamentos é adotado na exibição (Fua et al., 1999b, 2000). Na Figura 3.22 é apresentada a interface de navegação: dentro de um contorno triangular que representa a árvore de agrupamentos (a), há um contorno representando os nodos folha (c); a linha (b) pode ser movi-

Figura 3.21: Instantâneo da ferramenta *XmdvTool* com uma representação em Coordenadas Paralelas

mentada pelo usuário para determinar o nível de detalhe a ser apresentado (profundidade para um corte na hierarquia de grupos), e é colorida segundo a escala de cores (f); os dois pontos de controle apontados por (e) na base do triângulo podem ser movimentados pelo usuário para delimitar a região de interesse (d). A linha (b) permite ao usuário navegar entre representações com detalhes dos grupos contidos dentro e fora do triângulo que delimita a região de interesse, de modo independente (d). A Figura 3.23 apresenta uma seqüência de visualizações de um conjunto de dados em diferentes níveis de detalhe (obtidas pela variação da linha com identificação (b) da Figura 3.22).

Esta abordagem é útil quando o volume de dados a ser apresentado é grande, ocasionando muita sobreposição de poli-linhas, o que dificulta a interpretação da visualização. Entretanto, a seqüência em que os atributos são apresentados no conjunto de dados altera o resultado do agrupamento e, portanto, a visualização resultante. Apesar de ser possível “navegar” no agrupamento hierárquico (Figura 3.22), o processo de navegação e a interpretação das visualizações resultantes exigem algum treinamento.

Figura 3.22: Interface para controle de exibição do agrupamento hierárquico na ferramenta XmdvTool (Fua et al., 1999a).

Figura 3.23: Duas visualizações exibindo um conjunto de dados em diferentes níveis de detalhe

3.7 Considerações Finais

Como apresentado, o modelo de visualização proposto por Card et al. é bastante genérico. Por não especificar, por exemplo, as atividades a serem efetuadas em cada etapa, tampouco considerar características específicas dos dados, é necessário especificar como instanciá-lo no contexto de experimentação e replicação. Sem uma instanciação apropriada fica difícil, ou até mesmo impossível, uma aplicação consistente do processo de visualização pelos envolvidos no processo de experimentação, considerando a necessidade de se conduzir análises consistentes ao longo das diversas replicações. Observa-se que ao replicar experimentos há uma dificuldade muito grande em captar o conhecimento tácito quando uma definição é vaga, o que torna necessário explicitar as atividades em cada etapa. O pacote de laboratório, que deve incluir os procedimentos de análise, não prevê análise por visualização exploratória. Para que possa ser

incorporada, o pacote deverá também detalhar os procedimentos adotados e as decisões tomadas durante a sua execução, bem como os procedimentos adotados para a obtenção de visualizações relevantes. Isto é tanto mais relevante quando se considera que quem gera as representações visuais não é necessariamente o analista interessado no resultado do processo de análise.

Por se tratar de assunto de interesse deste trabalho, o processo de Mineração de Dados foi introduzido com o objetivo de mostrar a possibilidade de integrar Visualização e Mineração. As principais categorias de Técnicas de Visualização Exploratória foram apresentadas, juntamente com o potencial de aplicação considerando o contexto deste trabalho. De um modo geral, observa-se que as técnicas Iconográficas e Orientadas a Pixel não são as mais adequadas, a princípio, à análise dos dados disponíveis. Já as técnicas de Projeção Geométrica e Hierárquicas mostram-se adequadas não só pela característica dos dados a serem explorados, mas também pelos recursos disponíveis – implementados em ferramentas acessíveis.

Observa-se também que as ferramentas, em geral, limitam-se a implementar operações de mapeamento visual e transformações de visões, não permitindo a interação ao longo de todo o processo (inclusive o de preparação de dados), como sugerido no modelo de Visualização proposto por Card et al. (ver Figura 3.1). O *SpotFire Decision Site*, por exemplo, depois de obtidos os dados para a visualização, não permite o retorno a etapas anteriores, tampouco oferece recursos para a visualização de dados hierárquicos. Por outro lado, apresenta uma boa interface para filtrar dados e possui um rico conjunto de recursos integrados com visualizações para apresentação/sumarização. O *XmdvTool*, por sua vez, permite a visualização de dados hierárquicos, mas é mais difícil de usar e é uma ferramenta menos robusta.

Ressalta-se também a importância do uso de visualizações coordenadas, permitindo múltiplas visões simultâneas e integradas de um conjunto de dados, o que possibilita explorar aspectos complementares entre diversas técnicas. De modo geral, observa-se a existência de recursos para análise exploratória, mas não há diretrizes sobre como instanciá-las no contexto de experimentação. Esses fatores motivaram a definição de um processo de análise visual exploratória para dados de Engenharia de Software Experimental, exposto no próximo capítulo.

Processo de Visualização Estendido para Engenharia de Software Experimental

A análise visual de dados de experimentação em Engenharia de Software requer a adequação tanto do processo de experimentação quanto do processo de análise. Do ponto de vista do processo de experimentação, é necessário adequá-lo para inserir o procedimento exploratório como parte de suas atividades. Por outro lado, o processo de exploração visual precisa ser instanciado, e as tarefas executadas durante a análise registradas no Pacote de Laboratório, para futuras replicações do procedimento exploratório. Para isso foi proposto um processo de visualização estendido e instanciado nesse contexto.

Outro ponto tratado neste capítulo é a condução de meta-análise exploratória – a análise exploratória integrada de múltiplos conjuntos de dados, obtidos em diferentes experimentos (e replicações) – por meio da instanciação do processo de visualização estendido. Embora tal procedimento de análise seja conduzido fora do escopo da realização de um experimento isolado, os ciclos do FIRE (ver Capítulo 2) que ilustram as evoluções intra e inter-grupos definem o ambiente dessa instanciação. Para tanto, uma abordagem para a condução de meta-análise ex-

ploratória, também foi proposta. Dessa forma, o capítulo encontra-se organizado como segue:

- Na Seção 4.1 são apresentadas adequações ao processo de experimentação para a inserção de visualização exploratória, contemplando as exigências para empacotamento de procedimentos;
- Na Seção 4.2 uma breve argumentação sobre meta-análise exploratória no contexto deste trabalho é apresentada, juntamente com a abordagem proposta para conduzir esta atividade. Destaca-se que a meta-análise exploratória introduz atividades específicas ao processo de visualização, quando instanciado para tal atividade;
- Na Seção 4.3 o processo de visualização é apresentado, juntamente com as diretrizes para a sua instanciação;
- Na Seção 4.4 são apresentadas as atividades iniciais do processo de visualização estendido, vinculadas ao tratamento dos dados a serem analisados;
- Na Seção 4.5 são apresentadas as atividades do processo de visualização vinculadas à criação de representações visuais;
- Na Seção 4.6 é apresentado um arcabouço para apoiar a instanciação do processo de visualização proposto.

4.1 Visualização em Experimentação: Adequações e Desdobramentos

As três categorias de visualização mencionadas no Capítulo 3 (Apresentação, Confirmação ou Exploração) podem ser aplicadas para apoiar diferentes tarefas de análise, pois há diferentes papéis envolvidos no processo de experimentação, cada um com objetivos específicos. Esses objetivos específicos motivam a identificação de tais papéis para que sejam estabelecidos os respectivos escopos de atuação, juntamente com as técnicas em potencial. Os principais estágios de análise em um processo de experimentação que podem ser apoiados por técnicas de visualização foram definidos a partir dos estágios identificados por Wohlin et al. (1999), como integrantes de um processo típico de análise estatística de dados: o estágio de Pré-Análise envolve o processamento dos dados brutos e possíveis tratamentos, incluindo a redução dos dados

tanto em número de elementos quanto em número de atributos, quando aplicável; Verificação de Hipóteses, que é a principal atividade da análise estatística; Síntese, para a apresentação de resultados (que pode ser usada para a consolidação de resultados); e Meta-Análise – análise de resultados de múltiplos experimentos, ou replicações.

Quanto aos papéis envolvidos no processo de experimentação, foram identificados três: o Projetista do experimento, o Replicador e o Participante. Os dois primeiros têm como objetivo verificar a(s) hipótese(s) tratada(s) no experimento. Podem também ter como objetivo a análise de variáveis e hipóteses com o intuito de aprimorá-las, como apontado por [Carver \(2003\)](#) (ver Figura 2.8). Entretanto, do ponto de vista da análise de dados, há uma diferença entre esses papéis: supõe-se que o Projetista tenha acesso a dados de múltiplas replicações, enquanto o Replicador tenha acesso apenas aos dados da replicação por ele conduzida. Tal suposição parte do pressuposto que o replicador compartilhe os dados coletados com o projetista do experimento. O terceiro papel identificado – Participante – pode ter interesse em obter informações sobre o seu desempenho, analisar seus resultados e compará-los com os do grupo. Para os três papéis destacados pode-se assumir duas situações distintas: o próprio envolvido é o responsável pela exploração visual, ou há um especialista em visualização que apóia o processo, o que indica um papel extra para cada um.

Figura 4.1: Perspectivas de uso de Técnicas de Visualização em um processo de experimentação

Na Figura 4.1 é apresentada uma visão pessoal do potencial espaço de aplicação de técnicas

de visualização na análise de dados de estudos experimentais, considerando três eixos representando, respectivamente, o papel do usuário, o estágio de análise e a natureza da tarefa a ser apoiada pela visualização (Apresentação, Confirmação ou Exploração). O tamanho do círculo, embora não tenha uma conotação quantitativa precisa, é proporcional à percepção do potencial de uso. O primeiro plano, por exemplo, sintetiza o potencial de uso de técnicas de visualização sob a perspectiva do Projetista do Experimento: técnicas de visualização exploratória podem ser úteis em tarefas de Pré-Processamento de dados, Verificação de Hipóteses e Meta-Análise, assim como técnicas de apresentação podem ser úteis no estágio de Síntese. Similarmente o plano central sugere que Replicadores podem se beneficiar de técnicas de apresentação no estágio de Síntese e de visualização confirmatória no estágio de Análise Confirmatória. A representação da figura implica que um replicador não realiza meta-análise exploratória, embora isso seja possível se o mesmo conduzir múltiplas replicações. Neste caso, ele passa a assumir a possibilidade mapeada para o Projetista de Experimento, pois pode conduzir meta-análise exploratória com os dados das múltiplas replicações conduzidas. No último plano é mostrado que técnicas visuais podem ser usadas para exibir resultados a Participantes, que podem, por exemplo, estar interessados em informação sobre seu desempenho ou nas conclusões do estudo.

O processo de experimentação deve ser adaptado à incorporação da Análise Visual. Diante da possibilidade de usar múltiplas abordagens para análise de dados – Análise Estatística e Visualização de Informação –, é necessário adequar o processo de experimentação separando a atividade de “Análise e Interpretação”, proposta por Wohlin et al. (1999) (ver Figura 2.1), em duas: “Pré-Análise” e “Análise e Interpretação” propriamente dita (Garcia et al., 2004a). A separação tem por objetivo permitir que a Análise Estatística e a Análise Visual sejam conduzidas por diferentes analistas, sem interferência mútua decorrente do pré-processamento de dados. A separação permite o tratamento prévio dos dados coletados no experimento de modo a usar o mesmo conjunto de dados para subsidiar ambas as abordagens, evitando que considerações isoladas interferiram nos resultados e levem a conclusões conflitantes.

Na *Pré-Análise* são feitas as tarefas *Preparação do Conjunto de Dados* e *Avaliação da Qualidade do Experimento*. A primeira delas tem como foco a análise dos dados coletados e a classificação dos defeitos reportados, como proposto por Wohlin et al. (1999). Na segunda o foco é o registro de informações sobre a condução do experimento, principalmente variações

Figura 4.2: Processo de Experimentação Modificado a partir de Wohlin et al. (1999)

Figura 4.3: A atividade de Análise considerando meios alternativos (Visualização e Mineração) além da Estatística

com relação ao projeto experimental, para que se tenha dados sobre a qualidade da execução do experimento e, conseqüentemente, sobre a qualidade dos dados coletados. A avaliação da qualidade foi introduzida com o intuito de ter dados para subsidiar futuras meta-análises, o que até então não tinha sido considerado. Na Figura 4.2 é apresentado o processo de experimentação modificado, e na Figura 4.3 é apresentada a atividade de *Análise e Interpretação*. No procedimento de análise exposto na Figura 4.3 foram introduzidas a *Definição do Problema* e a *Avaliação dos Dados do Experimento*, e as demais atividades foram generalizadas para admitir múltiplas abordagens de análise. A *Definição do Problema* considera que seja definida uma hipótese a ser verificada, quando conduzido um procedimento estatístico de cunho confirmatório. Isso não inviabiliza a instanciamento de um processo de investigação exploratória, que trate hipóteses não definidas antecipadamente como objetivo do experimento, mas formuladas pelo analista na tentativa de prospectar os dados. As novas hipóteses formuladas podem ser verificadas usando o próprio conjunto sob investigação, se possível, ou podem motivar novos experimentos. Quanto à atividade *Avaliação dos Dados do Experimento*, o objetivo é que o analista adquira conhecimento sobre a condução do experimento, para que a análise, e conseqüentemente as conclusões, sejam consistentes. A aquisição de conhecimento sobre a condu-

ção do experimento é essencialmente importante quando a análise não é conduzida pelo próprio replicador.

Nesse cenário, lições aprendidas em diversas análises visuais conduzidas sobre os dados das replicações do Projeto *Readers*, simulando os papéis de *Projetista* e *Replicador*, permitiram determinar diretrizes para a condução de análises visuais no contexto de experimentação. Tais diretrizes integram o processo de exploração visual modificado para especificar as etapas da tarefa de análise, beneficiando os analistas de dados de estudos experimentais. Agregando-se o processo e as diretrizes ao pacote de experimentação, que agrupa e descreve as atividades a serem efetuadas, são fornecidos os subsídios necessários para que a análise dos dados coletados em estudos experimentais seja feita de maneira consistente e de modo a permitir a comparação e a agregação de resultados de diferentes replicações. Esse passo é fundamental para viabilizar a análise de dados de múltiplos experimentos – aqui chamada de meta-análise exploratória –, e a comparação de resultados de replicações diferentes do mesmo experimento (isso pode não ser viável no caso de experimentos de estrutura e natureza diferentes). O processo de visualização modificado – ViDA_{ESE} – é exposto na Seção 4.3, juntamente com as diretrizes estabelecidas. Mas antes, faz-se necessário uma explanação sobre a abordagem para meta-análise exploratória, pois o processo de visualização modificado pode ser instanciado para a tal atividade, e por isso há tarefas quem abrangem tal instanciação.

4.2 Abordagem para Meta-Análise Exploratória

O conceito de meta-análise é o de combinar resultados de vários estudos que tratam de hipóteses similares, como discutido por [Miller \(2005\)](#). Há várias razões para conduzir meta-análises, como aumentar o tamanho da amostra e verificar a sustentação de resultados sob condições experimentais variadas. Um aspecto importante é a identificação e entendimento de razões para qualquer variação ou discrepância entre estudos. Uma investigação sobre variações entre estudos pode ser planejada antes de qualquer análise, ou pode ser feita em resposta à análise prévia que tenha identificado heterogeneidade entre um conjunto de estudos. Entender como e porque estudos variam permite ao pesquisador entender melhor se tais estudos podem ser combinados e com que métodos ([Miller, 2000](#)).

A visão “tradicional” de meta-análise assume que o pesquisador busca investigar um conjunto predefinido de hipóteses. Se o interesse é explorar dados de um ou mais experimentos

para descobrir relações não-antecipadas e identificar novas hipóteses, uma abordagem estatística orientada por hipóteses não é efetiva. Neste caso, uma abordagem exploratória baseada em visualização pode ser útil.

Primeiramente pode-se agregar dados dos múltiplos experimentos e o primeiro passo para tal agregação consiste em identificar diferenças entre conjuntos de dados oriundos dos diferentes experimentos – análise de meta-dados –, decorrentes ou da evolução do Pacote de Laboratório ou de diferenças de objetivo entre os estudos em questão. O objetivo da análise de meta-dados é estabelecer os critérios para a agregação, que por sua vez, definem parâmetros e potenciais restrições para a análise exploratória, especialmente para comparar visualizações e conclusões. É possível identificar situações que podem introduzir requisitos específicos e sugerir diferentes tarefas de visualização, como: a) alterações no procedimento de coleta de dados – diferentes tipos de dados e/ou escalas não permitem a comparação ou visualização direta; b) dados ausentes – dados não coletados em um experimento particular; c) variáveis que foram expandidas ou resumidas em variáveis mais refinadas ou mais gerais, dependendo do objetivo do estudo.

Figura 4.4: Abordagem proposta para a meta-análise exploratória

A abordagem proposta para a meta-análise exploratória de dados oriundos de múltiplos experimentos e agregados é esquematizada na Figura 4.4. A estratégia consiste em comparar o conjunto de representações visuais geradas ao explorar dados de um experimento isolado (replicação em foco) e representações similares geradas a partir dos dados agregados: na Figura 4.4 as

setas contínuas conectando os círculos representam comparações entre representações visuais e as setas tracejadas representam a similaridade requerida entre os mapeamentos visuais para viabilizar a comparação dos resultados. Dados coletados em cada replicação devem ser submetidos ao mesmo processo de preparação: os resultados de cada nova replicação podem ser agregados àqueles coletados em replicações anteriores e, com isso, pode-se observar o impacto dos dados “novos” no conjunto de dados já existente.

Vale ressaltar que a meta-análise exploratória, assim como as diferentes perspectivas da aplicação de visualização apresentadas na Figura 4.1, introduzem tarefas na condução da análise visual que devem ser incorporadas ao processo de visualização. Entretanto, a especificidade das tarefas introduzidas requer que a instanciamento do processo considere a perspectiva do analista, trazendo ao processo a característica da sensibilidade ao contexto de sua aplicação – ou seja, as tarefas do processo dependem do papel assumido por quem está conduzindo a análise.

4.3 Análise Visual de Dados para Engenharia de Software Experimental

O processo de visualização apresentado no Capítulo 3 foi estendido considerando sua aplicação no contexto de experimentação e da condução de replicações. Foram estabelecidas diretrizes para gerar visualizações por meio da aplicação de técnicas, no caso, hierárquicas e de projeção geométrica aos dados de estudos experimentais. Tais diretrizes permitem ao analista a liberdade de escolha de técnicas e mapeamentos, viabilizando a ampliação da cognição de modo personalizado.

O *ViDA_{ESE}* – *Visual Data Analysis for Experimental Software Engineering* – descreve as etapas e as respectivas tarefas de um processo de busca de conhecimento, seja ele direcionado (análise confirmatória) ou não (análise exploratória). Inicialmente, as atividades do processo foram agrupadas considerando o espaço de atuação: no Espaço de Dados ou no Espaço Visual. O primeiro grupo de atividades tem por objetivo preparar os dados, adequando-os à análise visual de acordo com o seu objetivo. No segundo grupo representações visuais são geradas e manipuladas interativamente para apoiar a análise propriamente dita.

As descobertas propiciadas por uma visualização são diretamente influenciadas pelo pré-processamento aplicado aos dados no procedimento de conversão para um formato tratável pela

técnica/ferramenta. A literatura em Visualização considera, geralmente, que a preparação dos dados está fora de escopo da visualização propriamente dita, o que introduz uma dificuldade adicional no contexto de replicações. Em muitos casos, é necessário adaptar a organização e o formato dos dados às técnicas, ou especificamente às ferramentas que as implementam. A maioria das ferramentas de visualização não manipula a contento uma ampla variedade de tipos e formatos, pois são geralmente desenvolvidas para propósitos específicos. Por exemplo, dados do tipo numérico, de natureza contínua ou discreta, são facilmente mapeados para atributos gráficos, mas há um grande número de atributos representados por valores ordinais e categóricos em dados de estudos experimentais. Aplicar a dados nominais os mesmos mapeamentos visuais aplicados a dados numéricos pode induzir a erros de interpretação. Qualquer conversão de tipos para adequação a uma técnica específica deve ser registrada e mantida, não só para apoiar a interpretação, como também para compor o pacote do experimento – permitindo a repetição do procedimento sobre novos dados equivalentes.

Portanto, os primeiros passos do processo têm como foco os dados. Dada uma *tarefa* de análise, com um objetivo específico, o primeiro passo consiste em adequar os dados à tarefa em questão. No modelo do processo de visualização, apresentado na Figura 3.1, dados brutos são transformados em dados tabelados e submetidos à visualização. Como a própria denominação sugere, dados brutos não possuem qualquer organização explícita. A primeira adequação ao processo foi a introdução de uma interação em *Dados Brutos*, que representa possíveis alterações em elementos de dados sem implicar em organizar o conjunto em tabelas. Também foi introduzida uma nova etapa – *Seleção* – para gerar os *Dados Seleccionados*, com o objetivo de permitir que múltiplas seleções sejam feitas simultaneamente e submetidas à geração de visões (representações visuais). O *Mapeamento Visual* define as *Estruturas Visuais* a serem usadas para criar a visualização, ou seja, que define o mapeamento entre os atributos do conjunto de dados e os elementos gráficos que os representam. Por exemplo, um dado atributo pode ser mapeado para cor e, portanto, um mapa de cores que relacione os valores assumidos pelo atributo e uma cor deve ser especificado. Por meio das *Transformações Visuais*, é possível definir diferentes *Visões* a partir de uma visualização.

Além disso, admite-se a possibilidade de múltiplas instâncias de *Dados Seleccionados* para que sejam conduzidos procedimentos paralelos (eventualmente coordenados) de exploração vi-

Figura 4.5: ViDA_{ESE} : *Visual Data Analysis to Experimental Software Engineering*

sual, permitindo a exploração de aspectos complementares entre técnicas. Na Figura 4.5 é apresentado o processo ViDA_{ESE} com as adequações propostas. Na figura que ilustra o processo, as setas (em verde) entre duas *Estruturas Visuais* e duas representações visuais – *Visões* – denotam que deve haver correspondência entre elas, ou seja, os mapeamentos utilizados devem gerar visualizações que permitam explorar aspectos complementares entre técnicas. A correspondência entre os mapeamentos visa a facilitar o processo cognitivo de interpretação de múltiplas visões dos mesmos dados.

Conforme mencionado no Capítulo 3, a organização dos dados pode afetar o modelo visual e diferentes técnicas podem demandar diferentes organizações. Isso ocorre, por exemplo, quando a sequência dos atributos no arquivo influencia a representação visual gerada. Em outro exemplo, é possível introduzir relações semânticas de hierarquia entre os atributos e visualizá-las por

intermédio de visualizações hierárquicas. Entretanto, as relações semânticas devem ser registradas separadamente, pois diferentes usuários envolvidos no processo de análise, assumindo diferentes papéis, podem ter interesses específicos, ou seja, uma relação pode ser interessante para um papel e não para outro. Além disso, pode ser necessário aplicar a análise a outros dados, por exemplo em atividades de meta-análise exploratória. A descrição dos significados das relações semânticas pode facilitar o entendimento dos dados, e subsidiar a interpretação do mapeamento visual. Essas relações semânticas, mantidas separadamente, podem ser utilizadas como *template* para compor o pacote de laboratório do experimento, definindo, por exemplo, como gerar um mapeamento visual para atingir um dado objetivo de análise. Dessa forma, os *Dados Selecionados* podem ser estruturados para adequá-los a objetivos e ferramentas específicas. Tal fato motivou a definição de um arcabouço para visualização, apresentado na Seção 4.6.

Para cada atividade indicada no processo proposto foram definidas diretrizes para sua condução, como exposto na Tabela 4.1. Tais diretrizes devem ser sensíveis ao contexto da condução da tarefa, podendo variar, por exemplo, quando muda o objetivo da investigação, ou quando o processo é aplicado em tarefas de análise ou meta-análise. Por exemplo, a atividade *Adequar dados coletados ao objetivo da análise* pode ser o simples descarte de defeitos reportados repetidamente por um participante nos experimentos PBR (ver Seção 4.4), ou a junção de diferentes categorias de experiências de participantes para compensar diferenças na coleta dos dados, como efetuado por Carver (2003), que analisou algumas hipóteses por grupos de participantes separados em Experientes e Não-Experientes.

A exploração visual se dá em ciclos, como sugere a abordagem para meta-análise (ver Figura 4.4), sendo que cada ciclo é conduzido com um objetivo específico, perfazendo todas as atividades do processo, e seguindo as diretrizes estabelecidas. Como resultado de cada ciclo de análise, as atividades e conclusões devem ser acompanhadas de registros de decisões tomadas ao longo da mesma. Dessa forma, é possível repetir o procedimento, como requerido na replicação de experimentos. É possível também validar e verificar tal procedimento quando estudos são conduzidos por terceiros, ou seja, o projetista do experimento original não está envolvido na replicação e na análise dos dados coletados. Observa-se que as diretrizes propostas estão separadas de acordo com o seu espaço de ocorrência – espaço de dados ou espaço visual. A mesma separação é mantida na instanciação do processo, apresentada na Seção 4.4.

Tabela 4.1: Diretrizes para a análise visual de dados no contexto de experimentação (ESE)

Espaço de Atuação	Etapa do Processo de Análise	Atividade a ser desempenhada
Espaço de Dados	Iteração em <i>Dados Brutos</i>	<ul style="list-style-type: none"> • Análise de meta-dados • Adequar dados coletados ao objetivo da análise; • Manter cópia da instância da coleção de dados pré e pós-adequação; • Registrar as adequações (decisões, estado inicial e final do conjunto de dados, e justificativas).
	Iteração em <i>Dados Tabelados</i>	<ul style="list-style-type: none"> • Selecionar registros de interesse para a criação da representação visual - considerar o objetivo da análise; • Estabelecer a sequência dos atributos; • Converter os dados, adequando-os à técnica/ferramenta; • Registrar as decisões tomadas e transformações efetuadas (decisões, estado inicial e final do conjunto de dados, e justificativas).
	Organização dos Dados (Estruturas)	<ul style="list-style-type: none"> • Definir a organização dos dados • Registrar as decisões tomadas e transformações efetuadas (decisões, estado inicial e final do conjunto de dados, e justificativas).
Espaço Visual	Escolha da Técnica e da Ferramenta	<ul style="list-style-type: none"> • Definir as múltiplas técnicas e as ferramentas a serem utilizadas; • Registrar as decisões tomadas (inclusive sobre versões das ferramentas utilizadas).
	Definição das Estruturas Visuais	<p>Essa atividade é altamente dependente da escolha da técnica/ferramenta</p> <p>Deve-se:</p> <ul style="list-style-type: none"> • Manter a equivalência entre as múltiplas representações visuais (equivalência de mapeamentos); • Registrar as estruturas utilizadas;
	Interação com a Visualização	Exploração interativa e iterativa das representações visuais obtidas para análises

Vale destacar que tanto dados coletados em replicações conduzidas no Projeto *Readers*, quanto dados coletados em outros experimentos foram utilizados (explorados) neste trabalho. Quanto às replicações do Projeto *Readers* realizadas no ICMC-USP, os participantes das duas primeiras – nomeadas R1 e R2 – foram estudantes com baixa experiência em desenvolvimento, e as replicações nomeadas R3 e R4 tiveram a participação de estudantes mais experientes e profissionais, respectivamente. Todos os experimentos envolveram a comparação entre PBR e *Checklist*. Também comparando essas técnicas, pesquisadores de Maryland replicaram o experimento original com a participação de alunos de graduação e de pós-graduação, replicações nomeadas Fall97 e Fall98, respectivamente. Dados do experimento original – NASA – também

foram utilizados. Adicionalmente, [Carver \(2003\)](#) conduziu dois experimentos: um para investigar a experiência de um inspetor em PBR (Fall01) e outro cujo foco foi avaliar a influência da especificidade do processo de inspeção (Fall02).

4.4 Atividades no Espaço dos Dados

A primeira extensão no processo original de [Card et al. \(1999\)](#), tem como foco os dados brutos, e é apresentada na Figura 4.5 pela iteração sobre *Dados Brutos*. A motivação é que pode ser necessário tratar os dados antes de sua organização em tabelas. No caso específico dos dados manipulados no contexto do Projeto *Readers* tal atividade é necessária, por exemplo, quando um participante reporta um mesmo defeito mais de uma vez. Pode-se observar tal fato na Figura 4.6, que apresenta os defeitos do documento PG reportados pelos participantes de uma replicação: um mesmo defeito é reportado mais de uma vez pelo mesmo participante (por exemplo, o participante S3 reportou três vezes o defeito 5). O tratamento dessa ocorrência afeta a interpretação dos mapeamentos visuais e, portanto, decisões a respeito devem ser registradas e consideradas ao interpretar as visualizações. Por exemplo, considerando as medidas de efetividade e eficiência adotadas e o uso de todos os defeitos reportados – inclusive os repetidos –, ao analisar a Efetividade pode-se criar representações visuais que sugerem medidas maiores do que as reais. O tratamento dos dados também é especialmente relevante na meta-análise exploratória, pois “ajustes” devidos a variações no formato da coleta dos dados têm impacto direto nas representações visuais, e devem também ser registrados. Por exemplo, em algumas replicações a experiência do participante em um determinado aspecto foi coletada em meses e em outras em uma escala (de “pouco” a “muito”), o que requer uniformização para meta-análise exploratória.

A última etapa de preparação no espaço de dados – *Seleção* – implica em selecionar e organizar os dados tabelados de acordo uma estrutura. Em algumas ferramentas a estrutura dos dados de entrada é um parâmetro para gerar esquemas de exploração e visualização. Por exemplo, as visualizações em Coordenadas Paralelas em múltiplos níveis de detalhe (ver Seção 3.6.2) uma abordagem implementada no XmdvTool para definir um agrupamento hierárquico considera a ordem de apresentação dos atributos no arquivo de entrada. Diferentes seqüências geram diferentes agrupamentos e diferentes visualizações. Em seguida, os dados podem ser submetidos às ferramentas, para gerar representações visuais significativas para a análise em questão.

Defeitos - PG		The number in each grid square corresponds to the subject's number on their defect list and the time that the defect was found.																																
Class	C	C	C	O	O	O	O	O	C	C	C	O	C	C	O	C	C	C	C	O	C	C	C	O	C	O	C	O	C	O	C			
Type	II	IF	A	II	MI	MI	II	MI	MI	IF	II	A	II	II	MI	A	II	II	MI	II	E	MD	MI	MD	O	MI	A	MI	MI	MD	MI	A		
	OJA																OJA																	
Current	II	IF	A	II	O	O	II	A	O	IF	II	A	II	II	O	O	II	II	II	O	II	E	MD	O	MD	O	O	O	O	MD	O	A		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	Total	
S9 (PBR - D)		1								3																						2	3	
S10 (PBR - D)		6							10	12					13	14																8	9	6
S12 (PBR - D)		1								5					7	8									10							4	6	
S7 (PBR - T)		1	8		2					6					7	9															5		4	6
S14 (PBR - T)										4				5	6	7																	5	6
S18 (PBR - T)	1	2			4									8	9	10																	6	6
S2 (PBR - U)	1				3					6					8	10																	5	6
S5 (PBR - U)		1			5										3	4																	4	6
S11 (PBR - U)	2	3			3					5																							3	3
S1 (Checklist)		3		4	4		4			13			11	12		14					17										7	10	4	6
S3 (Checklist)		3		4	5,6,7					10																							1	4
S4 (Checklist)		6																															6	6
S6 (Checklist)	2	3							4	5	6			6																			4	6
S8 (Checklist)	4	1			5																		9										4	6
S13 (Checklist)		1					4				6		7									11									3	6	6	
S15 (Checklist)			7		4			3		9														8								2	6	6
S16 (Checklist)		1																		13				9									2	6
S17 (Checklist)					5							7	8		8															15			6	6
Total	4	16	2	2	10	0	1	0	1	0	2	4	8	3	1	4	2	8	7	0	0	0	0	0	0	0	0	0	0	0	0	2	6	6
	18	0,2	0,9	0,1	0,6	0,0	0,1	0,1	0,2	0,4	0,2	0,1	0,2	0,1	0,4	0,4	0,0	0,0	0,0	0,1	2	0	0	0	0	0	0	0	0	0	0	0	0,1	0,3
Participantes		Defeitos																																

Figura 4.6: Dados brutos coletados em um experimento

O objetivo da análise e os dados disponíveis influenciam a escolha do par técnica/ferramenta e, conseqüentemente, o mapeamento visual indicado.

4.5 Atividades no Espaço Visual

Tendo finalizado a preparação dos dados, as atividades seguintes – expostas na Figura 4.5 – têm como foco a geração das representações visuais.

4.5.1 Mapeamento Visual

A etapa de Mapeamento Visual foi subdividida em três fases – Seleção da Técnica; Seleção e Arranjo de Atributos; e Definição das Propriedades Gráficas – apresentadas a seguir.

4.5.1.1 Seleção da Técnica

A seleção da técnica de visualização (e da ferramenta) a ser utilizada é altamente dependente do objetivo da investigação. A experiência adquirida com a condução de várias análises visuais indicou que uma forma adequada de apoiar a verificação de hipóteses pode ser combinando técnicas de projeção geométrica com uma técnica de síntese. Dentre as técnicas de projeção geométrica foram utilizadas Coordenadas Paralelas e Gráficos de Dispersão (2D e 3D) para a visualização dos registros individuais (com foco em participante, defeito, etc). Essas técnicas foram aplicadas juntamente com uma técnica que permite a sumarização dos resultados por grupos, como gráficos em barra ou pizza. Dessa forma, é possível obter uma visão detalhada e

uma visão global, paralelamente. Na Figura 4.7 é apresentado um Gráfico de Dispersão 3D que mapeia *Defeito*, *Tipo de Defeito* e *Técnica Aplicada* para os três eixos e a cor dos marcadores (esferas) enfatiza o *Tipo de Defeito* (há uma sobrecarga de mapeamento, já que o tipo também é mapeado para um dos eixos), evidenciando a distribuição dos defeitos reportados por tipo. Na Figura 4.8 é apresentada uma sumarização desses dados, exemplificando como integrar duas técnicas distintas para apoiar uma mesma análise: na Figura 4.7 observa-se individualmente quais defeitos foram reportados e a sumarização permite uma comparação quantitativa entre os grupos – no caso, *Tipo de Defeito*. Esse exemplo não trata uma hipótese previamente definida no experimento, porém ilustra a facilidade de explorar múltiplas representações visuais para investigar outros aspectos, no caso a distribuição dos *Tipos de Defeito* detectados. Conforme discutido por Garcia et al. (2004b), observa-se maior ocorrência de relatos de defeitos do tipo *Omission* e menor ocorrência de relatos de defeitos do tipo *Extraneous Information* nos documentos utilizados nas replicações.

Figura 4.7: Distribuição de defeitos reportados na replicação R1, por tipo e documento (cor enfatiza o tipo de defeito)

Ambas as representações (Figuras 4.7 e 4.8) utilizam o mesmo mapa de cores, como especificado no processo proposto (ver Seção 4.3). Observa-se também que, para o documento PG (à direita da Figura 4.7) a escala exibe defeitos enumerados até 37, embora existam apenas 32 defeitos naquele documento. Isso acontece porque uma única seleção com dados dos dois documentos foi submetida à visualização, e a ferramenta considera todos os elementos de dados

para definir a escala. Este fato sinaliza que, para algumas comparações, é necessário separar o conjunto de dados em subconjuntos, pois as ferramentas consideram o conjunto todo ao determinar as escalas, dificultando a avaliação da relação todo/parte (mesma escala usada para intervalos diferentes). Neste caso, são introduzidos ruídos que induzem a falsas interpretações, como discutido em [Garcia et al. \(2004c\)](#). Este fato assume maior importância quando consideramos a meta-análise exploratória, pois o mesmo artefato de software pode ser preparado com diferentes distribuições de defeitos.

Em geral, técnicas de visualização de baixa dimensionalidade foram escolhidas para investigar as hipóteses dos experimentos, pois estas relacionam um número baixo de variáveis. Entretanto, na Pré-Análise e na Análise Exploratória é importante considerar simultaneamente todo o conjunto de atributos. Como discutido em [Garcia et al. \(2004c\)](#): na Figura 4.9 observa-se que um estudo comparativo entre a influência da experiência do participante como *Analista* e *Testador*, e com o *Uso* ou *Escrita* de requisitos seria irrelevante, dada à alta correlação entre esses atributos no caso dos dados desta replicação em particular. A meta-análise dos dados coletados em diversas replicações pode sugerir que a coleta de tais informações pode ser simplificada, se confirmada a correlação.

Figura 4.8: Sumarização de defeitos reportados na replicação R1, por tipo e documento (cor identifica o tipo de defeito)

Figura 4.9: Representação por Coordenadas Paralelas das diferentes Experiências dos participantes da replicação R1

4.5.1.2 Arranjo dos Atributos

Para a visualização, não só a escolha dos atributos a serem mapeados é importante, mas também a escolha de seu mapeamento em marcas, substratos espaciais ou propriedades gráficas. A escolha deve considerar o tipo de variável a ser mapeada, por exemplo Categórico, Ordinal ou Numérico. Especificamente no caso dos dados de engenharia de software experimental, em que predominam atributos categóricos, o mapeamento em técnicas hierárquicas é facilitado. É necessário considerar problemas de mapeamento: por exemplo, Gráficos de Dispersão 3D apresentam limitações com o mapeamento de atributos para o tamanho de marcas gráficas, já que a projeção da representação 3D no dispositivo 2D pode distorcer o resultado visual e prejudicar a interpretação. Para algumas técnicas, como Coordenadas Paralelas, a sequência em que os eixos são apresentados é relevante, como discutido no Capítulo 3.

Na análise visual dos dados dos experimentos PBR, em uma análise confirmatória os atributos envolvidos na hipótese tratada são conhecidos, o que direciona os atributos a serem selecionados. Entretanto, a sequência de mapeamento – em Coordenadas Paralelas, por exemplo – é um ponto que tem sido tratado por diversos pesquisadores (Wegman, 1990, Ankerst et al., 1998, Pagel et al., 2000, Jr. et al., 2000, Artero e Oliveira, 2004). Por outro lado, em uma análise exploratória, todos os atributos são potencialmente interessantes, a não ser que se tenha

uma hipótese em formulação que permita delimitar um subconjunto.

No caso específico dos estudos conduzidos, propõe-se usar as próprias hipóteses para estabelecer arranjos interessantes do conjunto de atributos. Em Análise Confirmatória, a hipótese em verificação direciona a escolha dos atributos. Considerando a hipótese apresentada na Figura 4.10, que relaciona a experiência prévia do inspetor em desenvolvimento de software quando é aplicada uma técnica de leitura não-procedural a documentos de requisitos, à detecção de mais defeitos, a sugestão é dispor os atributos na sequência em que aparecem na hipótese ficando, em Coordenadas Paralelas, por exemplo, a variável dependente à extrema direita. Em apresentações de resultados intermediários em workshops do Projeto *Readers*, os especialistas sugeriram tal organização, alegando que é mais fácil identificar a hipótese em estudo.

<i>(SoftwareDevelopmentExperience)</i>	<i>e</i>
<i>(RequirementsInspection)</i>	<i>e</i>
<i>(Non – procedural technique ou ad hoc technique)</i>	$\Rightarrow$ <i>(Moredefectsfound)</i>

Figura 4.10: Hipótese 1 para Análise Confirmatória

A representação obtida com essa estratégia é apresentada na Figura 4.11: o primeiro eixo indica a técnica aplicada (uma única categoria foi filtrada, pois a hipótese considera apenas técnicas *ad-hoc*); a experiência em desenvolvimento no eixo seguinte; e no último a variável dependente (*defects found*). O tipo de inspeção – Inspeção de requisitos – não foi apresentado, pois todos os dados considerados são de Inspeção de Requisitos. Para uma representação simples, que mapeia apenas três atributos, poderia ser utilizada outra técnica de visualização. Entretanto, uma técnica multidimensional permite extrapolar a confirmação de uma hipótese, passando à exploração ao avaliar concomitantemente múltiplas hipóteses, e nesse caso pode-se tentar isolar fatores comuns entre hipóteses.

Observando a hipótese apresentada na Figura 4.12, que relaciona o conhecimento do domínio da aplicação pelo inspetor quando aplicada uma técnica não-procedural a documentos de requisitos, à detecção de mais defeitos, verifica-se que há variáveis independentes em comum com a hipótese apresentada anteriormente. Nesse caso, seria interessante exibir as variáveis em uma mesma visualização para cruzar as hipóteses e avaliar as influências de diferentes variáveis independentes conjuntamente. A idéia é que uma mesma representação visual possibilite explorar cenários criados a partir de múltiplas hipóteses, evidenciando o comportamento e a influência dessas variáveis, no caso, na detecção de defeitos. A estratégia adotada consiste em

Figura 4.11: Representação por Coordenadas Paralelas obtida pela sequência intuitiva (R2)

variar a sequência de exibição dos atributos de modo que, modificando a posição da variável dependente, seja possível comparar diretamente o efeito de cada uma das variáveis independentes.

<i>(DomainKnowledge)</i>	<i>e</i>
<i>(RequirementsInspection)</i>	<i>e</i>
<i>(Non – procedural technique ou ad hoc technique)</i>	$\Rightarrow$ <i>(Moredefectsfound)</i>

Figura 4.12: Hipótese 2 para Análise Confirmatória

Na Figura 4.13 é apresentado um instantâneo de uma representação por Coordenadas Paralelas que permite a análise conjunta das duas hipóteses. O atributo que representa o número de defeitos reportados (*Defects_Found*) encontra-se entre os atributos que representam o conhecimento do domínio (*K_ATM*) e a experiência do participante (*Experience*). Nessa representação, segmentos de reta foram perturbados (o *jitter* reduz a sobreposição e mostra as várias ocorrências sobrepostas). O primeiro eixo indica a técnica (ambas as hipóteses consideram apenas a aplicação de técnicas *ad-hoc*), e o segundo indica que apenas os registros de um dos documentos (ATM) estão projetados (a avaliação do conhecimento no domínio exige a análise separada por documento). Essa organização permite a análise de ambas as hipóteses comparando-se os atributos aos pares e trios.

O tratamento aos pares limita a comparação às duas variáveis independentes (*Conhecimento do Domínio* e *Experiência*, representadas por *K_ATM* e *Experience*, respectivamente) adjacentes à variável dependente (*Defeitos Reportados – Defects Found*). As demais variáveis, nesse caso, indicam o contexto da aplicação demonstrando a correta configuração da filtragem realizada. Caso haja interesse, pode-se explorar recursos de interação para destacar subgrupos selecionando intervalos de interesse. Admite-se também a possibilidade de mapear uma variável para a cor dos segmentos de reta, o que permite analisar a influência de três variáveis independentes sobre a variável dependente. Na Figura 4.14 é apresentada uma visualização cuja cor foi mapeada para o atributo *Domínio da Língua (English)*. Observa-se que, nesse conjunto de indivíduos, quem tem maior domínio da língua, em geral, também tem maior conhecimento sobre o domínio.

Figura 4.13: Representação por Coordenadas Paralelas das hipóteses 1 e 2 na mesma projeção

Em análise exploratória, quando não se tem um ponto de partida, a estratégia estabelecida para o mapeamento em Coordenadas Paralelas consiste em: (1) aproximar os atributos semanticamente semelhantes (alta correlação) e com o mesmo intervalo de variação; e (2) identificar a presença de linhas cruzadas e reposicionar os eixos tentando reduzir os cruzamentos. Em uma

Figura 4.14: Representação equivalente à Figura 4.13, com mais um eixo: *Domínio da Língua*

análise exploratória, a escolha dos atributos e de sua sequência é facilitada rotulando o intervalo de variação de cada atributo.

Na Figura 4.11 o mapeamento dos intervalos de variação de cada atributo nos eixos adotou uma normalização local, e é exibida uma legenda em percentual (0% a 100%) comum a todos os eixos – a legenda comum é uma característica da ferramenta utilizada e não permite a verificação dos valores reais assumidos por cada atributo. Por outro lado, a normalização local facilita a observação de correlações entre atributos, principalmente quando esses têm intervalos de variação com amplitudes muito diferentes. Nas Figuras 4.13 e 4.14, o mapeamento dos intervalos de variação de cada atributo nos eixos adotou uma normalização global, em que a legenda do eixo indica o intervalo de variação. Quando a análise envolve comparação quantitativa entre atributos com semântica similar (por exemplo, níveis de diferentes tipos de experiência prévia) a normalização global é mais conveniente. Já a verificação de cruzamentos (passo 2), exige que o volume de dados seja pequeno – uma vez que, caso contrário, é impossível identificá-los visualmente. Uma representação por Coordenadas Paralelas obtida com uma sequência aleatória dos atributos é apresentada na Figura 4.15. Depois de algumas interações executando os passos

(1) e (2) foi estabelecida uma sequência de apresentação que permite a comparação direta entre os atributos associados, apresentada na Figura 4.16.

Figura 4.15: Representação por Coordenadas Paralelas com sequência aleatória de atributos

Figura 4.16: Representação por Coordenadas Paralelas depois de estabelecida uma sequência pela análise visual das correlações e com normalização global

Para validar a estratégia aplicada para obter a visualização da Figura 4.16, os mesmos dados foram submetidos à ferramenta desenvolvida por [Artero e Oliveira \(2004\)](#) e a seguinte sequência de exibição de atributos foi sugerida: *Developer*, *Writing_Req*, *Using_Req*, *Analyst*, *Tester*, *Manager*. Pode-se observar a equivalência entre ambas as sequências. Em resposta à solici-

tação de redução de dimensionalidade, a ferramenta de Artero e Oliveira sugeriu a eliminação do atributo *Using_Req*, devido à alta similaridade com *Writing_Req*, como já havia sido observado. Um instantâneo da ferramenta desenvolvida por Artero e Oliveira é apresentado na Figura 4.17, mostrando os atributos *Developer*, *Writing_Req*, *Analyst*, *Tester*, *Manager*, nesta ordem. Nessa figura, observa-se na representação por *Parallel Coordinate Density Plot* – que mapeia a quantidade de pares de elementos em densidade – que o maior número de defeitos foi reportado por participantes com baixa experiência em teste e gerência.

Figura 4.17: Representação por Coordenadas Paralelas obtida com a ferramenta de Artero e Oliveira (2004)

A dimensionalidade não é um aspecto particularmente crítico para os estudos em questão, já que muitas análises envolvem um número pequeno de atributos. Entretanto, a possibilidade de reduzir a dimensionalidade dos dados pode ser relevante para a condução dos experimentos e replicações, e as representações visuais podem apoiar descobertas interessantes nesse sentido. Por exemplo, é possível observar na Figura 4.16 uma alta correlação entre os atributos *Using Requirements Documents* e *Writing Requirements Documents*, indicando que talvez a

análise em separado da influência de cada uma dessas medidas possa ser descartada. A mesma visualização permite observar que, nesses dados, os níveis de experiência são baixos, exceto com relação à experiência com *Desenvolvimento*. Isso sugere que futuros experimentos devem ser conduzidos com indivíduos mais experientes nessa medida, para aumentar a cobertura dos resultados.

4.5.1.3 Definição das Propriedades Gráficas

Para criar a visualização, tipicamente cada atributo dos dados multidimensionais é mapeado em atributos visuais de alguma entidade gráfica. Para isso, o tipo do dado (i.e., discreto, contínuo) ou a escala de valores podem ser alterados para evidenciar uma informação útil. A interpretação de uma representação visual requer dois estágios de percepção: (1) quais são os elementos mapeados e (2) quais as relações entre esses elementos ([Bertin, 1981](#)). Sem a identificação dos elementos mapeados uma representação torna-se inútil, e por outro lado, quando o mapeamento favorece a identificação dos elementos mapeados, facilita a interpretação.

Considerando que, em geral, é útil utilizar múltiplas técnicas de visualização para explorar seus aspectos complementares, ou aplicar simultaneamente uma mesma técnica a diferentes subconjuntos dos dados, foram introduzidas no processo apresentado na Figura 4.5 as setas coloridas (verde, azul e vermelho), cujas cores e tracejados representam mapeamentos correspondentes e complementares ocorrendo em paralelo. Espera-se que a correspondência entre os mapeamentos facilite pelo menos a primeira etapa da interpretação. A correspondência tem por objetivo facilitar não só a interpretação, mas também a geração das representações visuais, pois um bom mapeamento pode servir de parâmetro para o seu complementar. Com duas ou mais representações visuais, observou-se que mapeamentos correspondentes facilitam a interpretação dos resultados, pois indicam que duas representações que mapeiam o mesmo atributo o fazem de modo consistente, como nas Figuras 4.7 e 4.8, nas quais a cor identifica o tipo de defeito. Deve-se evitar que múltiplos atributos, com características diferentes, sejam mapeados para a mesma propriedade gráfica percebida, o que pode induzir falsas interpretações. O tipo e a escala dos atributos devem ser considerados na escolha do mapeamento, pois certas propriedades gráficas não são adequadas para mapear certos tipos de dados ([Bertin, 1981](#), [Ware, 2000](#), [Chen, 2004](#)).

Além disso, indivíduos podem exercer diferentes papéis no processo de experimentação e

variações no mapeamento podem dificultar a interpretação. O mapeamento adequado deve ser transmitido ao compor o pacote de laboratório, de modo a permitir não só sua replicação, mas também facilitar a transferência de conhecimento para outros replicadores. As técnicas interativas permitem a definição e uso de mapeamentos variados. Além disso, para obter um “boa” visualização – que evidencie a informação necessária para realizar a tarefa de análise – é necessário executar um ciclo de *mapeamentos visuais e transformações de visões*, apresentado na Figura 4.5 pela junção identificada como *Representação Visual*. De modo geral, o procedimento para obter um mapeamento visual consiste em:

1. **Definição:** tendo escolhido uma técnica, define-se um mapeamento considerado adequado aos objetivos da tarefa e da representação (apresentação, análise confirmatória ou exploratória);
2. **Validação:** observa-se o mapeamento utilizado na tentativa de validar o significado semântico;
3. **Fixação:** registro dos mapeamentos e transformações para facilitar a interpretação e reúso do resultado.

Na busca por representações visuais efetivas, diversos mapeamentos podem ser experimentados na fase de *Definição*. Na fase de *Validação* deve-se analisar a representação visual não apenas com o intuito de encontrar a informação almejada – no caso de uma busca direcionada –, mas também para avaliar a facilidade de interpretação. No caso de uma busca não direcionada, quando encontrada uma representação que evidencie algum fato, recomenda-se que outras alternativas sejam geradas para evitar conclusões precipitadas: um mapeamento inadequado pode levar a falsas interpretações.

Além desses, um quarto passo é admitido na situação em que o mapeamento visual tenha sido estabelecido por terceiros, em substituição à *Definição*. O *Aprendizado* consiste em observar o mapeamento existente na tentativa de identificar os elementos utilizados e o seu significado semântico.

A definição de um mapeamento visual não pode ser dissociada das *Transformações de Visões* e nem da análise: como pode ser observado, a Validação implica em interpretar a visualização. Portanto, a análise dos resultados ocorre, em um primeiro momento, na fase de

Validação do mapeamento, principalmente se a análise está sendo conduzida pelo *projetista* (que está criando mapeamentos). Este, na condução de meta-análises, pode usar mapeamentos pré-definidos. Nesse contexto, três atividades formam o ciclo de análise: *Mapeamento*, *Busca* e *Reconhecimento*. A tarefa *Mapeamento*, neste caso, resume-se a aplicar o mapeamento como um *template*. A segunda – a *Busca* – implica em interpretar a visualização na procura por padrões, definidos pelo objetivo da análise. Por se tratar de um processo altamente interativo e também iterativo, mesmo tendo sido validado o mapeamento (passo 2), adequações podem ser necessárias e, portanto, o ciclo completo de mapeamento deve ser conduzido. No último passo – o *Reconhecimento* – admite-se a possibilidade de se verificar ou não um resultado, já que o padrão procurado pode não estar presente nos dados. O mais importante é registrar o fato observado. O que pode ser facilmente efetuado em caso de reconhecimento positivo, pode ser trabalhoso quando do não reconhecimento do padrão, uma vez que as diversas tentativas de mapeamento devem ser registradas, pois servem de base para a confiabilidade da observação.

4.5.2 Transformações de Visões

As atividades desta etapa são, em geral, as mesmas propostas por Card et al.: *exploração local*, *controle de ponto de visão* e *distorção*. *Exploração local* são transformações de visões que usam a localização na estrutura visual para revelar informações adicionais. *Controle de ponto de visão* são transformações do tipo *zoom* e *clipping* usadas para evidenciar detalhes da visualização. *Distorção* consiste em modificar estruturas visuais para criar visões de foco e contexto simultaneamente (visão geral e detalhes, na mesma representação). Entretanto, considerando o contexto de aplicação, destaca-se a importância das operações interativas de seleção e ligação. Por meio de seleção pode-se localizar um subconjunto de registros apontando diretamente os elementos gráficos, ou indiretamente por meio de controles auxiliares. A maneira de selecionar os elementos depende da ferramenta de visualização. A ligação é uma forma de coordenação – permite que objetos selecionados em uma visualização sejam também destacados em outra. Com isso é possível explorar aspectos complementares entre as técnicas, já que características não evidenciadas por uma determinada técnica de visualização podem ser identificadas em outras, como discutido na Seção 3.5.

4.6 Arcabouço para Visualização

Com base no Processo ViDA_{ESE}, foi proposto um arcabouço com o intuito de estabelecer diretrizes para o desenvolvimento de ferramentas de apoio ao processo. O arcabouço, apresentado na Figura 4.18, foi subdividido em dois grandes blocos que contemplam as atividades em ambos os espaços: Espaço de Dados e Espaço Visual. Nessa figura as setas pontilhadas representam intervenções do analista, seja definindo parâmetros ou interagindo com representações visuais, por exemplo. Já as setas contínuas representam a passagem de dados propriamente dita, como parâmetros reais, sendo azul para dados e vermelho para meta-dados.

Figura 4.18: Arcabouço de Visualização

A primeira atividade – no espaço de dados – tem início com a interação do analista com a base de dados (*Dados Brutos*), representada pela seta pontilhada que liga o analista à base de dados para a seleção de dados. Os dados selecionados pelo analista são submetidos a um tratamento que busca adequá-los às técnicas/ferramentas que serão utilizadas na análise. Nesta atividade, o analista deve fornecer parâmetros para a transformação e estruturação dos dados, conforme estabelecido no processo. Como resultado, são gerados os *Dados Selecionados* (conforme definido no processo) separados em *Dados* e *Meta-Dados*, que contém os dados conver-

tidos, se necessário, e sua descrição, respectivamente. A preparação de dados e seu registro como meta-dados tem por objetivo descrever a organização dos dados a serem visualizados, seja definindo relações semânticas ou agrupamentos (definidos pelo analista). Para ilustrar a instanciamento do arcabouço, uma ferramenta foi desenvolvida para apoiar as atividades básicas dessa etapa. Na Figura 4.19 é apresentado um instantâneo da ferramenta com a configuração parcial de um mapeamento, com parâmetros para a operação. É possível gravar e recuperar a configuração para aplicá-la a outros conjuntos.

Figura 4.19: Instantâneo da ferramenta de suporte à seleção e à estruturação de dados

Tendo sido gerados os arquivos formatados, as próximas atividades do processo consistem nas atividades no Espaço Visual que são dependentes das técnicas escolhidas. A ferramenta que apóia a aplicação da técnica também influencia a criação de visualizações, pois pode ser necessário definir parâmetros específicos para obter representações visuais significativas. Considerando que é preciso repetir “visualizações”, foi considerada a necessidade de oferecer artefatos de apoio ao analista, de modo a facilitar seu trabalho, mas sem restringir a liberdade necessária à exploração de dados em buscas não direcionadas. Para tanto, propõe-se um esquema para manter *templates* de visualizações, tornando possível a aplicação de *templates* previamente definidos aos dados em análise. A estrutura do *template*, definida em XML (Extensible Markup Language), deve representar os parâmetros necessários para gerar uma visualização e, portanto,

os parâmetros das ferramentas que provêm suporte às técnicas utilizadas. A estrutura do *template* deve representar:

- a identificação da ferramenta/técnica;
- a descrição da visualização (com intuito apenas documental);
- os atributos representados;
- os parâmetros da ferramenta e seus valores (que definem as configurações da ferramenta);
- e os critérios de seleção de registros, se necessário.

Tais elementos podem ser observados na estrutura do *template* apresentada na Figura 4.20. Os dois primeiros atributos têm por objetivo validar o *template* no ato de sua aplicação. Quanto à descrição da visualização, na estrutura foram incluídas: o autor, o conjunto de dados, o objetivo da visualização, o que foi observado, e comentários sobre como analisar a representação. Os atributos representados consistem em uma lista dos atributos utilizados na representação visual e uma descrição do mapeamento visual a ser utilizado (atributo visual e o respectivo parâmetro, por exemplo, mapa de cores). Os parâmetros da ferramenta descrevem como ela deve ser configurada para obter não só a representação visual, mas também ângulos de visão, níveis de transparência, legendas, dentre outros. Entretanto, essa lista de parâmetros é sensível ao par técnica/ferramenta, e por isso, pode ser necessário definir especificidades para que a estrutura seja genérica o bastante para suportar várias técnicas/ferramentas. Quanto ao critério de seleção de registros, deve-se armazenar os atributos, a relação usada no critério de seleção (maior, menor, igual, e combinações) e os respectivos valores. Inicialmente foi considerado que a representação de registros selecionados é feita por cor, representada pelos seus valores no padrão RGB.

Tomando como estudo de caso uma ferramenta desenvolvida como parte deste trabalho, que implementa a técnica Coordenadas Paralelas com o diferencial de permitir a seleção interativa de elementos e a representação de frequência de segmentos (pares de valores de atributos vizinhos) pelo grau de transparência, foram implementados módulos de gravação e leitura de arquivos XML. Na Figura 4.21 é apresentado um arquivo XML omitindo-se a lista completa de atributos. Nela, os atributos armazenados não possuem detalhamento de critério de representação, uma vez que a técnica Coordenadas Paralelas não requer esse tipo de dado. A ausência da

Figura 4.20: Estrutura do *template* proposto em XML ^a

^arepresentado usando a ferramenta XMLSpy (disponível a partir de www.altova.com)

```

<?xml version="1.0" encoding="UTF-8"?>
<Visualization>
<!--Este arquivo foi gerado pela ferramenta ViDA-ESE-->
<Tool>ViDA</Tool>
<Technique>CP</Technique>
<VisualizationDescription Author="Rogerio"
    Objective="Correlacao entreCodigo & Teste"
    Comment="Observe os atributos à direita"
    DataSet="C:\User\Rogerio\USP\Doutorado\Analises\
    Profile_Fall98_01.okc"/>
<Attribute>Replication</Attribute>
<Attribute>Subject</Attribute>
...
<Attribute>testing_based_on_requirements_use_cases</Attribute>
<Parameter Value="false" ID="0" Description="cbSelectFirst"/>
<Parameter Value="false" ID="1" Description="cbShowSelection"/>
<Parameter Value="false" ID="2" Description="cbShowFile"/>
<Parameter Value="true" ID="3" Description="cbTransparencia"/>
<Parameter Value="false" ID="4" Description="cbGlobalNormalization"/>
<Parameter Value="1145" ID="5" Description="slTransparencia"/>
<Parameter Value="false" ID="6" Description="cbShowCriteria"/>
</Visualization>

```

Figura 4.21: Parte de um *template* gerado pela ferramenta implementada

seleção de registros na representação visual armazenada pode ser também observada. Dentre os parâmetros armazenados, observa-se que a visualização utilizou o controle de opacidade dos segmentos de reta, já que o controle *cbTranparencia* foi gravado como *true* e o valor do controle da transparência (*slTransparencia*) foi gravado com *1145* (a ferramenta adota uma escala de 1 a 10000 para representar a transparência de zero a 100%, portanto, 1145 significa 11,45% de opacidade). Na Figura 4.22 é apresentado um instantâneo da ferramenta depois de aplicar o *template* apresentado na Figura 4.21. Em sua interface observa-se que:

- os dados utilizados nessa visualização são os dados dos experimentos Fall98, Fall01 e Fall02, como identificado pelo nome do arquivo de origem dos dados;
- o *template*, armazenado no arquivo *teste.xml*, tem como autor *Rogerio* e foi obtido durante as análises dos experimentos Fall98 e Fall01 (*Data Set: Profile_Fall98_01*). Além disso, tem como objetivo a análise da correlação entre as experiências em codificação e teste;
- como indicado no comentário do *template*, observa-se que há uma certa correlação entre os atributos *Coding based on requirements/use-case*, *Testing Software* e *Testing based on*

requirements/use-case. Mesmo sem o recurso da variação interativa do grau de opacidade, pode-se observar na Figura 4.22 que há diferenças na opacidade (visualmente reconhecida pela variação de níveis de cinza) de vários elementos de dados (poli-linhas).

Figura 4.22: Visualização obtida com a aplicação de um *template* gerado previamente

O arcabouço proposto facilita a intensa interação que ocorre em um processo de análise visual, e conseqüentemente as várias iterações têm suas execuções facilitadas. Esse benefício é fundamental, uma vez que as análises conduzidas devem ser repetidas (requisito da Experimentação). As análises conduzidas na Mississippi State University permitiram observar que a utilização de um conjunto de ferramentas que suportam o arcabouço proposto é fundamental para a condução das análises. As análises realizadas exigiram a observação de fatos não só nos estudos isolados, mas também nos ciclos de meta-análise, repetidos para as diversas combinações de dados dos experimentos. Vale ressaltar que a estrutura proposta para o *template* tem atendido às necessidades observadas até então, mas é restrita à técnica Coordenadas Paralelas. É necessário estendê-lo para suportar outras técnicas.

4.7 Considerações Finais

O uso de recursos de visualização para apoiar tarefas de análise de dados de estudos experimentais traz benefícios para várias perspectivas e papéis do processo de experimentação. Exemplos desses benefícios foram identificados ao longo deste capítulo, tais como a possibilidade de detectar antecipadamente a correlação entre atributos, eliminar a necessidade de análises mais detalhadas, e direcionar futuros experimentos, permitindo gerar dados mais representativos para futuras meta-análises. A visualização também pode apoiar a condução de meta-análise exploratória sobre dados agregados oriundos de múltiplos experimentos, que é essencial para a formação do corpo de conhecimento sobre o tema em estudo.

Entretanto, as características dos dados e os tipos das análises conduzidas demandam uma adequação do processo de visualização, como descrito na Seção 4.2, incorporando novas atividades e tarefas específicas. Como mencionado, a necessidade de reutilizar o processo de análise demanda atividades adicionais, que foram exemplificadas na tentativa de explicitar como as características das representações visuais podem influenciar as atividades envolvidas posteriormente. Entretanto, vale destacar que não é objetivo deste trabalho estabelecer um guia que determina efetivamente as atividades a serem efetuadas, o que poderia tolher a análise exploratória. É importante dar diretrizes, mas também dar liberdade de escolha. Por isso, evitou-se determinar quais técnicas devem ser aplicadas a cada atividade de análise, pois isso restringiria o escopo de possíveis aplicações. As diretrizes identificadas permitem ao analista escolher entre uma família de técnicas que possibilitam as análises necessárias. Observou-se, entretanto, que um conjunto de diretrizes específicas deve ser considerado dependendo da técnica adotada.

Depois de aplicado a diversos conjuntos de dados das replicações conduzidas no escopo do Projeto *Readers*, o processo foi instanciado em outros contextos – dados de outras replicações conduzidas pelos parceiros norte-americanos. Tal instanciação foi importante para o seu amadurecimento. Nessa experiência, foi possível aplicar a visualização como apoio à meta-análise exploratória, em um volume de dados maior. Adicionalmente, durante essa etapa foi avaliado o arcabouço proposto. Embora a sua aplicação tenha sido possível apenas com a ferramenta desenvolvida (que suporta a técnica Coordenadas Paralelas), foi possível observar que ele aumenta a aplicabilidade do processo por apoiar atividades repetitivas, sem tirar a liberdade necessária à exploração não direcionada.

Por fim, vale ressaltar que as extensões propostas tornam o processo mais adequado não apenas à análise de dados de estudos experimentais em Engenharia de Software, mas, genericamente, à análise de dados oriundos de fontes diferentes, mas que mantêm a mesma estrutura – característica observada nos estudos dessa natureza.

No próximo capítulo são apresentadas diversas análises demonstrando a instanciação do processo aos dados coletados nos experimentos PBR, juntamente com as conclusões obtidas.

Aplicação do Processo ViDA_{ESE}:

Resultados

O objetivo deste capítulo é apresentar a aplicação do processo proposto com o intuito de exemplificar análises conduzidas, e apresentar as principais conclusões. Foram executados diversos ciclos de análise, tanto para análise de dados isolados de experimentos individuais, quanto para análise de dados agregados oriundos de múltiplos experimentos (ou replicações). Alguns deles foram selecionados para ilustrar a aplicação do processo. Os critérios para a seleção foram o objetivo da análise, a abordagem, a técnica de visualização proposta e o papel do analista (replicador ou projetista do experimento – ver Figura 4.1). Análises sob o papel *participante* não foram apresentadas em vista da visão limitada dos indivíduos neste papel. Dados agregados de múltiplos experimentos permitem exemplificar a aplicação do processo, e apresentar a abordagem proposta para a meta-análise exploratória. Vale ressaltar que a apresentação de ciclos análise de dados de múltiplos experimentos engloba a análise de experimentos isolados, já que o primeiro passo da abordagem proposta é a análise isolada dos dados do experimento em foco (ver Figura 4.4).

Adicionalmente, foi proposta uma abordagem de simulação de dados de estudos experimentais, sob a perspectiva da taxonomia de estudos experimentais discutidas na Seção 2.1. A condução de estudos experimentais não é trivial e os custos associados são expressivos, o

que motivou uma abordagem baseada em Algoritmos Genéticos para a geração de dados hipotéticos. Dados gerados com tal abordagem são analisados visualmente, com a aplicação do processo proposto. O capítulo encontra-se organizado em três seções, como segue:

- Na Seção 5.1 são apresentadas as tarefas das etapas definidas para o espaço de dados e o resultado obtido.
- Na Seção 5.2 são apresentados ciclos de análise – com atividades no espaço de dados e no espaço visual – que exemplificam diferentes categorias de técnicas visuais aplicadas em função de diferentes objetivos de análise;
- Na Seção 5.3 é exposta a abordagem baseada em Algoritmos Genéticos para a geração de dados hipotéticos, o que permite a prospecção de novos cenários a partir de dados coletados em experimentos realizados.

5.1 Espaço de Dados: Etapas Gerais

Nesta seção são apresentadas as etapas consideradas gerais, ou seja, que são realizadas independentemente do objetivo da análise. As atividades no Espaço de Dados que são dependentes do objetivo da análise são discutidas juntamente com as atividades definidas no Espaço Visual, na Seção 5.2.

Para uma meta-análise exploratória seguindo a abordagem proposta, os dados coletados em cada replicação foram analisados separadamente para identificar possíveis problemas de inconsistência, como discutido no Capítulo 4. Em seguida, foi conduzida a análise de meta-dados, que é o primeiro passo definido no processo. Para exemplificar a instanciação do processo, foi considerada como foco a integração dos perfis dos participantes, coletados por diferentes questionários devido à evolução do pacote de laboratório ao longo das replicações. Na Tabela 5.1 é apresentado um sumário de observações registradas durante o processo de integração dos dados, seguindo a abordagem de agregar um conjunto de dados de cada vez, incrementalmente. A tabela não apresenta todos os atributos, apenas um subconjunto deles cujos nomes foram mantidos como aparecem no Pacote de Laboratório (não foram traduzidos). Nela, os círculos indicam que os valores do atributo em particular foram coletados em unidades de tempo (meses/anos de experiência), marcas de checagem indicam valores coletados em categorias pré-definidas. Os

triângulos representam valores coletados em ambos os formatos (mas poucos participantes responderam ambos). Observou-se que nos primeiros experimentos os diferentes atributos para registro da experiência dos participantes – como *Manager*, *Developer*, *Tester* e *Analyst* – foram coletados em meses, e em experimentos posteriores como uma escala de valores (1, 2, 3, 4 ou 5 indicando experiência crescente). O atributo *Experience in Test*, presente no Pacote de Laboratório das primeiras replicações, foi expandido para *testing software*, *equivalence-partition testing*, *testing based on requirements/use-cases* em replicações posteriores. O atributo *Design of System* foi coletado no experimento original idealizado e conduzido por Basili et al. (1996c), denominado NASA, somente em 1995 (dados do estudo piloto conduzido em 1994 não foram analisados). Os questionários para os experimentos NASA, R1 e R2 incluíram o atributo *Experience as Analyst*, que foi mapeado para *Design of System from requirements/use-case*, como observado na tabela. As considerações feitas na integração dos dados influenciam análises futuras, devendo ser registradas. Esta atividade – análise de meta-dados – pode ser apoiada por técnicas de Visualização de Informação.

Figura 5.1: Modelo da base de dados para análise que mantém dados de todas as replicações

O segundo passo no processo de meta-análise exploratória tem como foco o pacote de laboratório, e busca identificar restrições derivadas dos objetivos do estudo e, portanto, do projeto experimental, que podem afetar a integração dos dados. Por exemplo, Carver (2003) conduziu um experimento para identificar a influência da experiência prévia em aplicação de PBR no

Tabela 5.1: Lista parcial de atributos relativos à experiência de participantes

Atributos	Replicações Analisadas								
	NASA	Fall97	Fall98	R1	R2	R3	R4	Fall01	Fall02
English skills					✓	✓	✓		
English as First Language								✓	✓
English reading comprehension skills						✓	✓	✓	✓
English listening and speaking skills						✓	✓	✓	✓
English writing skills						✓	✓		
Software development in practice	•			•	•	✓	✓	✓	✓
Software management	•			•	•	✓	✓	✓	✓
User Interface (UI) design						✓	✓		✓
Software inspections						✓	✓	✓	✓
Writing requirements	•		✓	•	•	✓	✓	✓	✓
Writing Use Cases			✓			✓	✓	✓	✓
Reviewing Requirements	•	✓	✓	•	•	✓	✓	✓	✓
Reviewing Use Cases		✓	✓			✓	✓	✓	✓
Changing requirements for maintenance			✓			✓	✓	✓	✓
Design of systems	•	✓	✓	•	•	✓	▲		✓
Changing designs for maintenance			✓			✓	▲		✓
Creating Object-Oriented (OO) designs						✓	▲		✓
Reading OO designs		✓				✓	▲		✓
Unified Modelling Language (UML)						✓	✓		✓
Structured design		✓				✓	✓		✓
Design of systems from requirements/use-case	•		✓	•	•	✓	✓		✓
Design of system from use case							✓		
Maintenance of code			✓					✓	✓
Coding based on design								✓	✓
Coding based on OO design			✓					✓	✓
Coding based on requirements/use-case			✓					✓	✓
Testing software	•		✓	•	•	✓	✓	✓	✓
Equivalence-partition testing		✓				✓	✓	✓	✓
Testing based on requirements/use-cases		✓	✓			✓	✓	✓	✓
Experience (categories “High” or “Low”)									✓
Unit Testing									✓
Integration Testing									✓
System Testing									✓
Acceptance Testing									✓

desempenho de participantes. O projeto experimental original foi alterado com a introdução de um grupo de participantes – chamados de “observadores” – que aplicou PBR depois de observar a sua aplicação por outro grupo. Neste caso, os dados de participantes do grupo de observadores não podem ser combinados diretamente com dados de outros grupos, pois receberam um treinamento diferente – a observação como uma etapa adicional de treinamento. Entretanto, dados de grupos de participantes submetidos a treinamento similar podem ser integrados. Neste passo, as restrições identificadas definem quais registros podem ser combinados.

O próximo passo do processo, no espaço de dados (ver Figura 4.5), consiste em organizar

os dados em tabelas agrupando atributos relativos a diferentes fontes (documentos), seja em arquivo ou em uma base de dados (*Dados Tabelados*). Essa etapa é necessária porque os dados são coletados em formulários diversos e originar registros integrados em uma fonte única e uniforme para o processo de visualização. Deve-se definir uma instância de interesse e agregar os atributos disponíveis relativos a essa instância. Por exemplo, considerando que as instâncias de interesse são os defeitos reportados, deve-se associar os atributos que descrevem o participante que reportou, os atributos relativos ao documento analisado e à técnica utilizada. Tal etapa é crítica, especialmente diante das atuais condições disponíveis para a realização dos estudos experimentais, com baixo nível de automação em que os dados coletados em formulários são mantidos em planilhas. Por exemplo, no instantâneo apresentado na Figura 4.6 observa-se os defeitos reportados, apresentando apenas aqueles classificados como defeitos reais. Tal informação precisa ser relacionada aos dados que definem o perfil do participante quanto a sua experiência, disponível em outra planilha, para possibilitar análises da influência do *background* do participante em sua efetividade. Similarmente, o mesmo procedimento foi aplicado para documentos e defeitos, e como resultado foi gerada uma base de dados única que mantém os dados dos experimentos, cujo modelo é apresentado na Figura 5.1. Observa-se nessa figura que defeitos e defeitos reportados por participantes são mantidos separadamente, sem identificar explicitamente os participantes, que devem permanecer no anonimato. Além disso, uma tabela foi criada para manter os defeitos reportados em duplicidade (reportados mais de uma vez pelo mesmo participantes, em dada replicação). Juntamente com os atributos de interesse identificados no projeto do experimento, medidas estatísticas simples foram inseridas na base de dados, conforme estabelecido por Card et al. (1999), tais como total de defeitos reportados e valores médios de experiência dos participantes. Com o desenvolvimento de uma ferramenta de apoio à realização de estudos dessa natureza (Maltez Junior, 2006) os dados serão *naturalmente* mantidos em uma base de dados (SGBD), e não mais em planilhas, simplificando a atividade de agrupar atributos segundo a instância de interesse.

As demais atividades definidas no processo para serem realizadas no Espaço de Dados – *Interação em Dados Tabelados* e *Organização dos Dados* – devem ser conduzidas à luz dos objetivos da análise em questão. Dessa forma, serão apresentadas juntamente com as diversas representações visuais para apoiar a análise, na próxima seção.

5.2 Espaço Visual

Para exemplificar a aplicação do processo, são apresentados nesta seção alguns ciclos de análise e resultados obtidos pela aplicação de diferentes técnicas de visualização, com diferentes objetivos de análise. Dessa forma, nas subseções subsequentes são apresentados casos específicos (com objetivos de análise definidos) que norteiam a instanciação do processo.

5.2.1 Análise da Experiência dos Participantes

O primeiro ciclo de análise tem como foco investigar o perfil dos participantes dos experimentos. De acordo com a abordagem proposta para a condução de meta-análise exploratória (exposta na Figura 4.4), as representações visuais criadas comparando as experiências dos participantes nos diversos experimentos foram obtidas utilizando dados de experimentos isolados comparados aos demais. As possibilidades de agregação são determinadas pelas características dos dados sobre as replicações – evidenciadas no mapeamento dos atributos expostos na Tabela 5.1. Por exemplo, há diversas combinações possíveis, como *Fall98*, *Fall01*, *Fall02*, *R3* e *R4* para análise de experiência em desenvolvimento de software e com documento de requisitos (leitura, escrita e revisão). Observa-se que, dentre esses, em *Fall98* não foram coletados dados sobre domínio da língua, o que limita a avaliação dessa característica. Nas replicações em que foram coletados dados sobre domínio da língua, houve a adequação destes: coletados cadeias de caracteres, foram mapeados para números que refletem o grau de domínio, o que permite a projeção juntamente com outros atributos, mantendo suas relações semânticas. Outras replicações, como *R1* e *R2*, poderiam ser agregadas, mas a unidade de medida dos dados coletados foi alterada, o que dificulta a comparação direta.

Os conjuntos de dados de cada um desses estudos foram submetidos à análise isolada. Inicialmente, tomando-se como exemplo o estudo *Fall98*, uma representação inicial foi gerada, apresentada na Figura 5.2(a). Esta versão inicial pode ser melhorada aplicando-se a heurística definida para o problema da seqüência de atributos: por exemplo, observa-se o alto número de cruzamentos das poli-linhas, em especial ao redor dos atributos *maintenance_code* e *reviewing_requirements*. Na Figura 5.2(b) observa-se o resultado obtido após a aplicação da heurística em algumas interações: há menos “cruzamentos” entre poli-linhas, facilitando a comparação direta entre atributos correlacionados. Além disso, para evidenciar a sobreposição dos

segmentos, a ferramenta mapeia em tons variados a frequência de ocorrências de pares de valores de atributos vizinhos: em ambas as representações, segmentos mais escuros representam maior número de pares de valores entre os atributos vizinhos (analogamente, segmentos claros representam baixo número de ocorrências de pares de valores entre atributos vizinhos). Na Figura 5.2(a) destaca-se: alta correlação entre atributos que indicam baixa experiência, como *writing use case* e *reviewing use case*; alta correlação entre *reviewing requirements* e *changing requirements for maintenance* e *design systems from requirements* e *coding based on OO design*, com ocorrências de valores medianos.

Depois das análises isoladas, os dados foram agregados e novos ciclos de análises conduzidos. Na Figura 5.3(a) são apresentadas visualizações dos dados relativos ao perfil de participantes dos experimentos *Fall98* e *Fall01*, cujas escalas foram alteradas devido a valores ausentes em alguns atributos, mapeados em um valor negativo (-1). Em *a* observa-se a representação inicial, na qual os atributos ausentes foram selecionados – apresentados em vermelho – mostrando que todos são oriundos de *Fall01*: o primeiro eixo à esquerda mapeia o experimento, sendo *Fall98* e *Fall01* representados por 1 e 2, respectivamente. Após a aplicação do *template* obtido durante o mapeamento visual isolado de *Fall98* (apresentado na Figura 5.3(b)), os atributos com dados ausentes foram suprimidos. Para ilustrar quantitativamente o que se observa nas Figuras 5.2 e 5.3, a correlação entre os atributos *Reviewing Use Cases* e *Writing Use Cases* obtida numericamente pela equação 5.1 foi de 0,717224 e 0,855075, respectivamente, para dados de *Fall98* e dados combinados de *Fall98* e *Fall01*.

$$\rho_{x,y} = \frac{Cov(X, Y)}{\sigma_x \sigma_y} \quad (5.1)$$

$$Cov(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y) \quad (5.2)$$

Foram conduzidos diversos ciclos de análise, para analisar diversas combinações de agregação de dados dos vários experimentos. Por meio da interação com as representações obtidas, observou-se que:

- combinando *Fall98* e *Fall01*:

- Participantes com baixa experiência (≤ 3) em *Writing Requirements* têm também

(a) Representação inicial obtida usando a sequência de atributos presente no conjunto de dados

(b) Representação obtida após a aplicação da heurística proposta

Figura 5.2: Visualizações por Coordenadas Paralelas de atributos relativos à experiência do conjunto de dados de *Fall98*

(a) Representação inicial obtida usando a sequência de atributos presente no conjunto de dados

(b) Representação obtida depois da aplicação da heurística proposta e da redução de atributos

Figura 5.3: Visualizações por Coordenadas Paralelas de atributos relativos à experiência do conjunto de dados de *Fall98* e *Fall01*

baixa experiência em *Reviewing Requirements*, *Writing Use Case* e *Reviewing Use Case*;

- Participantes com baixa experiência (≤ 3) em *Testing Software* e baixa experiência em *Testing Based on Requirements* têm também baixa experiência em *Writing Use Case* e *Reviewing Use Case*, – com uma única exceção;
- combinando *Fall01* e *Fall02*:
 - Participantes com alta experiência (> 3) em *Software Management* têm também alta experiência em *Testing Software* e *Coding Based on Requirements*, com uma única exceção;
 - Participantes com alta experiência (> 3) em *Software Management* têm alta experiência em *Software Management* e *Reviewing Requirements*, com duas exceções;
- combinando *Fall98*, *Fall01* e *Fall02*:
 - Participantes com baixa experiência (≤ 3) em *Testing* têm baixa experiência em *Coding*;

Tais observações motivam uma discussão sobre revisão das hipóteses e do projeto dos experimentos. Pode-se tentar identificar e avaliar interferências entre as variáveis independentes nas conclusões estatísticas, já que não há como isolar os atributos altamente correlacionados. Tais fatos podem também subsidiar a evolução do pacote de laboratório, permitindo o aprimoramento do questionário de identificação do perfil dos participantes de futuros experimentos e/ou replicações.

5.2.2 Análise de Defeitos por Técnica/Documento

Outro ciclo de análise foi conduzido para explorar relações não antecipadas sobre os defeitos reportados. Para a análise dos defeitos deve-se considerar que os diferentes documentos utilizados possuem diferentes quantidades de defeitos semeados. Neste caso, a utilização de diferentes espaços de projeção para cada documento facilita a interpretação dos mapeamentos visuais (a projeção conjunta introduz ruídos que podem dificultar a interpretação, pois não permite uma comparação direta entre as proporções de todo/parte – ver Figura 4.7). Uma alternativa seria analisar separadamente cada conjunto de dados relativos a defeitos dos documentos. Entretanto,

pode-se contornar tal problema com técnicas de visualiação hierárquicas. A técnica Treemap, por lidar com dados numéricos e caracteres, não exige qualquer adequação no conjunto de dados. Para apoiar a definição da hierarquia foi considerado o projeto experimental, exposto na Figura 2.10. Dessa forma, a hierarquia para o mapeamento foi definida como *Documento*, *Técnica* e *Defeitos*, buscando observar a efetividade obtida pela aplicação de cada técnica a cada documento. Observa-se na Figura 5.4 tal representação, na qual as cores azul e verde representam as técnicas *Checklist* e *PBR*, respectivamente, e as perspectivas são representadas por diferentes tons de verde: *Designer*, *Tester* e *User* representadas por verde escuro, verde e verde claro, respectivamente. O tamanho dos retângulos representa o número de vezes que um defeito foi reportado e os rótulos identificam os defeitos.

Figura 5.4: Representação hierárquica dos defeitos de reportados em *R1* e *R2* obtida com Treemap

Na visualização apresentada na Figura 5.4 observa-se que foram utilizados os conjuntos de defeitos reportados em *R1* e *R2*, e que *replicação* define o primeiro nível de hierarquia (*R1* ou *R2*). A escolha dos dados dessas replicações para exemplificar a aplicação deve-se ao fato delas terem coletado as experiências em meses, como apresentado na Tabela 5.1. Adotando a

replicação como o primeiro nível da hierarquia é possível, com os recursos de interação oferecidos pela ferramenta, explorar separadamente as diferentes sub-árvores, como definido na abordagem proposta para meta-análise: analisa-se os conjuntos de dados separadamente e depois conjuntamente. Tal representação permitiu observar que os defeitos de *PG* foram mais reportados que os de *ATM*; em geral, *PBR* foi mais efetivo que *Checklist*, com exceção de *PG* em *R1*.

As observações motivaram uma extensão da organização para mapear as experiências dos participantes, buscando identificar possíveis influências da experiência na efetividade. Neste caso, o mapeamento hierárquico foi estendido incorporando a experiência dos participantes que reportaram cada defeito. Os nós folha, que representam detecções de um mesmo defeito e são mapeados para os retângulos interiores da Figura 5.4, foram subdivididos por participante, cujas experiências definem as cores dos retângulos. Tal mapeamento é esquematizado na Figura 5.5.

Figura 5.5: Organização hierárquica dos defeitos por documento, técnica e experiência

Entretanto, com esse número de níveis na hierarquia a interpretação da visualização fica dificultada pelas várias subdivisões, que resultam em retângulos relativamente pequenos, considerando o espaço disponível para projeção. Com isso, foi estabelecido um corte na hierarquia mantendo os parâmetros do projeto experimental (*Documento e Técnica*) e as regiões de cada nó foram coloridas conforme a experiência dos participantes que reportaram os respectivos defeitos reportados em cada par *Documento/Técnica*.

A partir desse mapeamento foram geradas três representações apresentadas nas Figuras 5.6, 5.7 e 5.8, dos resultados da análise isolada dos dados de *R1* e *R2*, e da análise dos dados agregados. A partir das experiências dos participantes, coletadas em meses, foi gerado valor médio das experiências, utilizado para a coloração. As várias experiências coletadas não introduziram variações significantes nas representações visuais quando usadas separadamente. Nessas figuras a escala de cores foi omitida, mas todas adotam a mesma escala, que varia do branco (zero em experiência) ao azul escuro (experiência máxima). Comparando as figuras, observa-se que:

Figura 5.6: Representação obtida a partir da organização hierárquica exposta na 5.5 com dados de *R1*

- Em *R1* (Figura 5.6), o grupo com maior experiência – que aplicou PBR no documento ATM – obteve melhor desempenho ao aplicar *Checklist* no documento PG. Este fato leva a dois questionamentos: i) maior experiência pode dificultar a absorção dos princípios do PBR? ii) é mais fácil, seja por familiaridade ou não, detectar erros no documento PG? A primeira delas considera a possibilidade de participantes terem mais dificuldade de se desvincular das suas práticas tradicionais em um primeiro momento, reduzindo a efetividade na primeira aplicação do PBR, o que remete a um dos questionamentos do experimento original, apontado no Capítulo 2;

Figura 5.7: Representação obtida a partir da organização hierárquica exposta na 5.5 com dados de *R2*

Figura 5.8: Representação obtida a partir da organização hierárquica exposta na 5.5 com dados de *R1* e *R2*

- Ainda em *R1*, o grupo com menor experiência – que aplicou *Checklist* no documento ATM – obteve melhor desempenho ao aplicar PBR no documento PG. Este fato leva a duas ponderações sobre a influência do documento na variação das medidas observadas; e sobre o próprio objetivo do estudo proposto por Basili et al. (1996c), qque era verificar se a efetividade de PBR é maior. Dois questionamentos podem ser derivados dessas ponderações: i) seria o PBR mais efetivo para inexperientes, já que as perspectivas impõem certa disciplina na sua aplicação? ii) para os experientes, após o domínio da técnica (fluência nas perspectivas), quanto ganha-se em eficiência?
- Em *R2* (Figura 5.7), observa-se que, em geral, PBR foi mais efetivo para ambos os grupos, e ainda, que foi mais efetivo que *Checklist* mesmo no grupo dos mais experientes. Isso impede a generalização da hipótese de que a experiência pode dificultar a absorção dos princípios do PBR. Mas, a maior efetividade na análise do documento PG persiste;

Figura 5.9: Representação obtida com os dados relativos às replicações *R1* e *R2* usando Treemap

- Ao analisar a representação dos dados agregados (Figura 5.8) observa-se uma variação na coloração, que ficou mais clara devido à diferença de experiência entre os grupos (experi-

ências de participantes de *R1* são, em geral, maiores do que em *R2*). Tal diluição dificulta a identificação da variação de experiência entre os grupos, mas pode-se recorrer a interações com outras visualizações, como a apresentada na Figura 5.9, na qual a cor representa a experiência (tons de verde claro, mais experientes) e é apresentada a experiência média do grupo que aplicou *Checklist* no documento PG. Em geral, observa-se que o grupo que aplicou PBR no documento PG é mais homogêneo e com participantes de menor experiência, no entanto foram mais efetivos ao aplicar PBR, do que quando aplicaram *Checklist* no documento ATM.

5.2.3 Análise de Defeitos Reportados por Documento, Tipo e Página

O mesmo conjunto de dados utilizado nas análises apresentadas na seção anterior foi submetido a outro ciclo de análise, tendo como foco a distribuição de defeitos por tipo e página do documento, que são fatores que podem influenciar a detecção de defeitos. Neste caso, outros mapeamentos visuais foram empregados. Observa-se na Figura 5.10 um mapeamento hierárquico que subdivide o espaço de projeção considerando documento (PG ou ATM), o tipo do defeito (segundo a taxonomia apresentada) e a replicação (*R1* e *R2*). O rótulo de cada retângulo identifica o defeito, o tamanho mapeia o número de vezes que o defeito foi reportado e a cor mapeia a página, de acordo com as escalas apresentadas na legenda (Figura 5.10(b)). Com base no histograma de distribuição de defeitos por página, o mapa de cores utilizado foi subdividido em três faixas de valores:

- defeitos associados a página zero são defeitos na estrutura do documento, e são representados por tons de azul;
- defeitos que ocorrem até a página dez são representados por tons de verde;
- defeitos que ocorrem a partir da página dez são mapeados em cores “quentes” (variando de amarelo a vermelho).

Observa-se na Figura 5.10 que os defeitos de estrutura do documento e aqueles contidos em páginas iniciais foram reportados em sua maioria no documento ATM, tendo sido reportado

(a) Instantâneo da Ferramenta

(b) “Legenda”

Figura 5.10: Representação hierárquica dos defeitos detectados em *R1* e *R2* obtida com técnica Treemap

apenas um no documento PG. Por outro lado, a grande maioria dos defeitos reportados no documento PG está concentrada em uma faixa de páginas (entre 10 e 18). Entretanto, a visualização não permite analisar a quantidade de defeitos reportados por tipo, uma vez que não há equilíbrio na distribuição de defeitos – há diferentes números de defeitos por tipo. Tal restrição motivou a utilização de um gráfico de dispersão para uma análise dessa distribuição. Na Figura 5.11 os eixos x e y representam, respectivamente, os tipos de defeitos (ênfatisados pela cor) e os

defeitos, exibindo todos os defeitos, reportados ou não, para permitir uma análise da sua distribuição. Observa-se que a maior variação entre documentos encontra-se nos defeitos do tipo *fato incorreto* (IF) e *informação inconsistente* (II), sendo que no documento PG eles ocorrem com mais frequência.

Figura 5.11: Gráfico de Dispersão com a distribuição de defeitos por tipo e documento (a cor enfatiza o tipo de defeito)

As observações relativas à distribuição por página e à variação de tipos de defeitos permitem formular novas hipóteses a serem consideradas na análise de efetividade das técnicas sob investigação: a maior eficácia observada na revisão do documento PG pode estar relacionada com essa concentração de defeitos e/ou com os tipos de defeitos. Observações como esta, assim como as apresentadas nas seções anteriores, são importantes para reformular os artefatos utilizados, pois permitem isolar fatores que podem influenciar as conclusões dos experimentos.

5.3 Geração de Dados: uma abordagem baseada em Algoritmos Genéticos

Como atividade subjacente, foi proposta uma abordagem para a simulação de dados de Engenharia de Software Experimental. A taxonomia de estudos experimentais proposta por [Travassos e Barros \(2003\)](#) apresentada na Seção 2.1, enfatiza o uso de modelos computacionais para a realização de experimentos, evitando o alto custo envolvido em experimentos *in vitro*. O conhecimento adquirido com as análises dos dados dos experimentos PBR inspirou a definição de um esquema de simulação de dados experimentais baseado em Algoritmo Genético que utiliza a idéia de alterações genéticas para criar de dados hipotéticos ([Garcia et al., 2005b](#)).

Embora os dados obtidos não permitam validar hipóteses, já que são simulados, podem ser úteis para refinar hipóteses e gerar novas hipóteses. A simulação permite criar e analisar cenários, e prospectar sobre novas situações para estabelecer requisitos para novos experimentos ou replicações, buscando ampliar a sua contribuição para a formação do corpo de conhecimento almejado, sem incorrer nos custos de realização de um estudo ([Garcia et al., 2005a](#)). O esquema, apresentado na Figura 5.12, toma como base os conjuntos de dados obtidos em estudos já realizados (estudos *In-Vitro*) para a geração de novos dados. Tais simulações dão origem a conjuntos de dados *In-Vituo* substancialmente maiores que os originais. O esquema de geração de dados hipotéticos deve respeitar as limitações impostas pelo projeto experimental do experimento original. Caso sejam utilizados múltiplos experimentos como base, os mesmos devem ser minimamente compatíveis, permitindo a junção de seus dados.

Um ponto importante, apresentado na Figura 5.12, é a validação do esquema de geração de dados. É preciso considerar que: 1) o esquema não deve gerar dados iguais aos dados de entrada; 2) os dados gerados, se forem muito diferentes dos dados originais, podem não ter correspondência com a realidade. No primeiro caso, os dados simulados não representam um ganho em relação aos dados já disponíveis, a não ser em quantidade. No segundo caso, o problema é que, ao introduzir uma variação exagerada, pode-se incorrer em dados que representam situações não observadas no mundo real. O importante é introduzir um componente de pequena variação aleatória para gerar algo diferente do que já existe, mas possível de ser encontrado no mundo real. Os dados obtidos devem ser analisados, não só para validar a estratégia de simula-

ção, mas também para prover subsídios à evolução da experimentação *In-Vitro*. Para apresentar a abordagem proposta, faz-se necessária uma breve apresentação dos conceitos de Algoritmos Genéticos.

Figura 5.12: Esquema de simulação de dados com base em Algoritmos Genéticos

5.3.1 Algoritmos Genéticos

Holland (1975) mostrou como o processo evolucionário de espécies pode ser aplicado a uma ampla variedade de problemas por meio de uma técnica conhecida como *algoritmo genético*. O algoritmo genético transforma uma *população* de objetos individuais, associados a uma função de medida (*fitness function*), em uma nova *geração* aplicando o princípio Darwiniano de reprodução e sobrevivência. Tal transformação se dá por meio de operações genéticas de reprodução, como a recombinação (*crossover*) e a *mutação*. Considerando a aplicação da abordagem para a solução de um problema, cada indivíduo na população representa uma possível solução. O algoritmo genético tenta encontrar uma solução “otimizada” para o problema em questão.

Os membros da população são representados por uma estrutura que codifica uma das possíveis soluções, chamada de cromossomo. Antes de aplicar algoritmo genético, é necessário definir uma estrutura para o cromossomo que represente o problema em foco (Koza, 1995). Também é necessário definir os parâmetros iniciais para controlar a execução do algoritmo,

como o tamanho da população e o número máximo de gerações a serem geradas, uma possível critério de parada para o algoritmo.

Em um processo de reprodução, um indivíduo é selecionado, re combinado e colocado na próxima geração, caso sua medida associada (*fitness*) assim o indique. A recombinação para criar novos indivíduos pode ocorrer por cruzamento “sexual” ou não. Na recombinação sexual, chamada *crossover*, um par de indivíduos (I_1 and I_2) é escolhido aleatoriamente e, considerando um ponto de corte entre 1 e o tamanho do cromossomo, são criados novos indivíduos pela combinação das partes inicial de um e final de outro, e vice-versa. Na Figura 5.13 é ilustrado este esquema de cruzamento.

Figura 5.13: Operação de *crossover*

Outra estratégia para mapear a evolução genética é a mutação. A operação de mutação consiste em alterar genes escolhidos aleatoriamente de um cromossomo semente, gerando uma evolução genética do mesmo. Considerando as estruturas usadas para representar a solução como seqüências de bits (0 e 1), a estratégia consiste em alterar genes escolhidos para o valor oposto, como apresentado na Figura 5.14. Esta estratégia é apropriada ao problema tratado, pois permite o controle da mutação para explorar diferentes cenários e perfis de participantes, que podem ser a “solução” mapeada no cromossomo.

Figura 5.14: Exemplo de operação de mutação

A função de medida transforma um cromossomo – uma seqüência de zeros e uns – em um número inteiro positivo que representa o valor da solução mapeada pelo cromossomo. Há dois passos envolvidos no mapeamento que podem ser considerados conjuntamente. O primeiro

passo é a *decodificação*, e o segundo é o cálculo da função de medida propriamente dita. No primeiro, a sequência de zeros e uns que mapeia uma solução deve ser convertida em parâmetros de entrada para a função. O segundo passo consiste em computar a medida, mapeando os parâmetros em valores inteiros positivos. Geralmente, o valor obtido deve ser inteiro positivo para permitir a maximização da solução.

5.3.2 Mapeamento do Problema

Primeiramente é preciso codificar dados relevantes dos experimentos em cromossomos, que possam ser manipulados pelo esquema de “evolução” baseado em algoritmo genético (AG)¹. Considerando que o objetivo é simular dados de experimentos a partir de dados de experimentos reais, a função de medida não é utilizada, pois não se busca maximizar uma solução, mas sim gerar novas “populações” de resultados.

Um aspecto importante é a definição do cromossomo. No caso, deseja-se mapear atributos de interesse (defeitos reportados, perfil de participantes, dentre outros) em seqüências de zeros e uns. Por exemplo, os defeitos reportados por participantes podem ser representados por uma seqüência de 32 ou 37 dígitos binários, para os documentos PG e ATM respectivamente, indicando cada defeito reportado como um e cada defeito não reportado como zero. A abordagem proposta para a mutação considera características do experimento expressas no projeto experimental, não só para mapear o problema, mas também para apoiar o controle da mutação – definindo parâmetros da mutação. A hierarquia apresentada na Figura 5.5 pode ser modificada considerando como folhas apenas dois grupos de experiências de participantes: alta e baixa. Dessa forma, obtém-se um conjunto de medidas estatísticas associadas ao grupo de sujeitos experientes que aplicaram PBR no documento PG, por exemplo. O mesmo é feito para os outros grupos: grupo de sujeitos com alta experiência que revisaram o documento ATM; grupo de sujeitos com baixa experiência que revisaram o documento PG; e grupo de sujeitos com baixa experiência que revisaram o documento ATM. Essas medidas – como média e desvio padrão – são usadas para definir quantos defeitos serão alterados em cada cromossomo, de acordo com o grupo ao qual pertence o participante usado como semente.

Para gerar dados, um cromossomo é escolhido aleatoriamente como “semente” e, conside-

¹Embora o esquema proposto seja baseado em algoritmos genéticos, ele será referenciado simplesmente como algoritmo genético

rando a experiência do participante mapeado no cromossomo escolhido, a partir das medidas estatísticas computadas para seu respectivo grupo, o número de mutações a serem feitas é definido. Apesar de ter sido estabelecida uma estratégia para a mutação, o critério para a determinação do número de mutações foi definido empiricamente: o número de defeitos mutados no cromossomo corresponde à metade do desvio padrão de defeitos reportados pelo grupo do participante usado como “semente”. Com essa abordagem é gerado o mapa de defeitos reportados por um participante “hipotético”, baseado em um caso real.

The screenshot shows the GA - In Virtuo software interface. It has a menu bar with 'Control', 'Subjects', 'ATM', 'PG', 'Types', and 'Result'. Below the menu bar, there are two data tables, 'ATM' and 'PG', each with a grid of data. The 'ATM' table has columns: Seed, SdId, Rep, ID, T/P, and 27 genes (1 to 27). The 'PG' table has columns: Seed, SdId, Rep, ID, T/P, and 27 genes (1 to 27). The data in the tables represents generated defects for different participants and techniques.

Seed	SdId	Rep	ID	T/P	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
10	15	2	1	3	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
5	9	2	2	1	1	0	0	0	0	0	0	1	0	2	0	0	0	0	1	0	0	0	0	0	0	2	0	0	1	0	0
17	17	1	3	3	0	0	0	1	0	0	0	0	0	-1	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0
13	21	2	4	2	0	0	-1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	2	2	5	1	0	1	1	0	0	1	0	1	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
9	14	2	6	2	0	0	1	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0
4	4	1	7	2	0	0	0	1	0	0	0	1	0	1	0	0	2	0	0	0	-1	0	0	0	0	0	2	0	0	0	0
16	16	1	8	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	-1	0	0
4	4	1	9	2	0	0	0	1	0	2	0	1	0	1	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	2
2	4	2	10	3	0	1	0	1	0	0	0	0	0	1	1	0	2	0	1	1	0	0	0	0	0	0	2	0	0	0	0
13	13	1	11	3	0	0	0	-1	2	1	0	1	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
8	8	1	12	2	0	0	0	1	0	0	0	1	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
11	17	2	13	3	0	-1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0

Seed	SdId	Rep	ID	T/P	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
9	9	1	1489	1	0	1	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0
11	11	1	1490	3	0	1	2	2	1	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
4	7	2	1491	2	0	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	2	0	0	0	0	0	0	0	0
8	12	2	1492	1	0	1	2	0	0	0	0	0	0	0	1	2	1	1	1	0	0	0	1	0	0	0	0	1	0	0	0
7	7	1	1493	2	0	1	1	2	1	0	0	0	0	1	0	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0
12	12	1	1494	1	0	1	2	0	0	0	0	0	0	1	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	1	0
4	7	2	1495	2	2	1	0	0	0	0	0	1	0	1	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
4	7	2	1496	2	0	1	0	2	0	0	0	1	0	1	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	0
11	11	1	1497	3	0	1	2	0	1	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	1	1498	3	1	0	0	2	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	2	0	0	0
2	2	1	1499	3	1	0	0	0	1	0	0	0	2	1	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	0
3	5	2	1500	3	1	1	1	0	1	0	0	0	2	1	0	0	0	2	1	0	0	1	0	0	0	1	0	0	0	0	0

Figura 5.15: Instantâneo da ferramenta que implementa a abordagem de mutação proposta exibindo defeitos gerados

A abordagem exposta foi implementada em uma ferramenta cuja interface é exibida na Figura 5.15. Nela são apresentados os defeitos gerados: as primeiras cinco colunas identificam o participante usado como semente pelos atributos *Seed*, *SdId*, *Rep*, *ID* e *T/P*, que representam respectivamente a indentificação do participante internamente à ferramenta, a identificação do participante na replicação, a replicação, a identificação do participante gerado e a identificação da Técnica/Perspectiva (0, 1, 2 e 3 para *Checklist*, *Designer*, *Tester* e *User*, respectivamente). Em seguida há o cromossomo com genes 0 e 1 e alguns números -1 e 2 representando as mu-

tações de 1 para 0, e de 0 para 1. Tais marcações servem apenas para visualizar as mutações na interface da ferramenta, ao gravar os dados gerados em arquivos, essas marcações são decodificadas para seus valores padrão (0 ou 1). Na ferramenta, é preciso estabelecer o número de elementos que se deseja gerar – esse parâmetro é usado como critério de parada – e o valor informado é dividido proporcionalmente conforme o projeto experimental (ver Figura 2.10).

5.3.3 Análise dos Dados Gerados

A estratégia de simulação apresentada na seção anterior, aplicada sobre os dados de R1 e R2 como semente, gerou um conjunto de defeitos reportados por 1500 participantes simulados, divididos em 750 para cada técnica (e documento), sendo 250 participantes para cada perspectiva do PBR. Os defeitos gerados foram armazenados em arquivo texto, sendo considerados como dados brutos para a instanciação do processo de visualização. Por se tratar de dados gerados, as atividades iniciais no espaço de dados (voltados para compatibilização) não se aplicam (pode-se considerar que os dados estão prontos para serem submetidos à exploração). Entretanto, é necessário estruturar os dados considerando os objetivos da análise.

Em um primeiro passo, técnicas de apresentação podem ser úteis para avaliar o conjunto de dados gerado, comparando-o aos dados utilizados como semente. Na Figura 5.16 é apresentada a distribuição de defeitos por tipo, para o conjunto semente e para conjunto simulado, considerando apenas os defeitos reportados. Observa-se que há uma distribuição similar, mas sem manter a proporção quantitativa entre elas. Observa-se também que defeitos do tipo *MD* (*Miscellaneous Defect*) aparecem em proporções pequenas no conjunto gerado, sendo que eles não ocorrem no conjunto original.

No conjunto de dados original foram reportados 151 defeitos dentre 2484 possibilidades (número total caso todos os defeitos fossem reportados por todos os participantes), o que corresponde a uma taxa da ordem de 6% de detecção de defeitos. Já o conjunto de dados simulados possui 18044 defeitos reportados, dentre 103500 possibilidades, o que corresponde a uma taxa da ordem de 17%. Tal variação é ocasionada pela combinação de dois fatores: a aleatoriedade na seleção de participantes usados como “sementes” e o uso do desvio padrão como parâmetro que guia a mutação. Dessa forma, os indivíduos gerados tendem a se concentrar em torno da média apresentada pelo conjunto original.

O aumento do número de registros justifica a aplicação da técnica Coordenadas Paralelas

(a) R1 e R2

(b) Dados simulados

Figura 5.16: Distribuição de defeitos por Tipo

Hierárquicas. Por isso foi necessário o tratamento dos dados obtidos para adequá-los à ferramenta XmdvTool: o atributo *Documento* (Doc) foi mapeado para 1 e 2, indicando ATM e PG, respectivamente; o atributo *Technique* foi convertido, sendo que os valores 1 e 2 representam *Checklist* e PBR, respectivamente; em perspectiva (*Persp*), 1 representa *Checklist*, 2, 3 e 4 representam as perspectivas *Designer*, *Tester* e *User*; por último, *Tipo de Defeito* foi mapeado para valores entre 1 e 6 representando os tipos A, E, IF, II, MD e O, respectivamente. Representações visuais desses dados por meio desta técnica são apresentadas na Figura 5.17. Observa-se na primeira delas (a) que há grupos bem definidos (indicados por diferentes cores) e coerentes com o conjunto de dados, o que não ocorre na segunda (b). O problema com a segunda é a indicação de um agrupamento (em verde) no eixo de documento (*Doc*) com valor médio em 1.5: a visualização não possibilita uma interpretação clara da semântica desses grupos, pois o agrupamento observado não reflete claramente as possibilidades existentes no conjunto de dados. A exploração interativa mostrou que os participantes simulados são menos eficazes ao

(a)

(b)

Figura 5.17: Agrupamentos hierárquicos dos dados simulados gerados com a ferramenta Xmdv-Tool – representados em Coordenadas Paralelas Hierárquicas

aplicar a perspectiva do usuário. O componente aleatório – na escolha da semente – pode ter influenciado tal resultado, mas os dados originais também apontam tal tendência, o que leva a um questionamento: a perspectiva do usuário possui alguma peculiaridade na aplicação que dificulta a assimilação de seu procedimento? Essa observação motiva, por exemplo, a revisão dos procedimentos adotados para treinamento nessa perspectiva.

5.4 Considerações Finais

Neste capítulo foi apresentada a instanciação do processo $ViDA_{ESE}$. Cada etapa do processo foi exercitada para demonstrar como as atividades propostas devem ser executadas. Os ciclos de análises expostos resultaram em observações que mostram que análises exploratórias podem contribuir para a evolução do Pacote de Laboratório, o que é fundamental para a formação do corpo de conhecimento almejado pelos pesquisadores da área. Muitas das observações apresentadas nas análises da Seção 5.2 constituem conhecimento novo que pode ser considerado em futuros experimentos.

Os exemplos apresentados têm como foco os papéis assumidos por replicadores de experimentos e, principalmente, projetistas, os quais têm a possibilidade de executar a meta-análise exploratória. Embora não tenham sido apresentados exemplos da instanciação do processo para participantes de experimentos, o processo supre as necessidades específicas para este caso, uma vez que trata-se de um sub-conjunto das possibilidades de análise.

A abordagem para simulação de dados de experimentos pode oferecer uma alternativa para o planejamento de experimentos, principalmente para o aprimoramento de hipóteses e lista de variáveis, além das apontadas por Carver (2003). No próximo capítulo são sumarizadas as contribuições deste trabalho, juntamente com perspectivas de trabalhos futuros.

Conclusões

Neste capítulo são apresentadas as principais contribuições do trabalho para a Engenharia de Software Experimental, bem como suas limitações. Adicionalmente, são apresentados trabalhos futuros que podem dar continuidade a esta linha de pesquisa, gerando outras contribuições.

Para investigar a tese proposta, foram realizados diversos ciclos de análise visual sobre os dados dos experimentos do Projeto *Readers*. Inicialmente, a exploração visual das duas primeiras replicações conduzidas no ICMC-USP (R1 e R2) permitiu identificar a necessidade de adequação do processo de visualização, dando origem ao processo $ViDA_{ESE}$. Para validar o processo $ViDA_{ESE}$ foram conduzidos outros ciclos de análise, envolvendo as quatro replicações conduzidas no ICMC-USP (R1-R4). Foi proposto um arcabouço de aplicação de análise visual exploratória que facilita a repetição de ciclos de análise e, conseqüentemente, o seu empacotamento no Pacote de Laboratório.

Também foram conduzidos ciclos de análise sobre dados agregados de múltiplas replicações, dando origem à abordagem proposta para a meta-análise exploratória. Buscando refinar e validar o processo proposto, durante um estágio na Mississippi State University, tanto a abordagem para meta-análise exploratória quanto o processo $ViDA_{ESE}$ foram instanciados sobre dados do experimento original (Basili et al., 1996c) e dados coletados em outras replicações conduzidas no âmbito do Projeto *Readers*, enumeradas no Capítulo 4.

O conhecimento adquirido durante os ciclos de análise motivou a proposta de uma abordagem de simulação de dados experimentais, baseada em Algoritmo Genético, que permite a prospecção sobre cenários hipotéticos, gerados a partir de dados reais.

6.1 Contribuições

A contribuição principal deste trabalho consiste no esforço de integração de técnicas de Visualização Exploratória na análise de dados de Engenharia de Software Experimental, apoiando atividades necessárias à sua evolução. Aliás, um dos méritos do trabalho é ser o primeiro passo no sentido de explorar essa integração. A evolução do corpo de conhecimento sobre técnicas e processos de Engenharia de Software, depende da efetividade dos estudos experimentais e de uma análise cuidadosa e extensiva dos dados coletados. São vários os fatores que influenciam a condução de experimentos, mas o Pacote de Laboratório é um elemento chave no processo de experimentação, e aprimorá-lo é uma das maneiras de contribuir para a formação do corpo de conhecimento. A integração da Visualização Exploratória permite extrair conhecimento não-antecipado, contribuindo para melhorar o Pacote de Laboratório e também motivando a realização de novos experimentos em vista do conhecimento adquirido com os ciclos de análise. A análise exploratória pode atuar como um elemento essencial para maximizar o conhecimento adquirido a partir dos dados coletados em experimentação.

Identificou-se a necessidade de adequar o processo de experimentação para integrar o uso de técnicas de visualização exploratória. Foi também necessário estender o processo de visualização, dando origem ao processo $ViDA_{ESE}$, para atender os requisitos da experimentação, que introduz a necessidade de replicação e de repetição das análises. Como consequência das dificuldades observadas na instanciação dos ciclos de análise, foi definido o arcabouço exposto no Capítulo 4, de estabelece requisitos desejáveis para ambientes de exploração visual. Dessa forma, foi estabelecida uma abordagem cuja sistematização facilita o reuso e provê consistência à análise.

Do ponto de vista de Visualização de Informação, o processo $ViDA_{ESE}$, juntamente com o arcabouço proposto são contribuições pontuais, pois a integração de ambos abre um conjunto de possibilidades de aplicação em análises de dados com características similares aos da Engenharia de Software Experimental, tratados neste trabalho.

Uma ferramenta que atenda os passos e requisitos identificados no processo pode ser útil

não só para a análise de dados oriundos de estudos experimentais, mas também para outros contextos em que seja necessário repetir de ciclos de análise e transferência de conhecimento. Segundo [Nonaka e Takeuchi \(1995\)](#), a interação social entre os membros de uma comunidade facilita o fluxo e a conversão de conhecimento implícito em formas explícitas de conhecimento. De uma forma geral, artefatos que expressam conhecimento, produzido por diferentes comunidades, refletem a perspectiva daqueles que estiveram envolvidos na sua produção. É necessário, portanto, usar meios para habilitar membros de diferentes comunidades a descobrir: i) como o conhecimento refletido em artefatos de uma comunidade relacionam-se com seus próprios conhecimentos; e ii) como o contexto do conhecimento refletido em artefatos está relacionado com o contexto de suas próprias comunidades. Segundo [Novak e Wurst \(2005\)](#), esse tipo de interação e integração entre comunidades envolve dois passos. Primeiro, é necessário compartilhar significados de informações, artefatos e procedimentos entre membros da comunidade, por meio de concretização de conhecimento em artefatos físicos, mentais e culturais gerados pela participação de membros na comunidade. Segundo, é necessário negociar e coordenar significados entre diferentes comunidades, que lidam com conhecimento especializado. Portanto, abordagens para compartilhar conhecimento e visualização colaborativa são importantes para apoiar as atividades propostas no FIRE (ver Figura 2.7). O impacto do uso de técnicas de visualização em outras questões, além da análise direta dos dados coletados, não foi investigado – como a transferência de conhecimento.

A partir das instanciações do processo $ViDA_{ESE}$ nos diversos ciclos de análise, foram feitos comentários e observações que sugerem que existe conhecimento não-antecipado relevante nos dados de experimentação. A abordagem proposta para meta-análise exploratória permite combinar dados de múltiplos experimentos, respeitando as restrições impostas pelos projetos experimentais. Dessa forma é possível explicitar não só o conhecimento adquirido com cada estudo, mas também a contribuição do estudo para o corpo de conhecimento, como um todo.

Finalmente, a abordagem proposta para simulação de dados experimentais permite a prospecção de cenários hipotéticos, baseada em informações coletadas dos dados reais. Pode-se criar cenários para explorar alternativas virtuais às composições de grupos de participantes, de acordo com o perfil definido. Essa possibilidade abre uma nova perspectiva à metodologia proposta por [Carver \(2003\)](#): além dos ciclos apresentados na Figura 2.8, pode-se usar dados

simulados para definir e/ou refinar hipóteses.

6.2 Limitações

O emprego efetivo de técnicas de visualização, e conseqüentemente a instanciação do processo $ViDA_{ESE}$, requer treinamento nas técnicas de visualização utilizadas. Esse treinamento é necessário principalmente para a definição de mapeamentos visuais efetivos, mas a interpretação também requer conhecimento mínimo das técnicas. Isso introduz uma limitação ao uso imediato das propostas desta tese. Entretanto, pode-se introduzir um papel adicional específico para a análise visual exploratória. Neste caso, poderia ser interessante considerar uma nova perspectiva de usuário, separando o criador das visualizações daquele que as interpreta. Entretanto, isso não tem impacto direto no processo $ViDA_{ESE}$. Também pela necessidade de treinamento, existe uma dificuldade associada a transferir o conhecimento embutido nos mapeamentos visuais. Portanto, o pacote de Laboratório precisa ser modificado para incorporar treinamento dos analistas nas técnicas visuais empregadas.

Outra dificuldade associada à instanciação do processo $ViDA_{ESE}$ consiste na indisponibilidade de um ambiente integrado capaz de suportar diversas atividades, nos diferentes espaços – espaço de dados e espaço visual. A ferramenta desenvolvida limita-se a exemplificar uma possível solução computacional que ilustra como as atividades propostas podem ser apoiadas, mas sem integrar múltiplas técnicas de visualização. Espera-se que o arcabouço proposto ofereça as diretrizes para o desenvolvimento de tal ambiente, no âmbito de outros trabalhos.

Quanto à abordagem para simulação de dados experimentais, há diversos pontos em aberto. Por exemplo, o número de mutações por geração é definido empiricamente; a distribuição desigual dos participantes quanto à experiência – por exemplo, se o cenário simulado considerar apenas participantes experientes como semente, o baixo número de participantes nessa categoria faz com que um mesmo participante seja escolhido como semente para várias mutações, e os resultados obtidos não são satisfatórios.

6.3 Desenvolvimentos Futuros

As limitações apontadas anteriormente indicam os principais passos a serem seguidos. Dentre eles destacam-se:

- I. Definir mecanismos de apoio à transferência de conhecimento embutido nas representações visuais (e seus mapeamentos). Tais mecanismos deverão explicitar os mapeamentos visuais e o que se observa em cada um deles, além daquilo que é definido no *template* de uma visualização. Para isso, é necessário incorporar treinamento em cada técnica utilizada, facilitando a assimilação do conhecimento.
- II. A integração de técnicas de visualização exploratória teve como foco os dados oriundo de experimentos. Outra possibilidade é investigar como a visualização pode ser usada para apoiar a identificação e a transferência de conhecimento sobre experimentos e sobre técnicas de Engenharia de Software.
- III. Extensão da ferramenta desenvolvida para facilitar a instanciación do processo $ViDA_{ESE}$, incorporando outras técnicas de visualização. Para isso, o arcabouço proposto por [Shimabukuro \(2004\)](#) pode ser usado como base.
- IV. Extensão e generalização da abordagem proposta para a simulação de dados experimentais, para a simulação de experimentos *in silico*. Seria interessante simular tanto o participante (definido por suas características), quanto os defeitos por ele reportados, seguindo parâmetros definidos pelo experimentador.
- V. Integração das atividades propostas no processo $ViDA_{ESE}$ à ferramenta desenvolvida por [Maltez Junior \(2006\)](#), considerando a base de dados extensível – que modela a evolução do Pacote de Laboratório ao longo das replicações.

Referências Bibliográficas

- AHLBERG, C.; WISTRAND, E. Ivey: An information visualization and exploration environment. In: *Proceedings of InfoVis 95, IEEE Symposium on Information Visualization*, 1995, p. 66–73.
- AMARAL, E. A. G. G.; CHAPETTA, W. A.; TRAVASSOS, G. H. A process for experiment packaging. VII Workshop on NSF-CNPq Readers Project, 2003.
- ANKERST, M. Visual data mining. Tese de Doutorado, Universidade de Munique, publicado em www.dissertation.de, 2000.
- ANKERST, M.; BERCHTOLD, S.; KEIM, D. A. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In: *INFOVIS '98: Proceedings of the 1998 IEEE Symposium on Information Visualization*, Washington, DC, USA: IEEE Computer Society, 1998, p. 52–60.
- ANKERST, M.; KEIM, D. A.; KRIEGEL, H.-P. Circle segments: A technique for visually exploring large multidimensional data sets. In: *Proc. IEEE Visualization'96, Hot Topic Session*, San Francisco, CA, 1996.
- ARTERO, A. O.; OLIVEIRA, M. C. F. Viz3d: Effective exploratory visualization of large multidimensional data sets. In: *Proceedings of XVII SIBGRAPI Brazilian Symposium on Computer Graphics and Image Processing*, Curitiba, PR, Brazil, 2004, p. 340–347.
- ARTERO, A. O.; DE OLIVEIRA, M. C. F.; LEVKOWITZ, H. Uncovering clusters in crowded parallel coordinates visualizations. In: *Proceedings of IEEE Symposium on Information Visualization*, Austin, TX, 2004, p. 81–88.
- ASIMOV, D. The grand tour: a tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, v. 6, n. 1, p. 128–143, 1985.
- BASILI, V. R.; CALDIERA, G.; LANUBILE, F.; SHULL, F. Investigating focused techniques for understanding frameworks. *Proc. of the 1st International Workshop on Empirical Studies of Software Maintenance (WESS 96)*, p. 49–53, 1996a.

- BASILI, V. R.; CALDIERA, G.; LANUBILE, F.; ; SHULL, F. Studies on reading techniques. *Proc. of the Twenty-First Annual Software Engineering Workshop*, v. SEL-96-002, p. 59–65, 1996b.
- BASILI, V. R.; GREEN, S.; LAITENBERGER, O.; LANUBILE, F.; SHULL, F.; SØRUMGÅRD, S.; ZELKOWITZ, M. V. The empirical investigation of perspective-based reading. *Empirical Software Engineering: An International Journal*, v. 1, n. 2, p. 133–164, 1996c.
- BASILI, V. R.; GREEN, S.; LAITENBERGER, O.; LANUBILE, F.; SHULL, F.; SØRUMGÅRD, S.; ZELKOWITZ, M. V. Lab package for the empirical investigation of perspective-based reading. Available in: http://www.cs.umd.edu/projects/SoftEng/ESEG/manual/pbr_package/manual.html, 1998.
- BASILI, V. R.; SELBY, R. W. Comparing the effectiveness of software testing strategies. *IEEE Transactions on Software Engineering*, v. 13, n. 12, p. 1278–1296, 1987.
- BASILI, V. R.; SELBY, R. W.; HUTCHENS, D. H. Experimentation in software engineering. *IEEE Transactions on Software Engineering*, v. 12, n. 7, p. 733–743, 1986.
- BASILI, V. R.; SHULL, F.; LANUBILE, F. Building knowledge through families of experiments. *IEEE Transaction on Software Engineering*, v. 25, n. 4, p. 456–473, 1999.
- BERTIN, J. *Graphics and graphics information processing*. Berlin: Walter de Gruyter & Co., 1981.
- BESHES, C.; FEINER, S. Autovisual: Rule-based design of interactive multivariate visualizations. *IEEE Computer Graphics & Applications*, v. 13, n. 4, p. 41–49, 1993.
- CARD, S. K.; MACKINLAY, J.; SHNEIDERMAN, B. *Readings in information visualization: Using vision to think*. San Francisco, CA: Morgan Kaufmann Publishers, 1999.
- CARD, S. K.; MACKINLAY, J. D. The structure of the information visualization. In: *Proceedings of InfoVis97 – IEEE Symposium on Information Visualization*, Phoenix, 1997, p. 92–99.
- CARPENDALE, M. S. T.; COWPERTHWAIT, D. J.; FRACCHIA, F. D. Extending distortion viewing from 2d to 3d. *IEEE Computer Graphics and Applications*, v. 17, n. 4, p. 42–51, 1997.
- CARVER, J. C. *The impact of background and experience on software inspection*. Tese de Doutorado, University of Maryland, Washington-DC, 2003.
- CHEN, C. *Information visualization: Beyond the horizon*. 2nd ed. ed. Springer, 2004.
- CLEVELAND, W. S. *Visualizing data*. Summit NJ: Hobart Press, aT&T Bell Laboratories, Murray Hill (NJ), 1993.

- CRAWFORD, S. L.; FALL, T. C. Projection pursuit techniques for visualizing high-dimensional data sets. In: G. M. NIELSON, B. S.; ROSENBLUM, L. J., eds. *Visualization in Scientific Computing*, IEEE Computer Society Press, 1990, p. 94–108.
- DOSHI, P. R.; RUNDENSTEINER, E. A.; WARD, M. O. Prefetching for visual data exploration. In: *Database Systems for Advanced Applications (DASFAA)*, 2003.
- EICK, S. G. Data visualization sliders. In: *Proceedings UIST'94*, 1994, p. 119–120.
- EICK, S. G. Visual discovery and analysis. *IEEE Transactions on Visualization and Computer Graphics*, v. 6, n. 1, p. 44–58, 2000.
- FAYYAD, U. M.; PIATETSKY-SHAPIO, G.; SMYTH, P. *Advances in knowledge discovery and data mining*, cap. From data mining to knowledge discovery: An overview Menlo Park (CA): AAAI/MIT Press, p. 1–34, 1996a.
- FAYYAD, U. M.; PIATETSKY-SHAPIO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *Ai Magazine*, v. 17, n. 3, p. 37–54, 1996b.
- FAYYAD, U. M.; PIATETSKY-SHAPIO, G.; SMYTH, P. Knowledge discovery and data mining: Towards a unifying framework. In: *Knowledge Discovery and Data Mining*, 1996c, p. 82–88.
- FEINER, S.; BESHES, C. Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. In: *Proc. UIST-90, ACM Symposium on User Interface Software and Technology*, Snowbird (UT), 1990, p. 76–83.
- FUA, Y.-H.; WARD, M. O.; RUNDENSTEINER, E. A. Hierarchical parallel coordinates for exploration of large datasets. In: *Proceedings of Visualization'99*, 1999a, p. 43–50.
- FUA, Y.-H.; WARD, M. O.; RUNDENSTEINER, E. A. Navigating hierarchies with structure-based brushes. In: *INFOVIS*, 1999b, p. 58–64.
- FUA, Y.-H.; WARD, M. O.; RUNDENSTEINER, E. A. Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. In: *IEEE Trans. on Visualization and Computer Graphics*, 2000, p. 150–159.
- FUSARO, P.; LANUBILE, F.; VISAGGIO, G. A replicated experiment to assess requirements inspection techniques. *Empirical Software Engineering*, v. 2, n. 1, p. 39–57, 1997.
- GARCIA, R. E.; CARVER, J. C.; DE OLIVEIRA, M. C. F.; MALDONADO, J. C. In-virtual experimental studies: An approach based on genetic algorithms. *2nd Experimental Software Engineering Latin American Workshop (ESELAW)*, v. 1, 2005a.
- GARCIA, R. E.; DE OLIVEIRA, M. C. F.; MALDONADO, J. C. Genetic algorithms to support software engineering experimentation. In: *Proceedings IV International Symposium on Empirical Software Engineering (ISESE)*, Noosa Heads, Australia, 2005b, p. 1–10.

- GARCIA, R. E.; DE OLIVEIRA, M. C. F.; MALDONADO, J. C.; MENDONÇA, M. Empirical software engineering using visual data analysis. *Experimental Software Engineering Latin American Workshop (ESELAW)*, v. 1, 2004a.
- GARCIA, R. E.; DE OLIVEIRA, M. C. F.; MALDONADO, J. C.; MENDONÇA, M. Visual analysis of data from empirical studies. In: *Proceedings of the 2004 International Conference of Distributed Multimedia Systems – International Workshop on Visual Languages and Computing*, San Francisco: Knowledge Systems Institute – Visual Computing Lab, UT-Dallas, 2004b.
- GARCIA, R. E.; DE OLIVEIRA, M. C. F.; MALDONADO, J. C.; MENDONÇA, M. Visual analysis of data from empirical studies. *Cadernos de Computação – ICMC - Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo*, v. 5, n. 1, p. 183–207, (Long Version), 2004c.
- HETZEL, W. C. *An experimental analysis of program verification methods*. Tese de Doutorado, University of North Carolina, Chapel Hill, 1976.
- HOFFMAN, P. E. *Table visualizations: A formal model and its applications*. Tese de Doutorado, University of Massachusetts, Lowell, MA, 1999.
- HOHN, E. N. *Técnicas de leitura de especificação de requisitos de software: estudos empíricos e gerência de conhecimento em ambientes acadêmico e industrial*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, (in Portuguese), 2003.
- HOLLAND, J. H. *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press, 1975.
- INSELBERG, A. Multidimensional detective. In: *Proceedings of IEEE Symposium on Information Visualization*, Phoenix (AZ), 1997, p. 100–107.
- INSELBERG, A.; DIMSDALE, B. Parallel coordinates: A tool for visualizing multidimensional geometry. In: *Proceedings of IEEE Visualization '90*, Los Alamitos-CA: IEEE Computer Society Press, 1990, p. 360–375.
- JR., C. T.; TRAINA, A.; WU, L.; FALOUTSOS, C. Fast feature selection using the fractal dimension. In: *XV Brazilian Symposium on Databases (SBBD)*, 2000.
- KAMSTIES, E.; LOTT, C. M. *An empirical evaluation of three defect-detection techniques*. Relatório Técnico ISERN 95-02, Department of Computer Science, University of Kaiserslautern, Kaiserslautern-Germany, 1995a.
- KAMSTIES, E.; LOTT, C. M. An empirical evaluation of three defect-detection techniques. In: SCHÄFER, W.; BOTELLA, P., eds. *Proceedings of the Fifth European Software Engineering Conference*, Springer-Verlag, 1995b, p. 362–383.

- KEIM, D. A. Pixel-oriented visualization techniques for exploring very large databases. *Journal of Computational and Graphical Statistics*, v. 5, n. 1, p. 58–77, 1996.
- KEIM, D. A. Visual database exploration techniques. In: *Proc. Tutorial KDD'97 Int. Conf. on Knowledge Discovery and Data Mining*, Newport Beach, CA, disponível em <http://www.informatik.uni-halle.de/keim/PS/KDD97.pdf>, 1997.
- KEIM, D. A. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, v. 6, n. 5, p. 59–78, 2000.
- KEIM, D. A. Information visualization and visual data mining. *IEEE Transactions on Visualization And Computers Graphics*, v. 7, n. 1, p. 100–107, 2002.
- KEIM, D. A.; KRIEGEL, H.-P. Visualization techniques for mining large databases: A comparison. *IEEE Transactions on Knowledge and Data Engineering*, v. 8, n. 6, p. 923–938, 1996.
- KEIM, D. A.; KRIEGEL, H.-P.; SEIDL, T. Supporting data mining of large databases by visual feedback queries. In: *Proc. 10th Int. Conf. on Data Engineering*, Houston-Texas, disponível em <http://www.dbs.informatik.uni-muenchen.de/dbs/projekt/papers/datamining.ps>, 1994, p. 302–313.
- KITCHENHAM, B.; LINKMAN, S.; LAW, D. Desmet: a methodology for evaluating software methods and tools. *Computing & Control Engineering Journal*, p. 120–127, 1997.
- KOZA, J. R. Survey of genetic algorithms and genetic programming. John R. Koza. Survey of genetic algorithms and genetic programming. In *Proceedings of the Wescon 95 - Conference Record: Microelectronics, Communications Technology, Producing Quality Products, Mobile and Portable Power, Emerging Technologies*, San Francisco, CA, 7-9 November 1995. IEEE, New York, NY., 1995.
- LANUBILE, F.; VISAGGIO, G. Evaluating empirical models for the detection of high-risk components: Some lessons learned. 20th Annual Software Engineering Workshop, 1995.
- LEBLANC, J.; WARD, M. O.; WITTELS, N. Exploring n-dimensional databases. In: *Proc. IEEE Visualization'90*, San Francisco, CA, 1990, p. 230–237.
- LEUNG, Y.; APPERLEY, M. A review and taxonomy of distortion-orientation presentation techniques. *ACM Transactions on Computer-Human Interaction*, v. 1, n. 2, p. 126–160, 1994.
- LINKMAN, S.; ROMBACH, H. D. Experimentation as a Vehicle for Software Technology Transfer - A Family of Software Reading Techniques. *Information and Software Technology*, v. 39, p. 777–780, 1997.
- LOPEZ, N.; KREUSELER, M.; SCHUMANN, H. A scalable framework for information visualization. In: *Transactions on Visualization and Computer Graphics*, 2001.

- MALDONADO, J. C.; CARVER, J.; SHULL, F.; FABBRI, S.; DÓRIA, E.; MARTIMIANO, L.; MENDONÇA, M.; BASILI, V. R. Perspective-Based Reading: a Replicated Experiment Focused on Individual Reviewer Effectiveness. *Empirical Software Engineering: An International Journal - (EMSE)*, v. 11, n. 1, 2006.
- MALTEZ JUNIOR, U. P. *Expvvt: uma ferramenta de apoio à condução de experimentos baseados em documentos de requisitos*. Dissertação de Mestrado, ICMC-USP, dissertação de Mestrado, ICMC-USP, São Carlos, SP, 2006.
- MENDONÇA, M. G.; MALDONADO, J. C.; DE OLIVEIRA, M. C. F.; CARVER, J.; FABBRI, S. C.; SHULL, F.; TRAVASSOS, G. H.; HÖHN, E. N.; BASILI, V. R. A framework to coordinate and evolve software engineering controlled experiments, submitted to *Empirical Software Engineering: An International Journal - (EMSE)*, 2006.
- MILLER, J. Applying meta-analytical procedures to software engineering experiments. *The Journal of Systems and Software*, v. 54, n. 1, p. 29–39, 2000.
- MILLER, J. Replicating software engineering experiments: a poisoned chalice or the holy grail. *Information & Software Technology*, v. 47, n. 4, p. 233–244, 2005.
- MORENO, V. *Concepção e implementação de um modelo de coordenação para uma plataforma de visualização exploratória*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, São Carlos-SP, 2006.
- MYERS, G. J. A controlled experiment in program testing and code walkthroughs/inspections. *Communications of the ACM*, v. 21, n. 9, p. 760–768, 1978.
- NASON, G. *Three-dimensional projection pursuit*. Relatório Técnico ISERN 95-02, Department of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TW, UK, 1996.
- NONAKA, I.; TAKEUKI, H. *The knowledge creating company*. Oxford University Press, 1995.
- NORTH, C.; SHNEIDERMAN, B. Snap-together visualization: can users construct and operate coordinated visualizations? *International Journal of Human-Computer Studies*, v. 53, n. 5, 2000.
- NOVAK, J.; WURST, M. Collaborative knowledge visualization for cross-community learning. In: TERGAN, S.-O.; KELLER, T., eds. *Knowledge and Information Visualization*, Springer, 2005, p. 95–116 (*Lecture Notes in Computer Science*, v.3426).
- DE OLIVEIRA, M. C. F.; LEVKOWITZ, H. Visual data exploration and mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, v. 9, n. 3, p. 378–394, 2003.
- DE OLIVEIRA, M. C. F.; SHIMABUKURO, M. H. Conceptual model for adaptable and extensible visual data exploration. In: *Proceedings SPIE Conference on Visualization and Data Analysis 2004*, San Jose, California, 2004, p. 212–222.

- PAGEL, B.-U.; KORN, F.; FALOUTSOS, C. Deflating the dimensionality curse using multiple fractal dimensions. In: *International Conference on Data Engineering*, San Diego, CA, USA, 2000, p. 589–598.
- PARSAYE, K.; CHIGNELL, M. *Intelligent database tools & applications: Hyperinformation access, data quality, visualization, automatic discovery*. New York: John Wiley & Sons, 560 p., 1993.
- PERRY, D. E.; SIY, H. P.; VOTTA, L. G. Parallel changes in large scale software development: An observational case study. In: *International Conference on Software Engineering*, 1998, p. 251–260.
- PICKETT, R. M.; GRINSTEIN, G. G. Iconographic displays for visualizing multidimensional data. In: *Proc. IEEE Conf. Systems, Man and Cybernetics*, Beijing and Shenyang, China, 1988, p. 514–519.
- PORTER, A. A.; JR., L. G. V.; BASILI, V. R. Comparing detection methods for software requirements inspections: A replicated experiment. *IEEE Transactions on Software Engineering*, v. 21, n. 6, p. 563–575, 1995.
- PRECHELT, L.; UNGER, B. A series of controlled experiments on design patterns: Methodology and results. In: *Proc. Softwaretechnik'98, GI Conference, Paderborn*, 1998, p. 53–60.
- RAO, R.; CARD, S. K. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In: *Proceedings of ACM Conference on Human Factors in Computing Systems – CHI'94*, Boston, Massachusetts: ACM, 1994, p. 318–322.
- REZENDE, S. O.; PUGLIESI, J. B.; MELANDA, E. A.; PAULA, M. F., eds. *Mineração de dados*, cap. 12 Manole, p. 307–335, 2003.
- SHIMABUKURO, M. H. *Visualizações temporais em uma plataforma de software extensível e adaptável*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, São Carlos-SP, 2004.
- SHIMABUKURO, M. H.; FLORES, E. F.; DE OLIVEIRA, M. C. F. Coordinated views to assist exploration of spatio-temporal data: a case study. In: *Proceedings of International Conference on Coordinated & Multiple Views in Exploratory Visualization*, Londres, 2004, p. 107–117.
- SHNEIDERMAN, B. Tree visualization with treemaps: A 2d space-filling approach. *ACM Trans. Graphics*, v. 11, n. 1, p. 92–99, 1992.
- SHNEIDERMAN, B. *Dynamic queries for visual information seeking*. Relatório Técnico CS-TR-3022, UMCP-CSD, College Park, Maryland 20742, U.S.A., 1994.
Disponível em citeseer.nj.nec.com/shneiderman94dynamic.html

- SHULL, F.; BASILI, V. R.; CARVER, J.; MALDONADO, J. C.; TRAVASSOS, G. H.; MENDONÇA, M.; FABBRI, S. Replicating software engineering experiments: Addressing the tacit knowledge problem. In: IEEE COMPUTER SOCIETY, ed. *2002 International Symposium on Empirical Software Engineering*, Nara, Japan, 2002, p. 7–16.
- SHULL, F.; RUS, I.; BASILI, V. R. How perspective-based reading can improve requirements inspections. *Computer*, v. 33, n. 7, p. 73–79, 2000.
- SIPS, M.; SCHNEIDEWIND, J.; KEIM, D. A.; SCHUMANN, H. Scalable pixel-based visual interfaces: Challenges and solutions. In: *Proceedings of IEEE Symposium on Information Visualization*, 2006, p. 32–38.
- SOBOL, M. G.; KLEIN, G. New graphics as computerized displays for human information processing. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 19, n. 4, p. 893–898, 1989.
- SOVA, D. W.; SMIDTS, C. Increasing testing productivity and software quality: A comparison of software testing methodologies within NASA. *Empirical Software Engineering: An International Journal*, v. 1, n. 2, p. 165–188, 1996.
- TRAVASSOS, G. H.; BARROS, M. Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering. In: VERLAG, F. I., ed. *The Future of Empirical Studies in Software Engineering - 2ND International Workshop*, Roman Castles, Italy, ISBN 3-8167-6418-5, 2003, p. 117–130.
- WARD, M. O. Xmdvtool: integrating multiple methods for visualizing multivariate data. In: *VIS '94: Proceedings of the conference on Visualization '94*, Los Alamitos, CA, USA: IEEE Computer Society Press, 1994, p. 326–333.
- WARE, C. *Information visualization: Perception for design*. Second edition ed. Morgan Kaufmann Publishers, 2000.
- WEGMAN, E. J. Hyperdimensional data analysis using parallel coordinates. *Journal of American Statistical Association*, v. 85, n. 411, 1990.
- WEGMAN, E. J.; LUO, Q. *High dimensional clustering using parallel coordinates and the grand tour*. Relatório Técnico 124, George Mason University, Fairfax, Virginia 22030, U.S.A., 1996.
- WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. *Experimentation in software engineering: An introduction*. Boston (USA): Kluwer Academic Publishers, 1999.
- WONG, P. C. Visual data mining. *IEEE Computer Graphics & Applications*, v. 9, n. 5, p. 20–21, 1999.

- WONG, P. C.; BERGERON, R. D. *Scientific visualization - overviews, methodologies and techniques*, cap. 30 Years of Multidimensional Multivariate Visualization Los Alamitos (CA): IEEE Computer Society Press, p. 3–33, 1997.
- WOOD, M.; ROPER, M.; BROOKS, A.; MILLER, J. Comparing and combining software defect detection techniques: A replicated empirical study. *6th European Software Engineering Conference/5th ACM SIESOFT Symposium on the Foundations of Software Engineering, LNCS*, v. 1301, p. 262–277, 1997.
- ZELKOWITZ, M. V.; WALLACE, D. Experimental validation in software engineering. *Information and Software Technology*, v. 39, n. 11, p. 735–743, 1997.