

**Universidade Federal do Rio de Janeiro
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
Programa de Engenharia de Sistemas e Computação**

Exame de Qualificação - Doutorado

**VisAr3D: Uma Abordagem Baseada em Realidade Virtual e
Aumentada para o Ensino de Arquitetura de Software**

Aluna: Claudia Susie Camargo Rodrigues

Orientadora: Cláudia Maria Lima Werner

**RIO DE JANEIRO, RJ – BRASIL
NOVEMBRO DE 2009**

Índice

Índice	ii
Índice de Figuras	v
Índice de Tabelas	vi
1. Introdução	1
1.1. Objetivo e Questão de Pesquisa	2
1.2. Organização da Monografia	3
2. Ensino de Engenharia de Software	4
2.1. Ferramentas de Apoio ao Ensino de Engenharia de Software	5
2.2. Arquitetura de Software	9
2.2.1. Definições	9
2.2.2. Benefícios	9
2.2.3. Estilos Arquiteturais	11
2.2.4. Padrões de Projeto (<i>Design Pattern</i>)	13
2.2.5. Propriedades Arquiteturais	13
2.2.6. Linguagens de Definição de Arquitetura	14
2.2.7. Análise de Arquitetura de Software	14
2.3. Considerações Finais	15
3. Revisão <i>Quasi</i> -Sistemática	16
3.1. Processo para Realização da Revisão <i>Quasi</i> -Sistemática de Literatura	17
3.2. Planejamento da Revisão <i>Quasi</i> -Sistemática	17
3.2.1. Questão de pesquisa	17
3.2.2. Fontes para busca, termos e sinônimos	17
3.2.3. <i>Strings</i> de busca	18
3.2.4. Critérios para inclusão e exclusão de estudos	18
3.2.5. Processo de seleção dos estudos	19
3.2.6. Avaliação da qualidade dos estudos	19
3.3. Condução da Revisão	19
3.3.1. Análise dos documentos recuperados	19
3.3.2. Resultado das Buscas nas Bibliotecas Digitais	20
3.3.3. Análise dos documentos recuperados	20
3.3.4. Extração da informação	20
3.4. Categorização das iniciativas encontradas	21
3.5. Discussão dos resultados	21
3.6. Considerações finais	24

4. Realidade Virtual e Aumentada	26
4.1. Um Breve Histórico	26
4.2. Definições	27
4.3. Dispositivos utilizados	29
4.3.1. Sistemas de Exibição	30
4.3.2. Dispositivos Hápticos	31
4.3.3. Dispositivos Móveis	32
4.4. Áreas de aplicações	32
4.5. Recursos e Ferramentas de RV e RA	36
4.6. Considerações finais	39
5. A Abordagem VisAr3D	40
5.1. Caracterização do Problema	40
5.2. Requisitos da abordagem	41
5.2.1. Apoiar o desenvolvimento e a participação dos alunos em projetos mais complexos	41
5.2.2. Reduzir a distância entre a teoria e a prática	41
5.2.3. Apoiar a utilização de aspectos não somente técnicos	42
5.2.4. Ser atrativo para o aluno	43
5.3. Visão Geral da arquitetura	43
5.4. Cenário de utilização	44
5.5. Ferramental de Apoio à abordagem VisAr3D	45
5.5.1. Módulo Arquitetural	45
5.5.1.1. Odyssey	46
5.5.1.2. FrameDoc	47
5.5.2. Módulo de Realidade Aumentada	48
5.5.3. Módulo de Realidade Virtual	48
5.5.3.1. Informações contextualizadas	49
5.5.3.2. Exploração no ambiente 3D	51
5.5.3.3. Busca através de filtros	51
5.5.3.4. Troca de informações entre alunos	52
5.6. Repositórios	52
5.7. Recursos necessários	53
5.8. Exemplos de Aplicação da Abordagem	55
5.8.1 Abordagem ROOSC	56
5.8.2. Abordagem PREVia	58
5.9. Considerações Finais	59

6. Conclusões	61
6.1. Trabalhos Relacionados	61
6.2. Contribuições esperadas	62
6.3. Situação Atual	64
6.3.1. Revisão da Literatura	64
6.3.2. Estudo de linguagens e ambientes	64
6.3.2.1. Protótipo de RA	64
6.3.2.2. Protótipo de RV	65
6.3.2.3. Evolução do Odyssey	67
6.3.2.4. Odyssey-XMI	67
6.3.2.5. Estudo do X3D	68
6.3.3. Redação de Artigos e Relatórios Técnicos	68
6.3.4. Avaliação da Abordagem VisAr3D	69
6.4. Fases para o Desenvolvimento e Cronograma	70
7. Referências	72

Índice de Figuras

Figura 2.1. Alguns cartões do <i>Problems and Programmers</i>	7
Figura 2.2. Tela do WriteOn	8
Figura 4.1. Representação simplificada da continuidade virtual (MILGRAM & KISHINO, 1994)	27
Figura 4.2. RA na transmissão de um jogo de futebol	28
Figura 4.3. HMD baseado em vídeo	30
Figura 4.4. Encanamento industrial (AZUMA <i>et al.</i> , 2001)	33
Figura 4.5. Marcador do ARToolkit	37
Figura 5.1. Visão geral da abordagem VisAr3D	44
Figura 5.2. Projeção da arquitetura de software na parede (simulação)	45
Figura 5.3. Dispositivo móvel dos alunos (simulação)	45
Figura 5.4. Exemplo de arquitetura em 2D com padrão gráfico	49
Figura 5.5. Visualização 3D com informações contextualizadas	50
Figura 5.6. Projeção Polarizada	54
Figura 6.1. Alunos interagindo com modelos de sistemas estruturais	65
Figura 6.2. Protótipo com RA mostrando reações de apoio em um modelo de viga	66
Figura 6.3. Protótipo com RA mostrando as regiões tensionadas em um modelo de viga	66
Figura 6.4. Protótipo de Aplicação de Realidade Virtual	67

Índice de Tabelas

Tabela 2.1. Grupos de Estilos Arquiteturais	12
Tabela 3.1. Artigos retornados nas respectivas máquinas de busca	20
Tabela 6.1. Cronograma	71

1. Introdução

Os profissionais de Engenharia de Software (ES) que trabalham na indústria apresentam uma insatisfação quanto ao nível de preparação dos universitários recém-formados que entram no mercado de trabalho (SHAW, 2000), (HILBURN & TOWHIDNEJAD, 2007), (THOMPSON & EDWARDS, 2009). A raiz do problema parece ser a forma que a ES é ensinada. Nos últimos anos, a academia tem investido muito esforço para mitigar esse problema através da elaboração de novas formas de ensinar ES. Algumas instituições chegam a inovar ao desenvolverem algumas ferramentas que têm como objetivo mobilizar os alunos e ensinar ou apoiar o ensino de ES (BAKER *et al.*, 2005), (DE LUCENA *et al.*, 2006), (TRONT *et al.*, 2006) e MOHRENSCHILDT & PETERS, 1998).

Dentro da ES, a Arquitetura de Software (AS), como uma etapa importante no desenvolvimento de software, torna-se fundamental, para atender a demanda de sistemas cada vez mais sofisticados e complexos. Faz-se necessário, portanto, um programa de educação que, além de ensinar as bases teóricas, possibilite a experiência prática e que tente aprofundar o entendimento das competências não apenas técnicas, na formação de um bom arquiteto de software (HUANG & DISTANTE, 2006), (CHENOWETH *et al.*, 2007), (CREIGHTON & SINGER, 2008).

Reconhecendo que a comunicação visual é um fator-chave no processo de ensino-aprendizado do futuro arquiteto de software (PERRY & WOLF, 1992), (GARLAN & PERRY, 1995), esta pesquisa de tese investiu nas novas tecnologias emergentes de visualização 3D, Realidade Virtual e Realidade Aumentada. Recentemente, percebe-se o aumento do interesse nestas tecnologias nas aplicações de jogos e na indústria dos cinemas (PIEKARSKI & THOMAS, 2002), (WAGNER, 2005), (SECOND LIFE, 2009). Contudo, o foco desta proposta de tese está relacionado ao ensino de ES, especificamente AS. A disciplina da AS recebe um impulso destas novas opções tecnológicas através da visualização, interação e análise de sistemas de software (RODRIGUES & WERNER, 2009b). O público-alvo é formado pelos chamados nativos digitais¹. São pessoas que já passam a maior parte do tempo lidando com diversões em ambientes tridimensionais de computadores, videogames, vídeos e celulares. Por isso, a adaptação dos alunos ao ambiente virtual é rápida e eles tendem a se engajar

¹ Expressão criada em 2007 por Marc Prensky, pensador e desenvolvedor de games, o termo Nativos Digitais corresponde às pessoas que cresceram com a tecnologia digital, tais como computadores, Internet, telefones celulares e MP3. Um nativo digital experimenta novos aplicativos, tem facilidade com blogs, em lidar com múltiplos links e sites e de interagir com os outros (MONTEIRO, 2009).

mais no treinamento.

1.1. Objetivo e Questão de Pesquisa

O objetivo deste trabalho é propor uma abordagem, denominada Visar3D (Visualização de Arquitetura de Software em 3D), para mobilizar os alunos no ensino de AS, utilizando as tecnologias de Realidade Virtual e Realidade Aumentada.

Tendo em vista a evolução da motivação desta pesquisa, apresentada na Seção anterior, propõe-se a seguinte questão de pesquisa:

A Realidade Virtual e a Realidade Aumentada podem contribuir como tecnologias de apoio ao ensino de Arquitetura de Software através de uma abordagem mais efetiva e atrativa para o aluno?

Para responder esta questão de pesquisa, este trabalho propõe a elaboração de um sistema de visualização 3D, com a utilização das tecnologias de Realidade Virtual e Realidade Aumentada, para a exibição dos modelos arquiteturais numa nova perspectiva, dentro de uma abordagem que inclui a utilização de projetos mais elaborados e complexos, análogos aos desenvolvidos na indústria. A ideia é permitir a exploração e a interação do aluno de AS, utilizando os recursos e facilidades presentes neste ambiente virtual, tais como (RODRIGUES & WERNER, 2009b): acesso à documentação multimídia, no formato de áudio, imagens e vídeos, associada à AS 3D, permitindo a visualização dos modelos em diversos ângulos, e a interação ou manipulação dos mesmos em um ambiente simulado. Portanto, espera-se que a abordagem proposta possibilite um aprendizado mais efetivo, agradável e atrativo para o aluno de AS.

Durante o desenvolvimento de um projeto arquitetural é produzida uma grande quantidade de documentação que facilita a comunicação entre os stakeholders, registra as decisões iniciais acerca do projeto de alto-nível, e permite o reúso do projeto (PERRY & WOLF, 1992), (GARLAN & PERRY, 1995). Estas informações são de grande importância para o arquiteto de software e, principalmente, para o aluno de AS, no entendimento de todo o processo de criação e desenvolvimento de software. No entanto, muitas vezes estas informações não são utilizadas devidamente. Neste trabalho, pretende-se disponibilizar a documentação através de um acesso fácil e rápido, ao permitir sua visualização sobreposta à AS no ambiente 3D.

Pretende-se, com isso, propor um sistema de comunicação entre os alunos e professores, favorecendo a construção do conhecimento coletivo. E ainda, adotar este

ambiente para apoiar o ensino numa abordagem de ES já existente, disponibilizando, possivelmente, recursos de simulação para o exercício da prática. Este trabalho utiliza soluções relacionadas à AS, tais como abordagens propostas em dissertações de mestrado desenvolvidas pelo grupo de Reutilização da COPPE, como exemplo de abordagem de ensino, ou seja, aplica seus recursos e funcionalidades para atender ao seu principal objetivo que é apoiar o ensino de AS.

Para avaliar o objetivo de apoiar o ensino de uma forma atrativa ao aluno, será elaborado um plano de estudos observacionais, para verificar sua viabilidade.

O enfoque visual e lúdico proposto nesta abordagem, empregando a Realidade Virtual e a Realidade Aumentada, na disciplina de AS, trata-se de um tema atual, inovador e com importância prática no ensino de ES (RODRIGUES e WERNER, 2009a), conforme o estudo e a pesquisa realizada e apresentada nesta monografia.

1.2. Organização da Monografia

Além desta introdução, esta monografia está organizada em cinco capítulos. Os três primeiros apresentam a fundamentação teórica da pesquisa.

O Capítulo 2 descreve sobre o Ensino da Engenharia de Software, as suas novas demandas e as novas propostas de ensino introduzidas na sala de aula. Apresenta também, os conceitos associados à disciplina Arquitetura de Software.

A fim de investigar na literatura todo o material relevante sobre as práticas de ensino de Arquitetura de Software, o Capítulo 3 descreve os pontos principais de uma Revisão *Quasi*-sistemática conduzida em abril deste ano.

O Capítulo 4 aborda as tecnologias de pesquisa e desenvolvimento escolhidas para desenvolver esta proposta de tese: Realidade Virtual e Realidade Aumentada.

O Capítulo 5 apresenta a abordagem proposta VisAr3D – Visualização de Arquitetura de Software em 3D – nesta pesquisa de tese.

Por último, o Capítulo 6 resume as atividades que compreendem a pesquisa proposta, identificando aquelas já desenvolvidas, as propostas para avaliação, as contribuições esperadas, uma análise dos riscos envolvidos e, ainda, um cronograma de atividades previsto para o desenvolvimento do trabalho proposto.

2. Ensino de Engenharia de Software

BAUER (1972) define a Engenharia de Software (ES) como: "O estabelecimento e uso de sólidos princípios de engenharia, a fim de obter software de maneira econômica, que seja confiável e que funcione eficientemente em máquinas reais. É uma disciplina que se concentra na prática da produção de um sistema de *software*, enquanto estuda os fundamentos teóricos dos aspectos computacionais".

Em um curso de graduação da área Ciência de Computação, os alunos estudam aspectos técnicos como métodos e técnicas para o desenvolvimento de software, que abordam especificação, modelagem, arquiteturas, verificação e testes de software, bem como aspectos não-técnicos como planejamento e gerenciamento do processo de desenvolvimento (HUANG & DISTANTE, 2006).

Atualmente, o ensino da ES está passando por vários tipos de questionamentos, principalmente, pelos estudantes e pela indústria (SHAW, 2000), (HILBURN & TOWHIDNEJAD, 2007), (THOMPSON & EDWARDS, 2009). A academia ensina tipicamente teorias e conceitos que estão presentes numa série de leituras e aos estudantes, geralmente, é solicitado, em termos práticos, o desenvolvimento de um pequeno projeto para colocar todo este conhecimento em prática num espaço de tempo muitas vezes restrito. Apesar de estes componentes serem necessários e úteis para a educação de futuros engenheiros de software, não são suficientes para atender todo o processo de ES. Os estudantes, em face de um grande volume de documentos necessários para o bom desenvolvimento de um projeto de ES, se desinteressam e associam-no como assunto teórico (VARMA & GARG, 2005). Estes alunos parecem preferir escrever programas e ver seu código funcionando, a documentar formalmente o desenvolvimento de suas aplicações (DE LUCENA *et al.*, 2006).

Paralelo a isso, de acordo com CONN (2002), os profissionais de ES, na indústria, estão insatisfeitos com a falta de preparo dos universitários que ingressam no mercado de trabalho. A indústria se vê obrigada a complementar sua educação com treinamentos e preparações que lhes forneçam conhecimento que supram esta deficiência. Geralmente, os alunos saem da faculdade sem terem participado de um processo de ES próximo ao que vão encontrar no mercado de trabalho, que envolve: sistemas grandes e complexos; a participação em uma equipe também grande e distribuída geograficamente, com mudanças de objetivos durante o projeto; problemas com os clientes; pressão com tempo de entrega do produto; uma maior demanda por qualidade de software; e outros fatores como gerenciamento, questões de espaço de trabalho e cultura corporativa.

Segundo MEYER (2001), a academia não deve assumir toda esta responsabilidade,

pois a universidade não é uma empresa e nem deve ser. Mas deve preparar seus estudantes para os reais desafios que encontrarão, sendo este um projeto a longo prazo. Recentemente, a academia colocou muito esforço em diminuir este problema, encontrando novas maneiras de ensinar a ES, utilizando diferentes abordagens que compartilham o mesmo objetivo: diminuir a distância entre a teoria e a prática. Há um consenso que diz que a maneira como a ES é ensinada deve ser mudada para refletir esta nova demanda por desenvolvimento de software mais sofisticado (BAKER *et al.*, 2005). HUANG & DISTANTE (2006) dizem, ainda, que o ensino de ES tradicional focalizado pesadamente em metodologias de ES, não é adequado. E acrescenta que os estudantes precisam aprender tanto aspectos técnicos, quanto aqueles não-técnicos no desenvolvimento de sistemas de software. Segundo VARMA & GARG (2005), esta busca por métodos alternativos, convencionais e não-convencionais, pode fazer o ensino da ES mais efetivo e interessante, e, ainda, estreitar a distância entre a indústria e a academia.

Baseados nestes objetivos e nas demandas descritas acima, novas propostas de ensino de ES foram introduzidas na sala de aula, principalmente aquelas que tornam o ensino mais atraente para o aluno. Algumas delas estão, sucintamente, descritas a seguir.

O restante do capítulo foi organizado da seguinte forma: a Seção 2.1 mostra algumas ferramentas de apoio ao ensino de Engenharia de Software em geral. Na Seção 2.2, são descritos os principais conceitos de Arquitetura de Software, já que este será o tema explorado por esta proposta de tese. As Considerações Finais fecham o capítulo na Seção 2.3.

2.1. Ferramentas de Apoio ao Ensino de Engenharia de Software

Devido à sua finalidade, é muito importante que as instituições de ensino forneçam educação apropriada e oportunidades de pesquisa para prepararem os estudantes para esses novos desafios e oportunidades. O ensino da ES deve preparar os estudantes para participações efetivas e produtivas de uma forma colaborativa e interdisciplinar (POUR, 2006). Esta seção tem como objetivo apresentar ferramentas propostas por algumas destas instituições, que de alguma forma, tentam tornar o ensino de ES em tarefas atrativas de laboratório. Na literatura existem muitos exemplos, contudo são citados aqui apenas quatro representativos, que têm em comum a utilização de recursos de ensino, às vezes, externas ao computador, que desafiam e mobilizam os alunos em tarefas mais próximas de objetivos reais. A seguir,

serão descritos alguns projetos responsáveis pelo desenvolvimento de ferramentas de apoio ao ensino de ES. São eles: *Problems and Programmers*, Projeto Unibral, Draw-Bot e WriteOn.

***Problems and Programmers* (BAKER et al., 2005)**

Problems and Programmers é a abordagem desenvolvida por pesquisadores da Universidade da Califórnia (Irvine) para ensinar o processo de ES. Trata-se de um jogo de cartões educacional que simula o processo de ES a partir da especificação de requisitos até a entrega do produto. Ele fornece aos estudantes uma visão geral de alto nível, possibilitando a experiência prática do processo de ES, num espaço de tempo reduzido e competitivo. Cada jogador assume o papel de um gerente de projeto e deve completar o projeto antes de qualquer oponente. Além disso, a sua natureza física, ou seja, a utilização de cartões (figura 2.1) e interação face-a-face, motiva os jogadores, encorajando o aprendizado colaborativo. No jogo, as leituras são incentivadas para ensinar os conceitos e teorias fundamentais da ES.

O jogador, que completar o projeto primeiro, será o vencedor. No entanto, eles devem se preocupar com questões como o orçamento ou a demanda do cliente por segurança no software a ser produzido. Ou seja, eles devem seguir práticas de ES apropriadas para evitar qualquer consequência adversa que poderá levá-los a falhar perante o seu oponente na corrida para completar o projeto.

O jogo utiliza 85 regras de ES coletadas através de uma pesquisa na literatura. Estas regras representam as boas práticas seguidas tanto pela academia quanto pela indústria.

Problems and Programmers é um jogo interessante para o aluno, pois combina conceitos de ES com a simulação. Segundo o artigo, esta abordagem representa a primeira tentativa em ensinar o processo de engenharia de software utilizando um jogo com cartões. Sua maior contribuição é a motivação dos alunos, disponibilizada, principalmente, pela interação face-a-face.

Projeto Unibral (DE LUCENA et al., 2006)

O Projeto Unibral é resultado de uma cooperação científica e acadêmica entre a Universidade Federal do Amazonas e o Instituto de Automação Industrial e Engenharia de Software da Universidade de Stuttgart. O projeto, iniciado em 2003, incentiva a interação entre estudantes de engenharia elétrica e estudantes de ciência da computação.

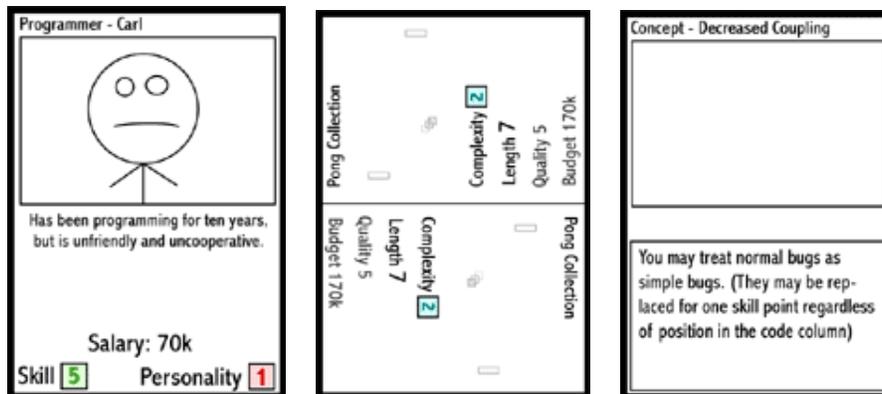


Figura 2.1. Alguns cartões do *Problems and Programmers*

A idéia do projeto é expor aos estudantes os mesmos problemas que são encontrados em situações profissionais reais. Portanto, o objetivo do laboratório é fazer com que os alunos, a partir de uma plataforma de hardware existente (um robô), desenvolvam um sistema de software que resolva um problema específico. Os alunos têm disponível o robô, seu sistema de controle e de comunicação. O desafio proposto é desenvolver uma estratégia (algoritmo) para controlar o robô, que deve partir de um ponto e se mover de maneira autônoma por um caminho com obstáculos desconhecidos, no prazo mais curto.

As fases do projeto a serem seguidas são: Definição, Projeto, Implementação, Integração e Aceitação, com documentos de garantia de qualidade para cada fase. No início do curso, os alunos recebem uma descrição textual dos requisitos dos usuários, que é incompleta e ambígua. Baseados nestes requisitos, eles devem formular cerca de 20 requisitos funcionais e 10 não-funcionais.

O Projeto Unibral é uma iniciativa importante, principalmente, por pertencer a duas universidades de culturas diferentes, garantindo o intercâmbio entre os pesquisadores para o aprimoramento do ensino da ES.

WriteOn (TRONT *et al.*, 2006)

O WriteOn é uma ferramenta apresentada por pesquisadores da empresa Virginia Tech e Microsoft. Ele permite ao instrutor utilizar tinta eletrônica para anotar sobre qualquer aplicação visível na tela. A figura 2.2 apresenta uma tela do WriteOn. Esta capacidade pode ser útil para melhorar tanto a apresentação da informação quanto melhorar a interatividade entre o estudante e o professor dentro da sala de aula. Utilizando o WriteOn, o instrutor pode optar pelo envio do conteúdo da tela dinâmica para seus alunos em tempo real. Ele apresenta ainda as seguintes funções: salvar as anotações de tinta como um arquivo de filme, distribuir estes arquivos entre os

estudantes, capturar o áudio e sincronizá-lo às anotações.

Para receber as transmissões das palestras e anotações pelo professor, os alunos devem conectar-se à sala de aula virtual. A capacidade de transmissão do WriteOn permite ao instrutor ministrar palestras envolvendo conteúdo dinâmico para os alunos.

Esta ferramenta tem um efeito essencial no processo de aprendizagem, principalmente, no ensino de ES, que necessita, cada vez mais, de suporte através de ferramentas simples, que facilitam a comunicação dinâmica entre o professor e o aluno.

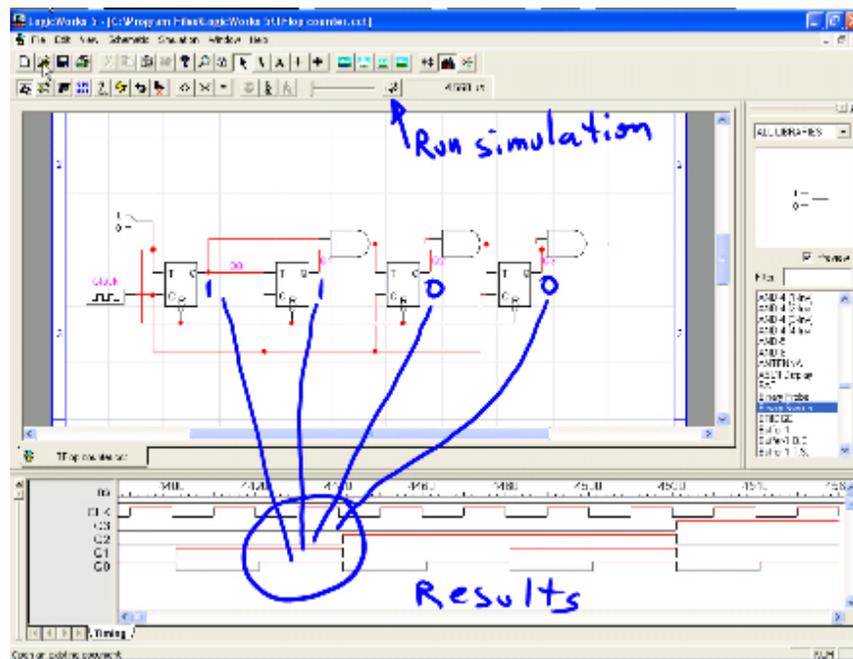


Figura 2.2. Tela do WriteOn

Draw-Bot (MOHRENSCHILDT & PETERS, 1998)

Com o objetivo de ensinar os princípios de ES como especificação de software, projeto e teste de sistemas com requisitos de segurança e em tempo real, num projeto concreto, um curso da Universidade de McMaster desenvolveu o projeto Draw-Bot. Durante o curso, os estudantes têm como objetivo controlar um robô para traçar um simples caminho através de um labirinto (no papel). O robô é construído através de um pequeno kit de robô educacional que é controlado por uma interface de software, ou seja, os alunos não devem se preocupar com aspectos de hardware para construir o robô. Eles acreditam que os seus requisitos podem e devem ser atingidos e que para isso, a quantidade de esforço requerido deve ser apropriada para que este objetivo seja cumprido no tempo restrito do curso. Segundo MOHRENSCHILDT e PETERS (1998), percebeu-se um aumento significativo do interesse pelos estudantes.

O controle de um dispositivo, neste caso o robô, para completar uma tarefa bem definida, possui um número grande de vantagens, entre eles: o entusiasmo dos estudantes e o aprendizado dos conceitos de ES num projeto real.

Tanto o DrawBot, quanto o Unibral, compartilham a ideia de que seu projeto, ao exigir o controle de um dispositivo para concluir uma tarefa clara, gera entusiasmo nos alunos e os ajuda a compreender os princípios ensinados no curso. A principal diferença entre eles é que o Unibral previa o intercâmbio de estudantes entre as universidades de Stuttgart e do Amazonas e foi planejado para forçar a cooperação entre estudantes de engenharia elétrica e ciência da computação.

2.2. Arquitetura de Software

Um dos assuntos que merece destaque no ensino de ES, a Arquitetura de Software (AS) procura sistematizar o processo de desenho de software, fornecendo uma sólida base para o reúso. À medida que os sistemas baseados em software vêm atingindo complexidade e alcance inimagináveis, e se tornando cada vez mais essenciais à sociedade e às organizações, a importância da especificação e do desenho destes sistemas vai aumentando. A necessidade da aplicação de um maior formalismo das representações estruturais de sistemas e o estudo dos possíveis benefícios alcançados e consequências diretas de sua aplicação levaram a AS ao status de disciplina, sendo uma área de pesquisa que tem recebido grande atenção (GARLAN & SHAW, 1994). LAGO e VAN VLIET (2005) citam, ainda, que a AS está se tornando um dos tópicos centrais da ES.

2.2.1. Definições

Encontram-se na literatura algumas definições relacionadas ao termo Arquitetura de Software (MENDES, 2002), (BASS *et al.*, 2003), (SEI, 2009). Segundo SHAW e GARLAN (1996), Arquitetura de Software é a descrição dos elementos a partir dos quais os sistemas são construídos (componentes), interações entre esses elementos (conectores), padrões que guiam a sua composição e restrições sobre esses padrões.

2.2.2. Benefícios

A arquitetura de software é um elemento de ligação fundamental entre os requisitos de um software e a implementação. Uma ponte que interliga um mundo não-técnico, com os *stakeholders* (o problema), e um mundo técnico, com os desenvolvedores e

designers de software (a solução) (LAGO & VAN VLIET, 2005). Ou melhor, a AS se tornou o meio principal pelo qual os requisitos são transformados em sistemas implementados que funcionam (GRISHAM *et al.*, 2007).

Apesar do fato de que a aplicação de um bom projeto arquitetural ser, imensamente, importante para a prática de ES, a prática comum, muitas vezes ainda é um projeto arquitetural informal, *ad hoc*, que não é analisado, não manutenível e artesanal (GARLAN & PERRY, 1995). Por outro lado, o projeto arquitetural de um sistema de larga escala é um fator determinante para o sucesso do sistema. A escolha de uma arquitetura não apropriada pode ter um efeito desastroso.

Uma AS reflete as grandes decisões de um projeto de software. E estas decisões são tomadas por arquitetos de software, baseadas, também, nas considerações de diferentes *stakeholders* envolvidos. Muitos profissionais da área de desenvolvimento recebem o título de arquitetos de software. Geralmente, estes possuem larga experiência na criação de software em suas organizações e bom conhecimento de tecnologias, no entanto, muitas vezes, as arquiteturas criadas pelos mesmos, não possuem uma representação adequada. O papel do arquiteto de software deve abranger competências tanto técnicas como não-técnicas. Ele deve ser o responsável por criar uma atmosfera de confiança e de motivação para se obter sucesso neste ambiente complexo, trabalhando em equipe formada por pessoas com diferentes *backgrounds* e pensamentos. Atender estas necessidades, segundo CREIGHTON e SINGER (2008), é indispensável.

A literatura (PERRY & WOLF, 1992), (GARLAN & PERRY, 1995) destaca os seguintes aspectos que justificam a importância de utilizar arquitetura de software:

- Compreensão:

A arquitetura de software simplifica a habilidade de compreender sistemas de larga escala ao apresentá-los num alto nível de abstração, gerenciando questões complexas do projeto e trazendo vantagens para todas as etapas do ciclo de vida do sistema. Para apresentar novos projetos de sistemas para outras pessoas, a arquitetura auxilia a ação entre os *stakeholders* e, também, representa a união da visão dos *stakeholders* sobre o projeto.

- Reutilização:

Um projeto arquitetural facilita o reconhecimento de estruturas comuns entre sistemas, possibilitando a reutilização ao longo de um mesmo projeto ou em projetos de outros sistemas. Ele suporta diferentes granularidades de reuso: em pequena escala, no nível de componentes e em larga escala (Subsistemas, Produtos ou Frameworks), reduzindo o esforço e melhorando a produtividade.

- Evolução:

A arquitetura de software propicia a longevidade a um sistema, permitindo a facilidade de sua evolução e de sua escalabilidade.

- **Análise:**

As descrições arquiteturas fornecem novas oportunidades para análise, gerando documentação clara e precisa e expõem vários tipos de problemas que podem não ser detectados. Elas favorecem a tomada de decisão pelos *stakeholders* sobre alternativas de projetos. A divisão do esforço de desenvolvimento dentro da equipe fica mais clara, pois com os itens dos sistemas identificados, o trabalho de desenvolvimento se torna simplificado. Ela, também, simplifica a análise de fatores como consistência, qualidade da aplicação e conformidade com as restrições previamente estabelecidas. O custo das mudanças e a identificação das possíveis alterações podem ser mais facilmente analisáveis.

- **Gerência:**

Fornecer uma base gerencial para estimativa de custo e gerência de processo.

2.2.3. Estilos Arquiteturais

As famílias de software podem possuir organizações estruturais extremamente semelhantes, ou seja, suas arquiteturas de software podem seguir alguns estilos ou padrões específicos. A observância de determinado estilo arquitetural possibilita a reaplicação de soluções conhecidas a muitos projetos. Eles são *templates* para arquiteturas concretas. A tabela 2.1 apresenta os estilos arquiteturais mais comumente usados (GARLAN & SHAW, 1994), (SHAW & GARLAN, 1996), (MENDES, 2002), que foram organizados em cinco grandes grupos.

Grupo: Sistemas de Fluxo de Dados

O estilo fluxo de dados é caracterizado por visualizar o sistema como uma série de transformações em pedaços sucessivos de entradas de dados. Estes dados entram no sistema e então fluem pelos componentes, um de cada vez até que eles sejam direcionados a algum destino final (saída ou depósito de dados).

Grupo: Arquiteturas de Call/Return

Arquiteturas de Call/Return (Chamada e Resposta, ou Fluxo Controlado) têm o objetivo de alcançar as qualidades de modificabilidade e escalabilidade. Estes são o estilo arquitetural dominante em grandes sistemas de software.

Tabela 2.1. Grupos de Estilos Arquiteturais

1	2	3	4	5
Sistemas de Fluxo de Dados	Arquiteturas de Call/Return	Componentes independentes	Sistemas centrados em dados	Máquinas Virtuais
Sequencial Batch Pipes and Filters	Programa principal e subrotinas Sistemas orientados a objetos Camadas Sistemas cliente-servidor Chamada de procedimento remoto	Sistemas Cliente-servidor Sistemas baseados em eventos	Banco de dados tradicionais <i>Blackboard</i>	Sistemas Baseados em Regras

Grupo: Componentes independentes

Arquiteturas de Componentes Independentes consistem em vários processos ou objetos independentes que se comunicam por mensagens.

Grupo: Sistemas centrados em dados

Este grupo tem como objetivo obter a integração de dados, os quais poderão ser acessados e atualizados por componentes do sistema.

Grupo: Máquinas Virtuais

As Máquinas virtuais são estilos de software que simulam alguma funcionalidade não nativa ao software e/ou hardware no qual são implementados.

Estilos Heterogêneos

Um estilo heterogêneo é aquele que consiste em uma combinação de diferentes estilos com o objetivo de unir as propriedades funcionais e/ou as propriedades não-funcionais destes em apenas um.

2.2.4. Padrões de Projeto (*Design Pattern*)

A ideia de padrões foi apresentada por Christopher Alexander em 1977 no contexto de Arquitetura (projetos de cidades e urbanismo): “Um padrão descreve um problema que ocorre repetidas vezes em nosso meio e inclui uma solução genérica para o mesmo, de tal maneira que se pode usá-la mais de um milhão de vezes, sem nunca fazê-lo de forma idêntica” (ALEXANDER *et al.*, 1977). Um padrão encerra o conhecimento de uma pessoa muito experiente em um determinado assunto de uma forma que este conhecimento pode ser transmitido para outras pessoas menos experientes. Este conceito apresenta grande importância na área de engenharia de software.

Padrão de projeto (também muito conhecido pelo termo original em inglês *Design Pattern*) é uma tecnologia onde experiências recorrentes no desenvolvimento de sistemas de software orientados a objetos são documentadas e validadas pela comunidade de padrões e que têm sido empregadas com resultados satisfatórios. Facilitam o aprendizado e utilização, uma vez que permitem uma discussão entre os desenvolvedores num maior nível de abstração, aumentando a qualidade dos softwares, e, conseqüentemente, diminuindo o seu custo de manutenção. Acarretam um vocabulário comum de desenho, facilitando além da comunicação, a documentação e o aprendizado dos sistemas de software. Os padrões de projeto oferecem soluções prontas para serem utilizadas em determinados problemas enfrentados no desenvolvimento de um software orientado a objetos. Ou seja, solucionam muitos dos problemas que os projetistas enfrentam no dia-a-dia, e de diversas maneiras diferentes.

Normalmente, existe certa confusão para diferenciar padrões de estilos arquiteturais ou esta diferença não é muito clara. No entanto, os padrões arquiteturais apresentam soluções mais concretas que os estilos, estabelecendo um conjunto de subsistemas pré-definidos, com suas responsabilidades. Os estilos arquiteturais, por sua vez, são mais abstratos (GAMMA *et al.*, 1994).

2.2.5. Propriedades Arquiteturais

As propriedades arquiteturais são descrições de como o sistema deveria se comportar (MENDES, 2002). Existem dois tipos de propriedades:

Propriedades funcionais: são aquelas realizadas pelo sistema, ou seja, as

funcionalidades do sistema. Exemplo: em um sistema cliente-servidor, o componente cliente deve realizar requisições, as quais devem ser respondidas pelo componente servidor.

Propriedades não-funcionais: são aquelas relacionadas às qualidades de um sistema. São os atributos de qualidade. Exemplo: funcionalidade, usabilidade e desempenho.

2.2.6. Linguagens de Definição de Arquitetura

As Linguagens de Definição de Arquiteturas (*Architecture Description Languages - ADL*) (MEDVIDOVIC & TAYLOR, 2000) são notações textuais que fornecem um vocabulário de elementos arquiteturais. São específicas para descrição de arquiteturas, representando expressões características estruturais e comportamentais dos sistemas.

Atualmente, existem diversas ADL's disponíveis. Cada uma delas fornece suporte para o desenvolvimento de um tipo de sistema, com base em um estilo arquitetural. Wright (WRIGHT, 2009), Darwin (DARWIN, 2009), Rapide (LUCKHAM *et al.*, 1995), Meta H (VESTAL, 1994), xADL (XADL, 2009), ACME (ACME, 2009), UniCon (UNICON, 2009) são alguns exemplos.

2.2.7. Análise de Arquitetura de Software

O principal objetivo da análise de arquitetura de software é fazer uma avaliação de quão apropriada é a sua proposta, a fim de satisfazer um conjunto de requisitos não-funcionais do sistema, ou seja, características de qualidade esperadas para o sistema. A partir desta avaliação, pode-se identificar a necessidade de refinamento.

Os principais métodos para a avaliação de arquitetura são:

- **ATAM:** *Architecture Tradeoff Analysis Method* (KAZMAN *et al.*, 2000) tem como objetivo prover uma forma de compreender as conseqüências de decisões arquiteturais em relação aos atributos de qualidade de um sistema.
- **SAAM:** *Software Architecture Analysis Method* (KAZMAN *et al.*, 1994) é uma abordagem que avalia a arquitetura de um software através da análise de como os requisitos não-funcionais e funcionais foram atendidos. Este método faz uso de cenários.

Há ainda na literatura, uma abordagem baseada em *checklist* configurável, o ArqCheck (BARCELOS, 2006), que visa a avaliação, principalmente, das características de qualidade representadas nos documentos arquiteturais em relação aos requisitos especificados para o seu projeto.

2.3. Considerações Finais

Nossa sociedade está cada vez mais dependente do software, com uma demanda grande pela sua qualidade. Esta sociedade, conectada via Internet, em busca de soluções ainda mais rápidas, requer melhores maneiras de produzir sistemas. À medida que sistemas maiores e mais complexos vão surgindo, a ES, assim como o seu ensino, torna-se vital. Estas são algumas das pressões sofridas pela educação de ES, que com o passar dos anos, tem tido sua comunidade mobilizada para atender estas demandas. Particularmente, a disciplina de AS assume um papel relevante neste cenário, incorporando a reutilização e suas vantagens e permitindo aos engenheiros de software tomar decisões sobre alternativas de projeto. Ao longo deste capítulo foram relatados alguns problemas e propostas de ensino de ES, e ainda, citadas algumas ferramentas de apoio, que tentam tornar esse aprendizado mais efetivo e interessante para os alunos. Ele apresentou, também, uma breve introdução sobre os principais conceitos ligados à disciplina de AS, que é o foco desta proposta.

Levando-se em consideração a importância da disciplina AS na preparação dos estudantes para o mercado de trabalho, o próximo capítulo descreve uma revisão *quasi-sistemática* realizada com o objetivo de buscar na literatura todas as iniciativas de ensino de AS que sejam relevantes e que caracterizem a prática de ensino desta disciplina.

3. Revisão *Quasi-Sistemática*

Antes de desenvolver o projeto de pesquisa descrito nesta proposta, que visa contribuir para o desenvolvimento de uma nova abordagem de ensino de AS, houve a necessidade de investigar na literatura, todo o material relevante sobre as práticas de ensino desta disciplina, realizadas pela comunidade acadêmica. Para alcançar resultados mais significativos, disponibilizando publicações pertinentes e que pudessem contribuir significativamente para a proposta desta pesquisa de tese, decidiu-se realizar uma revisão sistemática. Ao contrário da revisão informal da literatura, o pesquisador segue um protocolo de revisão pré-estabelecido. Uma revisão sistemática visa estabelecer um processo formal para conduzir este tipo de investigação, evitando a introdução de eventuais vieses, ou seja, tendenciosidades, erros ou desvios sistemáticos do estudo, que se distanciam da verdade. Em contraste com a revisão informal, a revisão sistemática investiga um tópico de pesquisa seguindo um protocolo bem definido e rigoroso, de forma que outros profissionais possam reproduzir este mesmo protocolo (BIOLCHINI et al., 2005).

Será adotada uma abordagem que estrutura a questão de pesquisa em 4 elementos básicos: população, intervenção, comparação e resultado (Pai *et al.*, 2004). Como o objetivo deste estudo é realizar uma caracterização da área, não haverá comparação e nem será possível a aplicação de meta-análise (método estatístico). Portanto, pode-se definir este tipo de estudo secundário, apesar de sistemático, como uma *quasi-revisão sistemática*.

Este capítulo descreve os pontos principais desta revisão, assim como seus resultados, que servem como embasamento teórico para a pesquisa desenvolvida. Em RODRIGUES e WERNER (2009a), pode-se encontrar a íntegra desta revisão *quasi-sistemática* realizada de março a abril de 2009¹.

Este capítulo é organizado da seguinte forma: na Seção 3.1, é apresentado o Processo para Realização da Revisão *Quasi-Sistemática* de Literatura. O Planejamento da Revisão é descrito na Seção 3.2 e a Seção 3.3 apresenta a Condução da Revisão. Na Seção 3.4, a Categorização das iniciativas encontradas é mostrada. A Seção 3.5 apresenta a avaliação dos resultados. E o capítulo é finalizado com as considerações finais, na Seção 3.6.

¹ Pretende-se repetir esta revisão sistemática no decorrer da pesquisa.

3.1. Processo para Realização da Revisão *Quasi-Sistemática* de Literatura

A revisão *quasi-sistemática* foi conduzida em três etapas: Planejamento da Revisão, Condução da Revisão e Publicação dos Resultados, de acordo com BIOLCHINI *et al.* (2005), descritas a seguir.

3.2. Planejamento da Revisão *Quasi-Sistemática*

3.2.1. Questão de pesquisa

Objetivo

Identificar as iniciativas, práticas ou experiências realizadas para o ensino de ES que caracterizam uma tentativa de ensinar a disciplina de AS, ou parte dela, não representando, necessariamente, um curso completo de AS.

Questão de pesquisa:

Quais iniciativas foram realizadas no ensino de Arquitetura de Software?

População:

Publicação tendo em vista a Arquitetura de Software

Intervenção:

Ensino de Arquitetura de Software

Resultados:

Iniciativas identificadas

3.2.2. Fontes para busca, termos e sinônimos

Fontes:

Pesquisa nas seguintes bases de dados eletrônicas:

Compendex (em modo Expert Search): <<http://www.engineeringvillage2.org/>>

ACM (em modo Advanced Search): <<http://portal.acm.org/dl.cfm>>

Elsevier (em modo Search do Science Direct): <<http://www.sciencedirect.com/>>

IEEE (em modo Advanced Search): <<http://ieeexplore.ieee.org>>

Springer: <<http://www.springerlink.com>>

Scopus (em modo Advanced Search): <<http://www.scopus.com>>

Termos e sinônimos utilizados na pesquisa:

Software Architecture –	architecture design, architectural representation, architectural model;
Education –	educational, training, teaching, learning, mentoring, course;
Initiative –	experience, best practices, benefit, guideline, tool, method, technique, curriculum.

3.2.3. Strings de busca

A string de busca utilizada nesta Revisão *quasi*-Sistemática foi a seguinte:

("software architecture" **OR** "architecture design" **OR** "architectural representation" **OR** "architectural model")

AND

(education **OR** educational **OR** training **OR** learning **OR** teaching **OR** mentoring **OR** course)

AND

(initiative **OR** experience **OR** "best practices" **OR** benefit **OR** guideline **OR** tool **OR** method **OR** technique **OR** curriculum **OR** experiences **OR** initiatives **OR** benefits **OR** guidelines **OR** tools **OR** methods **OR** techniques **OR** curricula)

3.2.4. Critérios para inclusão e exclusão de estudos

Os critérios definidos para inclusão e exclusão de estudo foram:

- Os documentos devem estar disponíveis na Web;
- Os estudos devem ser sobre iniciativas que foram realizadas, especificamente, no ensino de AS, ou parte dela, não representando, necessariamente, um curso completo da disciplina;
- Os artigos devem apresentar palavras da *string* de busca no seu título ou no seu resumo;
- Os artigos devem ser escritos em inglês;
- Foram excluídos artigos que apresentavam iniciativas de ensino de engenharia de software, sem o enfoque devido à AS.

3.2.5. Processo de seleção dos estudos

1. O pesquisador executa a busca nas fontes selecionadas utilizando a string de busca elaborada.
2. Os artigos retornados pela busca são inseridos na ferramenta JabRef (JABREF, 2009).
3. O conjunto de artigos é selecionado a partir da verificação dos critérios de inclusão e exclusão. Esta verificação se dará pela leitura do resumo e do título do artigo.
4. Os artigos incluídos e excluídos são documentados no Formulário de Seleção de Estudos, apresentado em (RODRIGUES & WERNER, 2009a).

Apesar de todo o protocolo formal, todo o critério empenhado e as iterações realizadas, esta revisão *quasi*-sistemática foi feita por apenas um pesquisador. Mesmo correndo o risco de aumentar o viés do estudo, ela representa um material valioso, como embasamento teórico, para a proposta de tese.

3.2.6. Avaliação da qualidade dos estudos

Procedimentos explícitos para avaliação da qualidade do material não foram preparados. A revisão se concentrou em procurar por estudos que descrevessem iniciativas de ensino de AS. A única questão considerada é que o artigo deveria incluir uma descrição da prática de ensino, pois esta faz parte dos dados a serem extraídos. Foi considerado que as fontes dos documentos são confiáveis, e que os textos tenham passado por revisões externas que serviram de filtragem para que os mesmos tenham qualidade suficiente para contribuir com a revisão *quasi*-sistemática.

3.3. Condução da Revisão

3.3.1. Análise dos documentos recuperados

A ferramenta JabRef version 2.4.2 (JABREF, 2009) foi o gerenciador de referências utilizado para manipular as publicações recuperadas pelas máquinas de busca.

Durante a etapa de Condução da Revisão, as fontes para a revisão sistemática são selecionadas, os estudos primários são identificados, selecionados e avaliados de acordo com os critérios de inclusão e de exclusão e de qualidades estabelecidos durante o protocolo da revisão (MAFRA & TRAVASSOS, 2006).

As *strings* de busca preparadas foram executadas nas respectivas máquinas de

busca das editoras selecionadas, como fontes no protocolo (IEEE, Elsevier, ACM, Springer e Compendex). No entanto, algumas máquinas de busca apresentaram limitações que impediram uma correta execução das *strings*. A máquina de busca da biblioteca digital da ACM foi descartada, pois impossibilitou a execução da seleção conforme o protocolo definido.

3.3.2. Resultado das Buscas nas Bibliotecas Digitais

A tabela 3.1 expõe a quantidade de referências recuperadas de acordo com as máquinas de busca utilizadas.

Tabela 3.1. Artigos retornados nas respectivas máquinas de busca

Maquinas de busca	Artigos retornados
Compendex	436
Elsevier	21
IEEE	124
Scopus	430
Spring	12
Total	479

3.3.3. Análise dos documentos recuperados

Numa primeira avaliação superficial, foi feita a exclusão das referências sem disponibilidade de acesso pela *Web* e dos artigos repetidos acessados por máquinas de busca diferentes. A nova situação quantitativa resultou em 479 artigos. Posteriormente, em uma avaliação mais apurada e detalhada, foram selecionados os documentos candidatos a fazer parte da revisão *quasi*-sistemática. Foram excluídas as referências que nitidamente tratavam de outros assuntos não pertinentes à pesquisa. Finalmente, a seleção quantitativa ficou em 28 artigos selecionados.

3.3.4. Extração da informação

Após a seleção dos estudos, os dados dos mesmos são extraídos e sintetizados para serem finalmente publicados durante a etapa de Publicação dos Resultados

(MAFRA & TRAVASSOS, 2006).

3.4. Categorização das iniciativas encontradas

Posteriormente à execução da seleção dos resultados da revisão *quasi*-sistemática, as iniciativas encontradas foram classificadas de acordo com critérios propostos para facilitar a sua análise. Os critérios propostos foram:

- Curso (quantidade): aponta se a iniciativa proposta é apresentada no formato de um ou mais cursos.
- Projeto em larga escala (sim/não): informa se o enfoque do ensino é utilizar projetos em larga escala ou não.
- *Design Pattern* (sim/não): informa se é utilizado *Design Pattern* (COPLIEN, 1995) no curso.
- Estudo ativo (tipo de estudo ativo): aponta o enfoque dado às atividades dos alunos no processo de assimilação do conhecimento e habilidades, como a conversação dirigida, a discussão, o estudo dirigido individual e em grupo, os exercícios etc. (LIBÂNEO, 1990).
- Abordagem utilizada (tipo da abordagem): mostra a abordagem utilizada para o ensino de AS, que pode ser uma ferramenta, um projeto etc.

3.5. Discussão dos resultados

Conforme dito anteriormente, durante a sua condução, foram obtidos 479 registros. Após a execução da seleção dos resultados, feitas as devidas filtragens baseadas nos critérios definidos para inclusão e exclusão de estudos, os dados foram extraídos de 28 registros selecionados. Com este levantamento sistemático, percebeu-se que a academia já está se preparando para entrar em sintonia com as necessidades do mercado de trabalho, oferecendo cursos com projetos em larga escala. Apesar de ser predominante a utilização da sala de aula com o auxílio de material teórico e alguma prática. Iniciativas como as de (DIKEL *et al.*, 1997), (GRISHAM *et al.*, 2007), (HOFMEISTER *et al.*, 2005), (KAZMAN *et al.*, 1996), (MAENNISTOE *et al.*, 2008), (SCHAUER & KELLER, 1998) e (WANG *et al.*, 2007) estão em sintonia com as idéias de CONN (2002), preparando os futuros profissionais de Engenharia de Software para o mercado de trabalho. Alguns artigos utilizam o recurso de estudo de casos para ensinar requisitos não funcionais em projetos reais.

Outro destaque foi que muitas iniciativas sugerem a utilização de Padrões de Projeto (*Design Pattern*), com o objetivo de atingir qualidade de software em diferentes

níveis de abstração, especialmente nos quesitos reutilização, modularidade e flexibilidade. Os artigos explicam que *Design Pattern* tem recebido muita atenção recentemente e é um tópico importante que tem sido usado de forma eficiente no ensino. Os artigos mencionam que os alunos têm problemas para aplicar o conhecimento teórico em projetos concretos e utilizando a abordagem de utilizar *Design Patterns* em projetos contínuos, diminui este problema. Os artigos que deram destaque ao *Design Patterns* foram (WANG *et al.*, 2007), (AL-TAHAT *et al.*, 2001), (DENZLER & GRUNTZ, 2008), (GAST, 2008), (SCHAUER & KELLER, 1998), (NAVEDA, 1999) e (LAGO & VAN VLIET, 2005).

Alguns artigos compartilham a preocupação da monitoração mais presente por parte do instrutor. E outros voltados para o incentivo à comunicação dentro da equipe e o aprendizado cooperativo. Muitos deles utilizaram recursos que incentivam a mudança de papéis dentro do grupo e uma abordagem atrativa ao aluno. Segundo HUANG e DISTANTE (2006), os estudantes precisam aprender tanto aspectos técnicos, quanto aqueles não-técnicos no desenvolvimento de sistemas de software e os artigos selecionados enfocam a importância destes estudos ativos.

CHENOWETH *et al.* (2007) descrevem um curso de AS que prioriza a mudança de papéis dentro das equipes: clientes, arquitetos e desenvolvedores. Este curso mostra os benefícios do aprendizado cooperativo enquanto os alunos são expostos a estes três tipos de perspectivas diferentes. O aprendizado cooperativo é um método próprio para dar este tipo de experiência porque suporta um tipo de trabalho em grupo empregado por arquitetos de software e projetistas. Este método enfatiza competências de grupos pequenos, a prática da comunicação face-a-face e reconhecer a responsabilidade individual para o sucesso do grupo. CREIGHTON e SINGER (2008) confirmam esta abordagem e acrescentam a necessidade de aumentar a percepção das forças e fraquezas individuais.

O artigo produzido por DIKEL *et al.* (1997) descreve os princípios críticos de sucesso de AS, baseado em seus estudos: focar na simplificação, minimização e clareza; adaptar a arquitetura para as necessidades de futuros clientes, tecnologia, competição e objetivos de negócio; estabelecer um ritmo de arquitetura consistente e difundida e lançamento de produto que ajudem a coordenar as ações e expectativas de todas as partes; parceria com os *stakeholders*; manter uma visão da arquitetura clara; e fazer gerenciamento de riscos e oportunidades.

Num ciclo de aprendizado desta disciplina, os estudantes devem criar uma AS, discutir sua solução com outros estudantes e podem descobrir que a sua solução ainda não esteja satisfatória. Segundo BOER *et al.* (2009), isto estimula os alunos a refletirem sobre o *feedback* obtido e encontrar alternativas.

As iniciativas identificadas utilizavam abordagens que ensinam a partir de métodos novos de ensino e de análise de AS, do desenvolvimento de sistemas, de editores de AS, de ferramentas desenvolvidas, da utilização de livros técnicos e de metodologias diversas. Cada relato contribui com as lições aprendidas com a utilização daquela ferramenta, metodologia ou estratégia.

BUCCI *et al.* (1998) seguem a seguinte posição: “É possível ensinar questões de nível arquitetural o mais cedo possível nos cursos de Ciência da Computação. Mas, para o sucesso desta abordagem, devem estar disponíveis ferramentas apropriadas para ajudar os alunos a construir os seus modelos”. Para isso, eles desejam construir um editor que ajude os estudantes nos seus modelos mentais iniciais que seja sofisticado o bastante para envolver noções de alto nível relacionadas à AS e não à linguagem de programação.

O artigo escrito por LEE (2003) descreve um curso de Arquitetura de Software Web com tempos de duração diferentes. Seu objetivo é ensinar tanto fundamentos quanto conhecimento prático, para atender as necessidades da indústria.

WANG e STAELHANE (2005) utilizam um método chamado "Análise *Post Mortem*" (PMA) para suscitar pontos fortes e fracos do projeto durante sua avaliação.

O artigo de WANG e SOERENSEN (2006) apresenta o método "*writing as a tool for learning*". Este é um método simples, usado nas leituras, as quais são introduzidas pausas. Os alunos pensam no tópico e escrevem tudo que sabem sobre ele, compartilham informações com o grupo e, finalmente, comparam com o livro texto.

WANG e SCANNELL (2005) descrevem uma ferramenta de modelagem de confiabilidade que incorpora estilos arquiteturais, desenvolvida para facilitar a predição de qualidade pelos estudantes. Esta ferramenta fornece aos estudantes *feedbacks* instantâneos e diminui as curvas de aprendizado. Com uma interface gráfica fácil de usar, ela trabalha com os estilos: *batch sequential, parallel, fault-tolerance* e *client/server*.

Como parte de um curso, é descrito em (VALLIESWARAN & MENEZES, 2007), o ArchKriti - *A Software Architecture Based Design and Evaluation Tool Suite*. Esta ferramenta dá suporte a passos importantes dentro do desenvolvimento baseado em arquitetura: requisitos dos *stakeholders*, projeto de arquitetura de um sistema, avaliação de arquitetura e documentação. Este é um projeto aberto que facilita a adição de novos estilos arquiteturais e visões.

A avaliação de AS está se tornando uma ferramenta muito importante durante os vários processos de decisão no desenvolvimento de software. Decisões estratégicas são tomadas baseadas nos resultados das avaliações de AS. As avaliações são também usadas para garantir que os objetivos de qualidade sejam atingidos. Para se

tornar um avaliador de AS eficiente, é importante ganhar experiência em conduzi-las. Os artigos de (SVAHNBERG & MAERTENSSON, 2007), (BOER *et al.*, 2009), (WANG & STAELHANE, 2005), (WANG *et al.*, 2007) e (HOFMEISTER *et al.*, 2005) mostram experiências de cursos que dão destaque a avaliação de AS.

O *Architecture Expert* (ArchE) (MCGREGOR *et al.*, 2007) é uma ferramenta de software que serve como um assistente de arquiteto de software que ajuda a criar arquiteturas que tem níveis específicos de qualidades requeridas. O ArchE incorpora teoria de atributos de qualidade, técnicas para resolver modelos destes atributos associados a um projeto arquitetural para dar respostas para determinadas situações e a habilidade de usar projetos legados como entrada. Neste artigo, a ferramenta ArchE é descrita, produzindo arquiteturas e ensinando sobre AS.

HOFMEISTER *et al.* (2005) utilizam a Análise Global, que segundo o mesmo, serve para guiar o processo de projeto arquitetural, para capturar o raciocínio e apoiar a rastreabilidade entre a fase de requisitos e a AS.

Os resultados representam um indicador da necessidade de aprofundar o entendimento da prática e de todas as competências ligadas à disciplina de AS, visando a formação de bons arquitetos. Foi observada, a importância da prática no processo de aprendizado da disciplina. Os especialistas demonstraram que ao ensinarmos, somente métodos e técnicas, os conhecimentos e as competências dos estudantes tornam-se frágeis. E ressaltaram, também, a necessidade de ser oferecida, a oportunidade dos estudantes projetarem sistemas mais complexos antes de saírem da faculdade, atendendo à demanda da indústria.

3.6. Considerações finais

Levando-se em consideração a importância da AS na preparação dos estudantes para a prática do mercado de trabalho, foi conduzida uma revisão *quasi*-sistemática sobre as iniciativas realizadas pela comunidade científica para o ensino de ES que caracterizam uma tentativa de ensinar esta disciplina ou parte dela, disponibilizando informações sobre a prática do ensino, como o formato dos cursos, a abrangência dos projetos dos alunos, o estudo ativo e a abordagem de ensino utilizada. Este capítulo descreveu como a revisão foi organizada e conduzida, e apresentou os resultados de acordo com critérios determinados.

As contribuições foram significativas para o tema proposto nesta monografia com sugestões de como lidar com projetos maiores; de utilização de mais prática aliada aos tópicos teóricos; e de necessidade de investimento no trabalho em equipe, na discussão em grupo e no apoio do instrutor.

Devido à característica deste tipo de revisão da literatura, esta revisão *quasi-sistemática* permite a reprodução deste mesmo protocolo numa outra etapa do trabalho de pesquisa, com a intenção de obter resultados mais atuais.

O próximo capítulo aborda as tecnologias de pesquisa e desenvolvimento escolhidas para apoiar o ensino de AS na abordagem de pesquisa proposta, a Realidade Virtual e Aumentada.

4. Realidade Virtual e Aumentada

Em algum momento, num programa de televisão, num noticiário ou num filme de ficção científica, o espectador deve ter visto uma pessoa de capacete experimentando sensações de um mundo simulado artificialmente pelo computador. Esta tecnologia que tenta recriar ao máximo a sensação da realidade para um indivíduo através de interações em tempo real com o uso de técnicas e de equipamentos computacionais é chamada de Realidade Virtual (RV). A Realidade Aumentada (RA), por sua vez, tem como base o mundo real, colocando sobre ele informações ou objetos virtuais. Ela pode ser reconhecida em jogos de futebol exibidos pela televisão, quando uma seta ou um círculo marca a distância entre a bola e o gol, ou entre a bola e os outros jogadores numa cobrança de falta. Ela também é utilizada nos sistemas de posicionamento global, ou GPS (*Global Positioning System*), utilizados nos carros, dando sugestões de rotas de um ponto a outro na cidade, indicações de nomes de ruas, acessos alternativos, distância até uma curva, velocidade etc.

A RV transporta o usuário para o ambiente virtual, diferentemente da RA, que mantém o usuário no seu ambiente físico e transporta o ambiente virtual para o espaço do usuário, permitindo a interação com o mundo virtual, de maneira mais natural e sem necessidade de treinamento ou adaptação (KIRNER & TORY, 2006).

Ao perceber a necessidade de inovar, quando a questão é mobilizar os alunos e ensinar ou apoiar o ensino de ES, este capítulo aborda a RV e a RA, tecnologias de pesquisa e desenvolvimento escolhidas para desempenhar este desafio. Ele forma um embasamento teórico necessário para a compreensão da proposta de tese de doutorado apresentada com mais detalhes no Capítulo 6. Na Seção 4.1, é apresentado um breve histórico sobre RV e RA. Na Seção 4.2, são abordadas algumas definições envolvidas com o assunto. Na Seção 4.3, são apresentados os principais dispositivos de interação utilizados nestes sistemas. A Seção 4.4 mostra alguns exemplos das áreas de aplicações. A Seção 4.5 destaca os primeiros estudos que utilizam estes sistemas visando o desenvolvimento desta proposta de tese. Na Seção 4.6, são apresentadas as considerações finais sobre este capítulo.

4.1. Um Breve Histórico

Com sua origem na década de 60, a RV surge com um primeiro experimento

realizado por Ivan Sutherland (SUTHERLAND, 1968) que utilizou um capacete com visor transparente para visualizar imagens 3D desenvolvidas por ele. Contudo, só ganhou forças na década de 90, com o avanço tecnológico. Foi nesta época que apareceu a RA.

Com a evolução das tecnologias de hardware e o avanço das capacidades gráficas, a RV surgiu como uma nova forma de interface homem-máquina, possibilitando a navegação, imersão e interação em ambientes tridimensionais. E a RA, como uma evolução desta tecnologia, diminuiu a linha que divide o que é real do que é gerado por computador, “aumentando” o que você vê, ouve, sente ou cheira.

4.2. Definições

A fim de compreender o que é RV e RA, é importante definir alguns conceitos relativos importantes.

A **Realidade Mista** ou **Misturada** (*Mixed Reality* ou RM): A Realidade Misturada envolve a união do mundo real com o mundo virtual em algum lugar na “continuidade virtual” (MILGRAM & KISHINO, 1994). A RM, como pode ser visto na figura 4.1, conecta completamente o ambiente real e o ambiente virtual. Ela é uma tecnologia que mistura o mundo virtual, como gráficos e/ou imagens geradas por computador, e o mundo real de forma mais natural possível, em tempo real. A RM incorpora elementos virtuais ao ambiente real ou leva elementos reais ao ambiente virtual, complementando os ambientes. Sua meta é criar um ambiente tão realista que faça com que o usuário não perceba a diferença entre os elementos virtuais e os reais participantes da cena, tratando-os como uma coisa só (KIRNER & TORY, 2006).

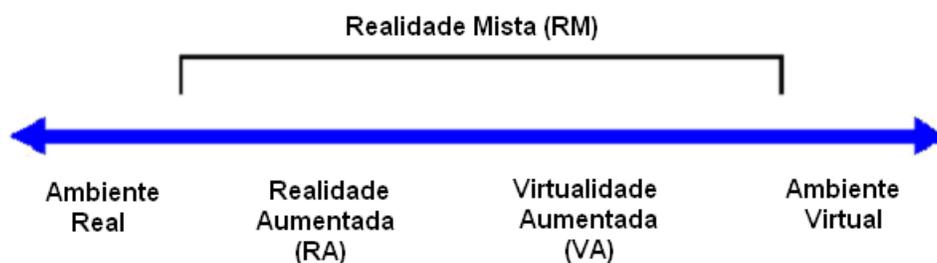


Figura 4.1. Representação simplificada da continuidade virtual (MILGRAM & KISHINO, 1994)

A **Realidade Virtual** (*Virtual Reality* ou RV) pode ser explicada como uma simulação em terceira dimensão do mundo real ou de um mundo imaginário qualquer. Esta

simulação é mais imersiva, ou seja, através de dispositivos, o usuário tem a impressão de estar em outro mundo. Aplicações de RV propiciam a visualização, movimentação e interação do usuário, em tempo real, em ambientes tridimensionais gerados por computador. Além da visão, outras sensações como o tato e a audição podem ser usadas para enriquecer a experiência do usuário. Enquanto imerso, o usuário não pode ver o mundo real em torno dele (MILGRAM & KISHINO, 1994).

A **Realidade Aumentada** (*Augmented Reality* ou RA) é uma tecnologia nova e emergente, uma evolução da RV. Segundo AZUMA *et al.* (2001), ela suplementa o mundo real com objetos virtuais que parecem coexistir no mesmo espaço do mundo real, através de algum dispositivo tecnológico. A RA pode adicionar gráficos, sons, tato e cheiro ao mundo natural. Como o nome já diz, a RA aumenta o ambiente para o usuário, sendo capaz de ampliar sua percepção e a sua interação com o mundo real. Ela enriquece o ambiente real com informações virtuais que ajudarão no desempenho de suas tarefas. AZUMA *et al.* (2001) também definiu algumas características de um sistema para que o mesmo seja considerado de RA: “combina objetos reais e virtuais num ambiente real”; “opera interativamente, e em tempo real”, e “registra (alinha) objetos reais e virtuais, uns com os outros”. As figuras 4.2a e 4.2b ilustram exemplos de RA.



Figura 4.2. RA na transmissão de um jogo de futebol

A RA e a RV podem ser comparadas da seguinte forma:

- Em relação à característica de imersão no sistema, a RA precisa estar em contato com a visão do mundo real, inserindo objetos virtuais em todos os momentos. Já a RV, através da imersão total, faz o isolamento completo da realidade, ou seja, trabalha, unicamente com o mundo virtual.

- A RV produz um ambiente que tenta fazer a substituição do mundo real, enquanto que as aplicações de RA não possuem esta necessidade. Em alguns casos, os objetos virtuais podem ser até monocromáticos.
- Na RA, combinar objetos reais e virtuais exige que estes sejam precisamente posicionados em relação ao mundo real, ou seja, a questão de registro (alinhamento) é crucial. Já na RV, toda cena é construída artificialmente e o posicionamento dos objetos é bem conhecido.

Na **Virtualidade Aumentada** (*Augmented Virtuality* ou VA), ao contrário da RA, o usuário está imerso em um mundo completamente virtual onde serão aplicadas texturas do mundo real em alguns objetos gráficos 3D. A VA é uma simulação imitando o mundo real para apoiar uma experiência imersiva, tal qual a “Matrix” do filme homônimo (BASTOS, 2005).

No ambiente da realidade misturada, a RA ocorre quando objetos virtuais são colocados no mundo real. A interface do usuário é aquela que ele usa no ambiente real, adaptada para visualizar e manipular os objetos virtuais colocados no seu espaço. A VA ocorre, quando elementos reais são inseridos no mundo virtual. A interface do usuário é aquela que transporta o usuário para o ambiente virtual, mesmo que ele veja ou manipule elementos reais ali inseridos (KIRNER & TORY, 2006).

4.3. Dispositivos utilizados

A interação é fundamental num sistema de RV, que deve dar ao usuário a sensação de presença no ambiente virtual. Dependendo dos estímulos providos por este ambiente, em tempo real, a reação do usuário pode ser diferente. A ele, é permitida a movimentação e a manipulação de objetos. Para proporcionar estas sensações, é necessária uma interface entre o usuário e o ambiente virtual que deve ser capaz de compreender estes estímulos, passá-los para a forma de sinais digitais e fornecer uma resposta perceptível para o usuário.

A seguir, são apresentados os principais dispositivos utilizados tanto na RV, quanto na RA, que oferecem maneiras mais intuitivas de interação ou que aumentam o nível de imersão do usuário. É importante observar que a cada dia, novos dispositivos são constantemente desenvolvidos, com o objetivo de oferecer modos mais intuitivos de interação. E a partir dos anos 2000, eles ficaram mais acessíveis, pelos melhores índices

de custo-benefício.

4.3.1. Sistemas de Exibição

Os equipamentos de visualização, como os HMD (*Head-Mounted Displays*), são tecnologias tradicionais de saída para aplicações de RV e RA. O **HMD baseado em vídeo** (figura 4.3) não permite nenhuma visão direta do mundo real e é o mais usado em RV, fornecendo imagens reais através de câmeras de vídeo acopladas no exterior deste equipamento. Na RA, a imagem projetada ao usuário é mostrada em tempo real e a imagem virtual é sobreposta ao vídeo (AZUMA, 1997).



Figura 4.3. HMD baseado em vídeo

Os **HMDs com tecnologia óptica** são capacetes ou óculos de RV para apresentar o ambiente virtual diretamente sobre o mundo real em sistemas de RA. Diferentes dos HMDs baseados em vídeo, eles permitem uma visão direta do mundo real. Eles funcionam com uma ou duas câmeras posicionadas na cabeça do usuário, provendo as imagens do mundo real. O vídeo que advém dessas câmeras é, então, combinado com as imagens gráficas que foram criadas pelo gerador de cenas, combinando assim real e virtual. Pode ser utilizado em aplicações nos quais se necessita de uma visão ampla, como, por exemplo, em simuladores de torre de controle de tráfego aéreo e em aplicações como sistema de cirurgia guiada do MIT, focado em cirurgia cerebral (GRIMSON *et al.*, 1997).

O **Sistema Baseado em Monitor** pode utilizar monitores convencionais de computador (*desktop*) ou portáteis (*Palm* ou *Pocket PC*) como dispositivo de apresentação. Esse tipo de esquema é o mais simples de ser construído e não fornece um alto nível de imersão, pois em sua configuração mais convencional exige apenas um monitor e uma câmera como equipamento necessário. As imagens capturadas pela câmera são alinhadas aos objetos virtuais.

A **estereoscopia** acrescenta a dimensão de profundidade às telas de projeção dos mundos virtuais, tornando-os mais próximos e realistas da forma que os usuários os vêem no mundo real. A visualização de imagens 3D é obtida através de duas imagens diferentes que são geradas, uma para o olho esquerdo e outra para o olho direito. Existe um ângulo entre estas imagens que varia em torno de 6,5 cm, equivalente à distância que separa os olhos humanos. Há várias técnicas de estereoscopia, entre elas, a Técnica de Polarização da Luz. Nesta técnica, dois projetores comuns são usados para projetar imagens diferentes para os olhos esquerdo e direito. Filtros de luz com polarização oposta são colocados sobre a lente de cada projetor. Cada usuário deve usar óculos com as lentes polarizadas de forma correspondente, de modo que cada olho enxergará somente a imagem correta. Uma tela de projeção prateada (ou aluminizada) é necessária para preservar a polarização da luz (SISCOOTTO *et al.*, 2006).

O dispositivo de projeção de imagens diretamente na retina ou **Dispositivo Virtual de Retina** (VRD) é um sistema de exibição mais interessante e simples do ponto de vista do usuário. Ele utiliza varreduras moduladas com feixes de *laser* de baixa potência, para criar a percepção da imagem virtual, que é desenhada diretamente na retina do usuário. O modo pelo qual se enxerga é o seguinte: um *laser* de diodo vermelho provê um *display* monocromático. Um *laser* de gás argônio produz linhas verdes e azuis. E os dois combinados provêem a criação de um *display full color* (BASTOS, 2005).

4.3.2. Dispositivos Hápticos

Um dispositivo háptico é aquele que envolve sensação tátil ao usuário quando em contato com objetos virtuais. Estes dispositivos podem ser usados por pessoas com deficiências ou pessoas que aprendem melhor através de experiências táteis ou sinestésicas¹(BASTOS, 2005).

¹ Sinestesia é a relação subjetiva que se estabelece espontaneamente entre uma percepção e outra que pertença ao domínio de um sentido diferente (Aurélio, 1999).

Como exemplo de dispositivo háptico, as **luvas de dados** (*data gloves*) são equipadas com sensores que sentem os movimentos dos dedos da mão e fazem a interface desses movimentos com o computador, permitindo o usuário manipular objetos ou realizar movimentos em um ambiente virtual.

4.3.3. Dispositivos Móveis

São os dispositivos menores e mais ergonômicos, como celulares e computadores de mão, como *Pocket PC*, *tablet PC* e PDA. Estes dispositivos têm uma boa relação entre poder de processamento, tamanho e peso, além de serem fáceis de manipular e possuírem conectividade sem fios.

Outra solução seria não utilizar dispositivo, ou melhor, utilizar a própria mão. Estudos incentivam a utilização deste tipo de interação (interação via gestos) muito mais intuitivo nos ambientes virtuais. E ainda, os usuários com necessidades especiais poderiam se beneficiar com este tipo de interação. A mão é gravada por uma câmera de vídeo e digitalizada por um hardware especial. Um software de reconhecimento de imagem usa os dados da imagem para determinar a posição atual e a forma ("gesto") da mão. Esta informação é então usada para conduzir a aplicação (PAVLOVIC *et al.*, 1997).

4.4. Áreas de aplicações

É imenso o potencial da RV e da RA e, nesta seção, são mostrados alguns exemplos de áreas de aplicações.

Manutenção

Com o intuito de ajudar em reparos e manutenções de um modo geral, pesquisas têm sido feitas para o desenvolvimento de aplicações. Para facilitar o entendimento das instruções, ao invés de ler manuais e observar figuras, objetos 3D podem ser sobrepostos a um equipamento qualquer, mostrando passo a passo as tarefas que devem ser feitas e como fazê-las. Estes objetos podem ainda ser animados para que os passos sejam mostrados de uma maneira mais explícita. Algumas aplicações, já existentes, consistem na manutenção de uma impressora a laser, e ainda na manutenção de um encanamento industrial (figura 4.4), onde são visualizados um mapa 2D e um modelo 3D (AZUMA *et al.*, 2001).

Militar

Desde os primeiros momentos na história da RV, as forças do exército dos Estados Unidos foram um fator importante no desenvolvimento e na aplicação de novas tecnologias de RV. Junto com a indústria de entretenimento, a militar é responsável pela evolução mais drástica no campo da RV. Os simuladores de vôo são as aplicações mais utilizadas, treinando os soldados como voar em batalha, como se restabelecer em caso de emergência ou como coordenar a sustentação no ar com operações terrestres. Os simuladores de veículos terrestres não tão precisos quanto os simuladores de vôo, são uma parte importante da estratégia do exército. Existem ainda cenários virtuais realistas de imersão que ajudam os soldados a se adaptarem a vários ambientes e situações de combate, além da visualização do campo de batalha, essencial ao determinar as estratégias de combate em tempo real (AZUMA *et al.*, 2001).



Figura 4.4. Encanamento industrial (AZUMA *et al.*, 2001)

Visualização e Explicação

Uma aplicação muito usada em RA consiste em colocar pequenas notas em objetos e ambientes, notas estas que contém informações públicas e particulares. O interessante nesse tipo de aplicação é a ajuda em tarefas cotidianas, como guiar o usuário através de um prédio desconhecido, mostrando uma variedade de sugestões para a navegação, ou como guiá-lo através de um museu, mostrando várias informações à medida que o usuário caminha.

Sistemas de RV e RA oferecem inúmeros recursos para a visualização de informações

de um grande volume de dados, uma vez que não há limite de espaço para exploração dos dados a serem representados.

Entretenimento

As aplicações na área de entretenimento são as mais diversas possíveis, desde jogos até “contadores” de histórias. Entre os jogos criados com RA, podemos destacar o ARQuake (PIEKARSKI & THOMAS, 2002). ARQuake é jogado no mundo real, dando ao usuário a mobilidade para se mover por onde desejar. Tudo que é visto é determinado exclusivamente pela orientação e posição da cabeça do usuário que está usando um HMD. Outra aplicação que mistura entretenimento e aplicação móvel é o *Invisible Train* (WAGNER, 2005). Esse jogo foi desenvolvido, primeiramente, para crianças de nível primário, e é um jogo multiusuário no qual os jogadores guiam um trem virtual através de um trilho real construído em miniatura. Este trem só é visível para os jogadores através de PDAs. Uma tendência tanto para jogos eletrônicos quanto para RV é o aparecimento de ambientes virtuais tridimensionais acessíveis pela Internet, como o *Second Life* (SECOND LIFE, 2009).

Medicina

Na área médica, as aplicações que são desenvolvidas exigem ambientes virtuais muito realísticos, onde os usuários devem ter a sensação de vivenciar aquela situação, ou há a necessidade da geração de sensações muito próximas às reais, com tempo de resposta e métodos implementados muito precisos.

A medicina tem se beneficiado bastante dos recursos tecnológicos oferecidos pela RV e RA, nos últimos anos. Estes sistemas fornecem um recurso de ensino e treinamento muito poderoso das estruturas anatômicas. A maioria das aplicações médicas é feita com a orientação de imagens cirúrgicas. Estudos de imagens no pré-operatório como, por exemplo, tomografia computadorizada, ressonância magnética e ultra-som (sensores não-invasivos) provêm ao cirurgião a visão necessária da anatomia interna do paciente e, através do estudo dessas imagens, a cirurgia é planejada. Os sistemas também podem ser usados para o treinamento de cirurgiões novatos. Instruções virtuais poderiam lembrar ao médico os passos requeridos. Entre as inúmeras vantagens das aplicações de RA, na área médica, as mais visíveis dizem respeito à fidelidade das imagens coletadas, já que as mesmas são feitas em tempo real na sala de cirurgia, aumentando com isso o desempenho de toda a equipe cirúrgica e também propiciando a eliminação da

necessidade dos procedimentos dolorosos e mais invasivos (AZUMA *et al.*, 2001). Muitos projetos estão sendo desenvolvidos para suportar a cirurgia à distância.

Planejamento Urbano

A utilização de aplicações de RA e RV permite a discussão detalhada de um projeto de arquitetura em 3D, antes mesmo da construção de uma maquete. Qualquer modificação no projeto é visualizada em tempo real. Desta forma, o projeto pode ser discutido simultaneamente por arquitetos, engenheiros, construtores e quem mais estiver envolvido em seu planejamento, desenvolvimento e implantação.

As aplicações nesta área incluem a exibição ou a recuperação de prédios e de outros elementos inexistentes ou em ruínas, enfatizando o planejamento urbano e a arqueologia. A grande vantagem em usar este tipo de tecnologia no planejamento das cidades é a facilidade de interação e visualização da visão de cada usuário, bastando para isso apenas pegar e arrastar os prédios como se os mesmos fossem simples caixas em cima de um mapa sobre a mesa (BASTOS, 2005).

Educação

A RV e RA contribuem de maneira significativa na área da educação como processo de exploração, descoberta, observação e construção de uma nova visão do conhecimento, oferecendo ao aprendiz a oportunidade de melhor compreensão do objeto de estudo. Essas tecnologias, portanto, têm potencial de colaborar no processo cognitivo do aprendiz, proporcionando não apenas a teoria, mas também a experimentação prática do conteúdo em questão (CARDOSO & LAMOUNIER, 2008).

A introdução da RA na matemática, por exemplo, pode eliminar uma das principais dificuldades do aluno: visualizar um problema complexo de geometria. Ela fornece a professores e estudantes um método intuitivo e colaborativo de aprender. Os alunos vêem os objetos tridimensionais, sem precisar imaginá-los ou desenhá-los numa folha de papel (LOPES, 2005). Quando envolvido e totalmente imerso no ambiente virtual, o usuário desenvolve um comportamento natural e intuitivo, buscando agir como agiria no mundo real e através da interação, receber resposta ideal para suas ações.

Aqui, a RV e RA não são tratadas apenas como "mais uma ferramenta" para melhorar a aprendizagem e sim, como um poderoso instrumento de aprendizagem. Segundo BELL & FOGLERL (1995), a principal vantagem de utilizar a realidade virtual é a sua capacidade de visualizar situações e conceitos que não poderiam ser vistos de outra forma, e ainda

imersão o aluno dentro dessa visualização. Segundo ele, o interesse e o entusiasmo do aluno são também evidentes benefícios da realidade virtual.

A potencialidade destas novas tecnologias está exatamente no fato de permitir a exploração de alguns ambientes, processos ou objetos, não através de livros, fotos, filmes ou aulas, mas através da manipulação e análise virtual do próprio alvo do estudo.

A introdução da RV, e conseqüentemente a RA, na educação demonstra um novo paradigma que relata uma educação de forma dinâmica, criativa, colocando o aluno no centro dos processos de aprendizagem e buscando uma formação de um ser crítico, independente e construtor de seu conhecimento. A grande preocupação é com o investimento em hardware e software, mas hoje em dia, é possível montar-se uma boa plataforma com custo, relativamente, baixo.

4.5. Recursos e Ferramentas de RV e RA

Desenvolver aplicações de RA ainda é uma tarefa desafiadora, até mesmo depois de alguns anos de pesquisa, avanço tecnológico, disponibilidade de produtos com custos acessíveis e da construção de protótipos. Vários grupos de pesquisa têm desenvolvido uma série de aplicações para RA, que permite a criação de software para explorar as possíveis interfaces da RA. Dentre eles estão: Coterie (BUTZ *et al.*, 1999), Tinmith-evo5 (PIEKARSKI & THOMAS, 2003), DWARF (BAUER *et al.*, 2003), ARToolkit (KATO, 2009) e Studierstube (FUHRMANN & PURGATHOFER, 2001) (SCHMALSTIEG, 2009). No entanto, tiveram papel fundamental os pesquisadores que disseminaram essas áreas com o desenvolvimento de recursos disponibilizados à sociedade gratuitamente. A linguagem VRML – *Virtual Reality Modeling Language* (VRML97, 2009) e a biblioteca ARToolkit são os exemplos mais marcantes de recursos gratuitos e livres, mas existem vários outros disponibilizados por pesquisadores e, mais recentemente, por empresas.

O ARToolKit foi, originalmente, desenvolvido para servir de apoio na concepção de interfaces colaborativas pelo Dr. Hirokazu Kato, na Universidade de Osaka. E, desde então, tem sido mantido pelo *Human Interface Technology Laboratory* (HIT Lab) da Universidade de Washington e pelo HIT Lab NZ da Universidade de Canterbury, em Christchurch. O ARToolKit é uma biblioteca de código aberto, escrita na linguagem C, para concepção de aplicações em RA. O pacote inclui bibliotecas de rastreamento e disponibiliza o código fonte completo, tornando possível o transporte do código para diversas plataformas, ou adaptá-lo para resolver as especificidades de suas aplicações. O

rastreamento óptico oferecido pelo ARToolkit possibilita extrair, de forma rápida, a posição e orientação de padrões marcadores, apenas com o uso de um computador e uma *webcam* convencional.

O funcionamento do ARToolKit baseia-se no uso de marcadores (cartões de papelão com um símbolo impresso e uma moldura retangular, como no exemplo da figura 4.5), permitindo o uso de técnicas de visão computacional para calcular a posição da câmera e sua orientação em relação aos marcadores, de forma a fazer com que o sistema possa sobrepor objetos virtuais sobre os marcadores. A saída, portanto, nada mais é do que uma imagem formada de um ou mais objetos virtuais, animados ou estáticos, sobre o marcador.

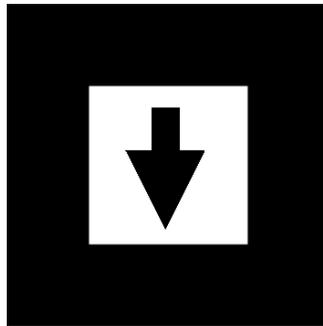


Figura 4.5. Marcador do ARToolkit

Com a RV, as formas de interface entre homem e máquina poderão ter uma enorme evolução em termos de qualidade, pelo fato dela proporcionar modos mais intuitivos dos usuários interagirem com o sistema.

A interação mais simples com os ambientes virtuais é a navegação, isto é o movimento dos usuários num espaço 3D, acarretando novos resultados em ponto de vista da cena. Neste caso, não existem mudanças no ambiente virtual, mas somente um passeio exploratório. Interações, lidando com mudanças nos ambientes virtuais, ocorrem quando o usuário entra no espaço virtual, vê, explora, manipula e desencadeia alterações aos objetos virtuais, utilizando movimentos de translação e rotação naturais do corpo humano (KIRNER & SISCOOTTO, 2007).

Um recurso muito útil em ambientes virtuais são as técnicas de visualização de informação, que, segundo FREITAS (2001), buscam representar graficamente os dados de um determinado domínio de aplicação de modo que a representação visual gerada explore a capacidade de percepção do homem e este, a partir das relações espaciais

exibidas, interprete e compreenda as informações apresentadas e, finalmente, deduza novos conhecimentos. A visualização tridimensional de informações num ambiente de RV pode facilitar a análise e a compreensão dos dados, e ao mesmo tempo despertar o interesse de alguns usuários, por conta da forma que os dados são representados graficamente, como também, pela forma de interação disponível. Este tipo de visualização permite a movimentação do usuário, possibilitando a sua exploração. Através do controle do mouse, o aluno pode se mover no espaço para a direita, esquerda, para cima, para baixo, se afastar ou se aproximar no plano e fazer movimentos de rotação para melhor visualizar a informação. Estes ambientes disponibilizam diferentes pontos de vista, onde o usuário pode se mover rapidamente para pontos de interesse que estejam distantes (animando a transição de um ponto ao outro), no entanto, pré-definidos.

A RV oferece muitos recursos para a visualização de informações, particularmente para a visualização de um grande volume de dados, uma vez que não limita o espaço de exploração dos dados a serem representados. Ela pode utilizar cores, tamanhos e formas geométricas diferentes para representar os dados e suas grandezas, de forma que facilite a manipulação e sua visualização.

A técnica de zoom, também disponibilizada, é importante porque permite focar em um subconjunto de dados, ou analisar um determinado contexto. Além disso, conforme o zoom é aplicado, mais detalhes são mostrados sobre uma determinada visão dos dados.

O VRML foi a linguagem de programação, inicialmente, utilizada nos estudos e protótipos de ambientes virtuais. A linguagem de programação VRML, abreviação de *Virtual Reality Modeling Language*, ou Linguagem para Modelagem em Realidade Virtual, surgiu da idéia de se levar Realidade Virtual para a Internet. Ela foi desenvolvida por Mark Pesce e Tony Parisi em 1994. A VRML impulsionou o desenvolvimento da RV no mundo, sendo uma das primeiras iniciativas bem sucedidas de software livre. Segundo STANEK (1996), a VRML é uma linguagem de criação de ambientes virtuais em três dimensões através de uma definição textual de objetos a serem desenhados por um *browser*. Ela permite ainda a interação entre o usuário e este ambiente. A interatividade se dá através do uso de funções básicas de navegação e também pela interferência do visitante que pode clicar sobre objetos que determinam ligações com outras cenas, ou páginas web, arquivos de texto, imagens, sons, vídeos; ou possuem comportamentos associados. Devido ao formato de arquivo, ela pode ser facilmente transportada entre diversas plataformas. Para a visualização e manipulação deste ambiente virtual, é necessária a utilização de um software de navegação para internet (*browser*) que possua um *plugin* de

reconhecimento dos códigos VRML. Este *plugin* instalado no *browser* é o encarregado de interpretar o código e gerar o ambiente descrito por ele (HEIDRICH, 2003).

4.6. Considerações finais

A RV e RA são tecnologias de pesquisa e desenvolvimento que vem dando, aos usuários, melhores condições de interação com aplicações computacionais, propiciando interações naturais e potencialização de suas características. Por isso, muitos recursos são utilizados, envolvendo hardware, software, periféricos, tecnologias especiais e o desenvolvimento de aplicações.

Para MORAN (2000), uma mudança qualitativa no processo de ensino/aprendizagem acontece quando conseguimos integrar dentro de uma visão inovadora todas as tecnologias: as telemáticas, as audiovisuais, as textuais, as orais, musicais, lúdicas e corporais... É importante diversificar as formas de dar aula, de realizar atividades, de avaliar.

Estas tecnologias parecem oferecer aos educadores maneiras alternativas e inovadoras de ensinar e engajar os alunos. No entanto, segundo BRAGA (2001), estas tecnologias não podem ser vistas como uma fórmula milagrosa que irá alterar de forma definitiva o sistema de ensino. Elas estarão, sim, a serviço de alunos e professores, que continuam tendo um papel fundamental de auxiliar os alunos na sua utilização, trabalhando de forma interativa com os mesmos e aprendendo com eles. Reúnem especificidades e atributos que as tornam ideais para as múltiplas situações e contextos de pesquisa e aprendizagem, indo ao encontro das ideias desta proposta.

Este capítulo apresentou uma visão geral sobre RV e RA, que constitui um suporte tecnológico importante para o trabalho aqui relatado.

O próximo capítulo descreve a proposta de uma nova abordagem para apoiar o ensino de AS, chamada Visar3D (Visualização de Arquitetura de Software em 3D), baseada na utilização das tecnologias de Realidade Virtual e Realidade Aumentada.

5. A Abordagem VisAr3D

Este capítulo tem como objetivo apresentar a abordagem proposta VisAr3D – Visualização de Arquitetura de Software em 3D e os fatores que motivaram a sua elaboração.

O capítulo é organizado da seguinte forma: a Seção 5.1 apresenta a caracterização do problema desta pesquisa de tese. A Seção 5.2 descreve os requisitos da abordagem proposta para atendimento a este problema. A Seção 5.3 apresenta uma Visão Geral da Arquitetura da VisAr3D. A Seção 5.4 exhibe um cenário de utilização e a Seção 5.5 descreve as funcionalidades previstas para o ferramental de apoio à abordagem. A Seção 5.6 detalha os repositórios utilizados na abordagem e a Seção 5.7, os recursos necessários para sua utilização. A Seção 5.8 mostra dois exemplos de aplicação da abordagem. Por fim, a Seção 5.9 finaliza o capítulo com as considerações finais.

5.1. Caracterização do Problema

Em função da proposta inicial, a questão de pesquisa a ser respondida ao longo deste trabalho, conforme apresentado, anteriormente, é: A Realidade Virtual e a Realidade Aumentada podem contribuir como tecnologias de apoio ao ensino de Arquitetura de Software através de uma abordagem mais efetiva e atrativa para o aluno?

A partir do que foi exposto no capítulo 4, percebe-se o potencial da Realidade Virtual, assim como da Realidade Aumentada, para propiciar uma educação como processo de exploração, descoberta, observação e construção de uma nova visão do conhecimento, oferecendo ao aprendiz a oportunidade de melhor compreensão do objeto de estudo (CARDOSO & LAMOUNIER, 2008). Neste contexto, segundo os mesmos autores, representam um grande avanço no processo educacional.

Segundo BELL e FOGLERL (1995), PINHO (2000) e MEIGUINS (1999), além de diversificar a forma de lecionar e de realizar suas atividades, estas novas tecnologias podem ser utilizadas para a criação de ambientes propícios para engajar os estudantes no processo de aprendizagem. Elas exigem interação, tornando cada participante ativo dentro de um processo de visualização, e ainda possuem possibilidades de visualizações de detalhes de objetos ou de objetos que estão a distâncias muito grandes.

Apesar de ainda serem necessárias várias pesquisas, principalmente baseadas em evidências, para a comprovação do potencial destas tecnologias na área educativa, esta

abordagem procura contribuir no ensino de ES, especificamente, no apoio ao ensino de AS.

O público-alvo da VisAr3D são os alunos iniciantes da disciplina AS, que têm como objetivo tornarem-se bons arquitetos de software.

5.2. Requisitos da abordagem

No mapeamento da área, feito através de pesquisas na literatura, incluindo uma revisão sistemática (conforme visto nos capítulos 2 e 3), foram extraídos os principais requisitos necessários para esta abordagem:

- Apoiar o desenvolvimento e a participação dos alunos em projetos complexos;
- Reduzir a distância entre a teoria e a prática;
- Apoiar a utilização de aspectos não somente técnicos;
- Ser atrativo para o aluno.

A Seção 5.5 apresenta detalhes de como estes requisitos podem ser implementados na abordagem VisAr3D. A seguir, é feita uma descrição de cada um deles.

5.2.1. Apoiar o desenvolvimento e a participação dos alunos em projetos mais complexos:

Na academia, a prática dos alunos consiste, muitas vezes, no desenvolvimento de pequenos projetos num tempo restrito. Enquanto isso, os projetos encontrados no mercado de trabalho são, geralmente, sistemas maiores, com mais complexidade, com exigências como desempenho, confiabilidade, menor custo e qualidade. A abordagem proposta, para apoiar o desenvolvimento e a participação dos alunos em projetos deste tipo, deve incentivar, também, a modelagem de um sistema grande por grupos de alunos e deve dar suporte ao entendimento dessas partes, para facilitar o entendimento do todo. Além disso, pretende-se incentivar e possibilitar a reutilização ou evolução de projetos anteriores e, ainda, facilitar o acesso à documentação.

O objetivo principal deste requisito, na abordagem VisAr3D, é preparar os futuros profissionais de ES para o mercado de trabalho.

5.2.2. Reduzir a distância entre a teoria e a prática:

Aliada à teoria, o apoio à prática traz uma dinâmica maior à sala de aula e se traduz numa participação mais expressiva dos alunos. Além disso, segundo DONGSUN *et al.*

(2008), num projeto, exemplos práticos, consistentes e repetidos de um mesmo contexto devem ser introduzidos para se obter um ensino efetivo.

Para isto, esta proposta sugere a aplicação de um estudo de caso utilizando o ferramental da abordagem VisAr3D. Este estudo deve seguir o processo de desenvolvimento de um sistema de software completo, como os encontrados em um ambiente profissional, contribuindo com a experimentação prática do conteúdo em questão e na redução da distância entre a teoria e a prática.

Através do apoio à prática, a abordagem deve fornecer facilidades para o entendimento dos benefícios associados à utilização de AS (PERRY & WOLF, 1992), (GARLAN & PERRY, 1995), como: o reconhecimento de estruturas arquiteturais comuns, a tomada de decisão sobre as alternativas de projetos, a comunicação entre os *stakeholders* e uma compreensão mais clara da arquitetura.

5.2.3. Apoiar a utilização de aspectos não somente técnicos

Além de prover aspectos técnicos (próprios da área de conhecimento da engenharia de software), a abordagem deve se propor a apoiar a utilização de aspectos não-técnicos no ensino de AS, ao apoiar as atividades dos alunos no processo de assimilação do conhecimento e habilidades. Os aspectos não-técnicos são aqueles que agregam conhecimento de áreas complementares, envolvendo fatores sociais, culturais, comportamentais, lingüísticos e políticos (confiança, diferenças culturais, idioma, entre outros) (PILLATTI, 2007). A abordagem deve incentivar e prover facilidades que apoiem o trabalho em grupo, a participação do instrutor, a discussão sobre um tópico e a troca de ideias e informações entre os alunos.

Com este requisito, pretende-se contribuir para melhorar a comunicação entre os participantes do grupo, ajudar na organização e divisão de tarefas dentro da equipe, permitir a integração dos resultados individuais como atividade do grupo de trabalho e colaborar com a resolução de problemas complexos, através da cooperação.

Segundo DONGSUN *et al.* (2008), a efetiva participação do instrutor é muito importante para a eficácia do projeto de software. Ela impede que falhas encontradas na fase inicial do projeto, sejam propagadas para as demais etapas.

Por exemplo, estas funcionalidades podem colaborar com o rastreamento (ou acompanhamento) de uma atividade de forma gráfica e comentada pelo instrutor ou pelos alunos, que possa ser facilmente visualizada, contribuindo para o trabalho do grupo e o progresso da tarefa.

5.2.4. Ser atrativo para o aluno:

Recentemente, percebe-se o aumento do interesse nas tecnologias tridimensionais nas aplicações de jogos e na indústria dos cinemas. Ambos ligados ao entretenimento, oferecem diferentes formas de lidar com a percepção do usuário. Com o requisito de ser atrativo para o aluno, a abordagem se propõe a fornecer um ambiente com esta perspectiva, um ambiente exploratório, um ambiente propício a descobertas e desafios e que responda à sua interação.

Esta proposta busca atingir este requisito utilizando os recursos e facilidades presentes no ambiente virtual, disponibilizados pela RV e RA. No entanto, a confirmação de um aprendizado mais efetivo e, ainda, atrativo para o aluno só poderá ser feita após estudos experimentais.

5.3. Visão Geral da arquitetura

Diante do contexto exposto, a abordagem VisAr3D se propõe a atender os requisitos listados na Seção anterior, para ensinar AS, a partir dos recursos disponíveis nos ambientes de visualização 3D oferecidos pela RV e RA.

Uma visão geral da arquitetura da VisAr3D, mostrada na figura 5.1, é composta de três grandes módulos:

Módulo Arquitetural: A abordagem proposta nesta monografia deve apresentar a possibilidade de integração com um ambiente de apoio à modelagem arquitetural. Neste ambiente, professor e alunos editam a AS e inserem informações (documentação) sobre a mesma. Estas informações, associadas aos diagramas, estarão disponíveis na ferramenta educativa da abordagem VisAr3D.

Módulo de Realidade Aumentada: Este módulo utiliza a RA como apoio à aula prática de AS, ajudando o professor e os alunos na identificação e acesso rápido à arquitetura exposta pelo professor.

Módulo de Realidade Virtual: Após o reconhecimento da AS exposta em 2D, o módulo de RV representará esta arquitetura em 3D. É através deste ambiente tridimensional que se fará o apoio ao ensino de AS.

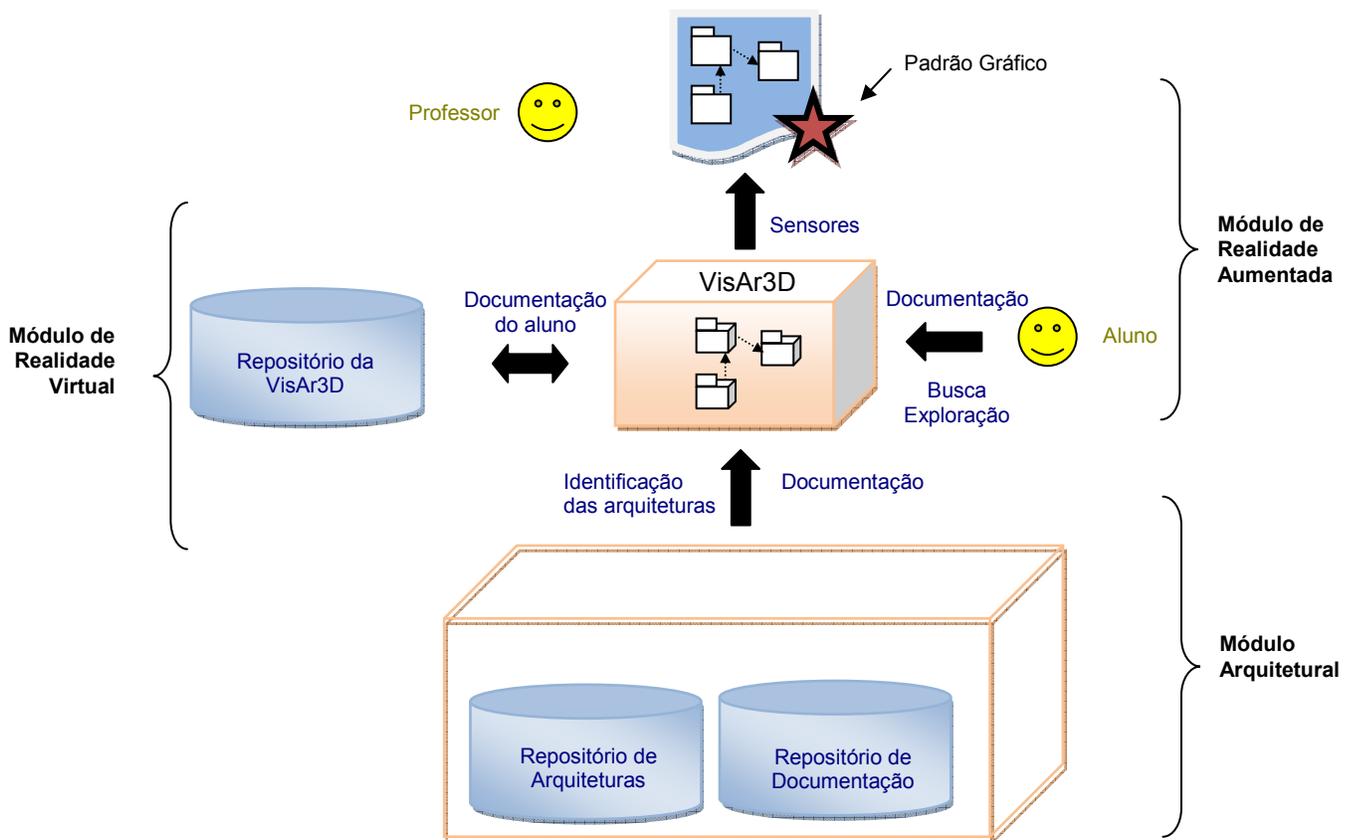


Figura 5.1. Visão geral da abordagem VisAr3D

5.4. Cenário de utilização

A abordagem VisAr3D proposta pode ser explicada por meio de um cenário de utilização. Um professor de ES exibe a modelagem de uma arquitetura de software (Módulo Arquitetural) na parede (figura 5.2) ou na tela, projetada por um editor, num cartaz exposto ou numa folha de papel impressa (em 2D). Parte de um diagrama de um sistema grande e complexo é apresentada, detalhando as funcionalidades, componentes e conectores. Seus alunos, com dispositivos móveis em mãos (PDA ou *Tablet PC*), utilizam o ferramental de apoio à abordagem VisAr3D para identificar esta AS (Módulo de Realidade Aumentada) e para disponibilizar informações sobrepostas a ela, agora exibida em 3D na tela do seu equipamento (figura 5.3). A VisAr3D torna-se a partir de agora, um ambiente de estudo da disciplina AS com possibilidades de: exploração do sistema maior que contém os diagramas dos alunos; de comunicação entre professor e alunos; de troca

de informações e documentação entre colegas; e de prática da disciplina (Módulo de Realidade Virtual).

5.5. Ferramental de Apoio à abordagem VisAr3D

Esta seção apresenta as funcionalidades previstas para o ferramental de apoio à abordagem VisAr3D, organizadas nos módulos de sua arquitetura, brevemente apresentada na Seção 5.2. Nesta seção, será utilizado o termo “VisAr3D” como referência à ferramenta de apoio à abordagem VisAr3D, por motivo de simplificação.

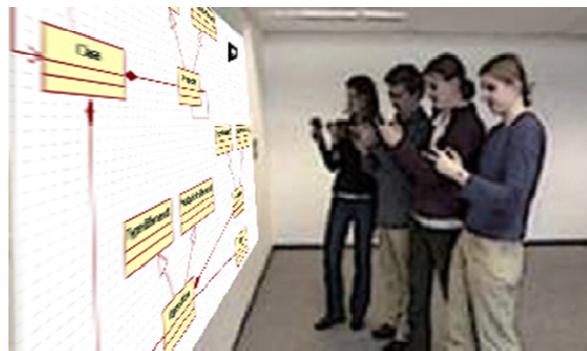


Figura 5.2. Projeção da arquitetura de software na parede (simulação)

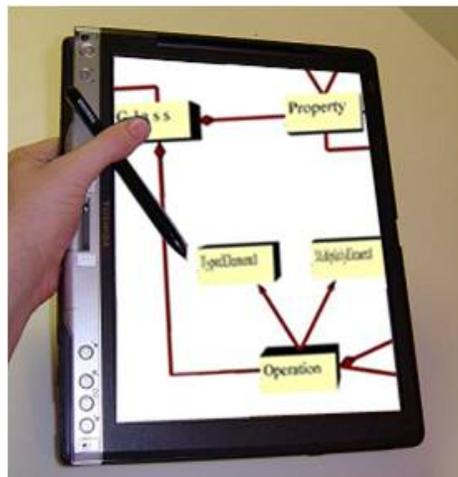


Figura 5.3. Dispositivo móvel dos alunos (simulação)

5.5.1. Módulo Arquitetural

O ambiente de apoio à modelagem arquitetural adotado será o ambiente de reutilização

Odyssey (ODYSSEY, 2009). O Odyssey foi selecionado por oferecer recursos que dão apoio à abordagem proposta, entre eles um editor de AS e a ferramenta FrameDoc (MURTA *et al.*, 2001) de documentação. Este é um projeto que está em desenvolvimento pelo grupo de reutilização de software da COPPE/UFRJ (REUSE, 2009) desde 1998, e está em evolução até os dias atuais. Os recursos providos pelo Odyssey e do Framedoc são detalhados a seguir.

5.5.1.1. Odyssey

Segundo HEER e AGRAWALA (2006), não existe uma única ferramenta ou *framework* que seja apropriada para todos os problemas de um dado domínio. E completam afirmando que os desenvolvedores devem migrar entre ferramentas ou construir seus próprios sistemas. Existe no mercado uma grande quantidade de editores de arquitetura e, na abordagem proposta, havia a necessidade de desenvolver um editor de AS, ou ampliar algum existente. Como o editor não é o foco principal deste trabalho de pesquisa e sim, o controle sobre a modelagem resultante com o uso deste tipo de ferramenta, foi decidido fazer adaptações em um editor conhecido. Por essa razão, é que a solução proposta faz uso de editores de diagramas OO desenvolvidos no contexto do Projeto Odyssey.

O Odyssey é um projeto que visa explorar técnicas e ferramentas que apoiem a reutilização de software, permitindo a evolução de um ferramental de apoio. O Odyssey (WERNER *et al.*, 2003) é um ambiente de reutilização baseado em modelos de domínio, provendo tecnologias de apoio à engenharia de domínio (ED), linha de produtos (LP) e Desenvolvimento Baseado em Componentes (DBC). Ele cobre tanto o desenvolvimento para reutilização, através de processos de ED (BRAGA, 2000; BLOIS, 2006), quanto o desenvolvimento com reutilização, através da Engenharia de Aplicação (EA) (MILER, 2000). Os modelos criados na ED são instanciados através de um processo de EA, que tem por objetivo a construção de aplicações em um determinado domínio. O Odyssey serve como um arcabouço onde modelos conceituais e AS são especificados para domínios de aplicações específicos.

Uma série de trabalhos, desenvolvidos na COPPE/UFRJ, são responsáveis pela evolução do Odyssey, dentre eles aqueles ligados aos modelos arquiteturais em teses de doutorado (BLOIS, 2006), (VASCONCELOS, 2007) e em dissertações de mestrado (XAVIER, 2001), (KÜMMEL, 2007), (MOURA, 2009), além de outros trabalhos que

desenvolveram ou estenderam funcionalidades e ferramentas para o ambiente.

Durante a aula, para que o aluno utilize a VisAr3D e visualize uma AS em 3D, esta deve ter sido criada e documentada no Odyssey. Seu editor modela a arquitetura de um sistema de software com as suas características e complexidades. Antes mesmo de executar a VisAr3D, os alunos podem compartilhar o mesmo projeto arquitetural no ambiente Odyssey, dividir suas tarefas e somar esforços para modelar não só um projeto exemplo, mas um sistema complexo, mais próximo da realidade, atendendo à demanda da indústria.

5.5.1.2. FrameDoc

Durante o desenvolvimento de um projeto arquitetural, é produzida uma grande quantidade de documentação que facilita a comunicação entre os *stakeholders*, registra as decisões iniciais acerca do projeto de alto-nível, e permite o reuso do projeto. Estas informações são de grande importância para o arquiteto de software e, principalmente, para o aluno de arquitetura de software, no entendimento de todo o processo de criação e desenvolvimento.

O Odyssey utiliza uma infra-estrutura para a documentação de artefatos de software, denominada FrameDoc (MURTA *et al.*, 2001). O FrameDoc é uma ferramenta de documentação, que controla a criação e disponibilização de documentos que contenham informações incorporadas aos componentes da arquitetura dentro do Odyssey. Este *framework* permite a criação de *templates* de documentação, que são posteriormente preenchidos para cada tipo de componente (classe, caso de uso, diagrama de estados etc.). Os tipos de campos fornecidos para a construção do *template* são: texto, *memo*, *checkbox*, *radiobutton*, som, imagem, vídeo e HTML.

Algumas das características do FrameDoc são: a exportação da documentação utilizando o padrão HTML, utilização de hiperligações no campo HTML para outros documentos ou páginas na Internet, criação de padrões de documentação (representados pelos *templates*), geração de documentação modular, utilização de multimídia na documentação, geração da documentação implícita ao componente (diagramas UML) e visualização da documentação a ser gerada.

Com o objetivo de incentivar a utilização destas informações no processo de ensino proposto por esta abordagem, a VisAr3D deve disponibilizar a documentação de forma contextualizada, ou seja, exibida sobreposta à arquitetura 3D. Este recurso facilitará o

acesso à documentação.

5.5.2. Módulo de Realidade Aumentada

A VisAr3D será uma ferramenta instalada no dispositivo móvel do aluno como PDA ou *Tablet PC*, ligada à rede de computadores. Ela utilizará a RA para identificar a AS apresentada pelo professor, exibido na parede por um projetor ou em qualquer material impresso, como um relatório ou um cartaz. Esta AS, criada, anteriormente, na ferramenta de edição do Odyssey, conterá um padrão gráfico que identifica aquele diagrama (figura 5.4). Ao utilizar as bibliotecas de RA do ARToolkit (KATO, 2009), este padrão gráfico pode ser capturado pela *Webcam* embutida no dispositivo do aluno. A VisAr3D utilizará este recurso para identificar a arquitetura que está sendo estudada e o ponto exato de visualização nesta arquitetura. Para acessar uma AS, sem utilizar este recurso de RA, o aluno terá que conhecer o nome do arquivo e o diretório onde ele foi salvo. E, finalmente, posicionar a arquitetura na tela no mesmo ponto daquela exibida pelo professor.

O módulo de RA torna o objeto de estudo mais acessível ao usuário e integrado de forma natural às atividades de sala de aula, trazendo benefícios à aprendizagem dos alunos. A intenção é quebrar a resistência dos professores à adoção das novas ferramentas na sua prática docente. Além de ser facilmente utilizável, garante a exibição da modelagem na versão mais recente. Outra utilidade deste módulo é a exibição de pequenas mensagens, comentários, ou ícones identificadores a respeito daquela AS, no momento da sua identificação. Ao capturar o padrão gráfico, o aluno vê sobreposto a ele, por exemplo, um indicador mostrando o estágio do desenvolvimento daquela arquitetura, um texto mencionando as próximas tarefas a serem realizadas, ou ícones indicando pontos a serem observados na modelagem. Algumas destas informações são pessoais, ou seja, só podem ser visualizadas pelos destinatários daquela mensagem (são recados privados, semelhantes ao email). Alunos diferentes visualizam informações diferentes quando utilizam este módulo de RA.

5.5.3. Módulo de Realidade Virtual

Após o reconhecimento da AS pelo dispositivo móvel do aluno, a VisAr3D representará essa arquitetura em 3D. Haverá a geração automática de modelos tridimensionais, a partir da identificação dos modelos 2D exportados do Odyssey. A VisAr3D passa a ser um

sistema responsável pela criação e manutenção de um ambiente virtual, no qual os objetos são acessados via rede para exploração e interação. Estes objetos correspondem à mesma arquitetura exibida pelo professor, agora modelados no formato tridimensional. Este ambiente virtual é atualizado em tempo real.

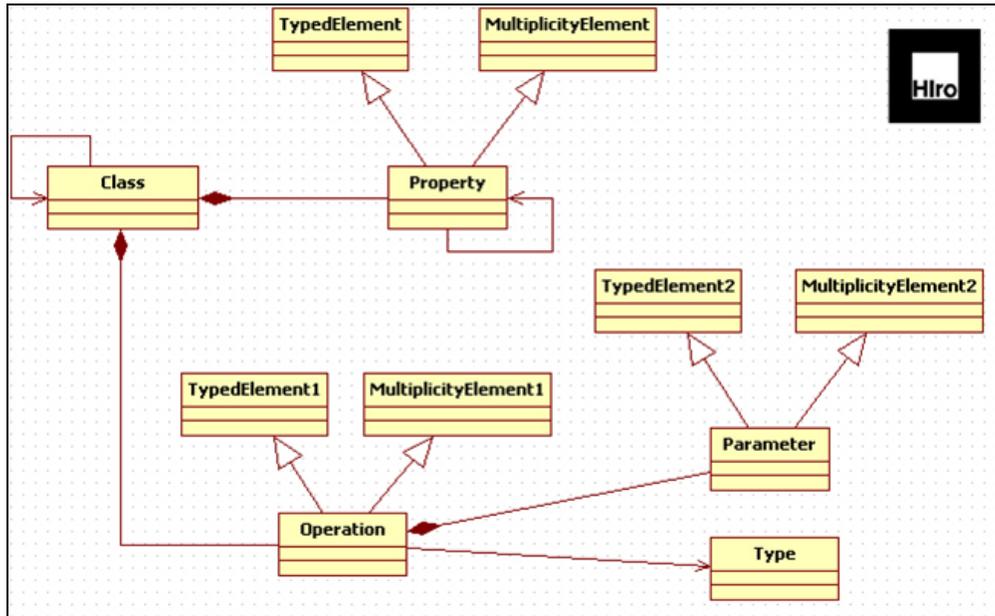


Figura 5.4. Exemplo de arquitetura em 2D com padrão gráfico

A comunicação da VisAr3D com o Odyssey, que permitirá a geração automática dos modelos 3D é feita através da utilização de arquivos XML.

O ambiente virtual da VisAr3D deve prover as seguintes funcionalidades que serão utilizadas no apoio ao ensino de AS:

- Informações contextualizadas
- Exploração no ambiente 3D
- Busca através de filtros
- Troca de informações entre alunos

5.5.3.1. Informações contextualizadas

Esta funcionalidade está associada à documentação disponibilizada no Odyssey pelo FrameDoc. Na VisAr3D, esta documentação deve ser visualizada sobreposta à arquitetura, são informações contextualizadas. Ela deve ser de fácil percepção, sendo

identificada por ícones, objetos em cores ou bandeiras (*flags*), ressaltando a sua correspondência ao seu conteúdo (figura 5.5).

Para evitar a poluição de informação associada aos diagramas, estas informações contextualizadas devem aparecer somente quando o usuário se aproximar do artefato documentado. No entanto, elas podem estar sempre disponíveis, se esta função for configurada pelo aluno.

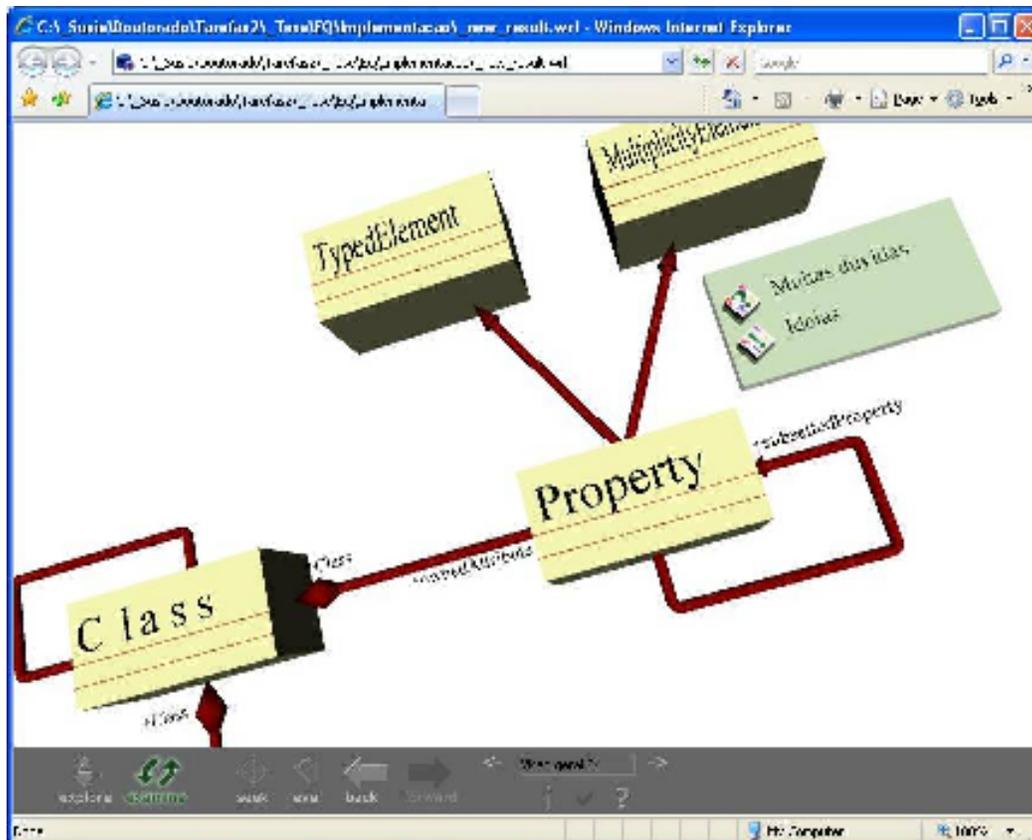


Figura 5.5. Visualização 3D com informações contextualizadas

Este recurso é interessante por exibir informações associadas a objetos gráficos, que apóiam a aula, ajudam na sua compreensão ou constituem-se num material rico para discussão. Tornam-se, também, imprescindíveis para o estudo de um grande volume de dados dispersos na rede. Elas podem ser: descrições sobre detalhes da arquitetura, anotações sobre pontos de impasse numa reunião, indicadores de qualquer tipo, mensagens entre usuários ao discutir sobre aquela modelagem, vídeos de reunião, áudio de relatos ou qualquer documentação que contribua para o aprendizado do aluno sobre aquela arquitetura. Em muitos casos, informações como estas, certamente existem, mas

podem estar soltas ou perdidas, sem estarem associadas àquele modelo arquitetural. Esta é uma maneira de resgatar informações pertinentes, que se perderiam com o tempo, e disponibilizá-las para utilização.

5.5.3.2. Exploração no ambiente 3D

A VisAr3D, como ambiente virtual, deve permitir a exploração do ambiente pelo aluno com o auxílio do mouse (ou outro dispositivo como as *data-glove* ou os gestos das mãos, por exemplo), movendo-se no espaço para a direita, esquerda, para cima, para baixo, se afastar ou se aproximar no plano e fazer movimentos de rotação.

O estudante deve ser capaz de visualizar a arquitetura em 3D em ângulos e distâncias diferentes com o auxílio de ícones e cores. A VisAr3D deve, também, disponibilizar o zoom, que permite ao estudante se aproximar e perceber a presença de mais informações contextualizadas e se afastar fazendo com que estas informações desapareçam e, dependendo da distância, que, seja exibido um nível mais alto de abstração da arquitetura estudada. Este recurso permite a análise da arquitetura numa nova perspectiva e, assim, é possível descobrir similaridades entre partes daquele diagrama, entender relacionamentos mais complexos e perceber diferentes técnicas e estilos arquiteturais. Desta forma, a VisAr3D serve de apoio à participação do aluno em sistemas mais complexos, atendendo ao primeiro requisito da abordagem.

O espaço ilimitado para exploração dos dados neste ambiente de visualização 3D facilitará o entendimento de diagramas maiores, com diferentes níveis de abstrações, versões ou que tenham sido projetados por pessoas diferentes.

Um recurso de “ponto de vista” (*viewpoint*) poderá facilitar a movimentação do aluno em pontos de interesse pré-definidos. O professor, ou o aluno, poderá marcar (definir) uma posição na arquitetura como um novo ponto de vista e o usuário da ferramenta será capaz de mover-se entre estes pontos de interesse, distantes na arquitetura, rapidamente.

5.5.3.3. Busca através de filtros

A VisAr3D deve disponibilizar um agente que permitirá a busca de documentos por meio palavras-chave ou filtros. O resultado pode ser uma lista de *links* ou a visualização gráfica através de cores ou ícones sinalizadores. Através de sua pesquisa, por exemplo, o aluno poderá visualizar, graficamente, os componentes da arquitetura criados por certo

usuário. Este subsistema de busca permitirá acessos fáceis e rápidos por todo tipo de informação associada à AS estudada. O resultado de uma busca poderá, ainda, disponibilizar outras arquiteturas associadas, visões ou documentações.

5.5.3.4. Troca de informações entre alunos

A VisAr3D, como ferramenta pedagógica interativa, deve permitir a troca de mensagens e informações entre alunos e entre alunos e professores, tornando as aulas mais interessantes e favorecendo a construção do conhecimento dos alunos por meio da aprendizagem colaborativa. O mecanismo a ser disponibilizado é semelhante ao bate-papo das ferramentas de comunicação disponíveis no mercado e muito utilizadas pelos alunos no dia-a-dia.

Esta troca de informações é fundamental para a realização de atividades educativas. Através de situações espontâneas e algumas vezes até lúdicas, o aluno poderá explorar, criar e, principalmente, comunicar-se para obter novas informações e construir novos processos para o seu desenvolvimento. De posse dessas informações interligadas e acessíveis, o aluno tem a possibilidade de comunicar-se com os colegas ou com o professor, compartilhar descobertas, dúvidas ou simples mensagens. Todo o histórico desta comunicação será armazenada no repositório de documentação da VisAr3D. Estas informações poderão ser valiosas para ajudar no entendimento de pontos polêmicos e na elaboração da documentação da AS.

Este ambiente de troca de informações é propício para a realização de jogos ou desafios propostos pelo professor, como outra possibilidade de interação e colaboração entre os alunos, não excluindo a interação face-a-face. Como exemplo desta utilização, tem-se o estímulo de competição utilizando “Estudos de Caso”, tais como o desafio de refatoração de uma parte da arquitetura, dando ênfase em diferentes propriedades não-funcionais, e o estímulo à exploração de uma arquitetura complexa com a tarefa de achar um estilo arquitetural e justificar a sua utilização, ou observar as propriedades não-funcionais utilizadas.

5.6. Repositórios

A VisAr3D utiliza três repositórios, conforme figura 5.1, que estão disponíveis em rede e, através de arquivos XML, permite a interoperabilidade com Odyssey:

- Um repositório de AS, que identifica e disponibiliza os diagramas modelados nos editores do ambiente Odyssey.
- Um repositório de documentação sobre estas arquiteturas, que disponibiliza textos, vídeos, sons, imagens e hiperlinks para outros documentos, também disponibilizados pelo ambiente Odyssey através do FrameDoc;
- Um repositório de documentação gerado pela própria VisAr3D.

Os dois primeiros são repositórios gerados a partir do Odyssey e alimentam o VisAr3D com todas as informações sobre a arquitetura estudada. Esta interoperabilidade é feita através de arquivos XMI (XML Model Interchange) (OMG, 2005), detalhada no próximo capítulo. Através deste tipo de arquivo, a VisAr3D é capaz de gerar automaticamente os modelos 2D exportados pelo Odyssey, em modelos 3D.

O repositório da VisAr3D armazena dados resultantes da utilização do ambiente virtual pelos alunos e professor. Ele contém informações sobre as interações, explorações, configurações pessoais do ambiente e troca de mensagens entre os usuários feitas durante a utilização da ferramenta.

5.7. Recursos necessários

Os recursos mínimos para a utilização do ferramental de apoio à VisAr3D são: uma AS projetada na parede ou impressa no papel com um padrão gráfico, um dispositivo móvel com webcam, conectado à rede de computadores.

No entanto, a abordagem VisAr3D, ao utilizar as tecnologias de RV e RA, pode oferecer, ainda, oportunidades para a utilização de outros dispositivos interativos, diferenciados do padrão (mouse e teclado), como, também, os óculos e as luvas (*data-gloves*), permitindo novas experiências aos alunos e ao professor.

A estereocopia é uma técnica relacionada com a percepção espacial que pode ser utilizada no apoio ao ensino da disciplina de AS. A possibilidade de proporcionar a imersão aumenta a sensação de realismo e, por isso, pode trazer motivação ou um ganho didático aos estudantes, que pode ser averiguado através de uma avaliação da utilização deste recurso.

A possibilidade de utilização de estereoscopia na abordagem VisAr3D já é viável, atualmente, pois a infra-estrutura necessária encontra-se disponível através do

investimento no laboratório recém inaugurado Lab3D – Laboratório de Realidade Virtual¹, da COPPE. Financiado pelo MEC (Sesu) e pela da COPPE, o Lab3D tem como objetivo o trabalho de excelência em visualização 3D, animação, convergência digital e suas áreas correlatas. Possui parcerias estratégicas com grupos de pesquisa que atuam em diversas áreas do conhecimento e a integração com os cursos de graduação e pós-graduação. Pretende-se, através de um ambiente multidisciplinar, favorecer a criação de soluções inovadoras que atendam à sociedade, à pesquisa, à produção científica e à formação de alunos, conciliando os trabalhos de desenvolvimento de aplicativos e interfaces com as atividades de ensino. Seus equipamentos e recursos atendem as necessidades da proposta desta tese. A infra-estrutura oferecida é uma sala de visualização 3D, um laboratório de computadores e dispositivos interativos.

Para obter o efeito estereoscópico serão usados os recursos apresentados na figura 5.6.

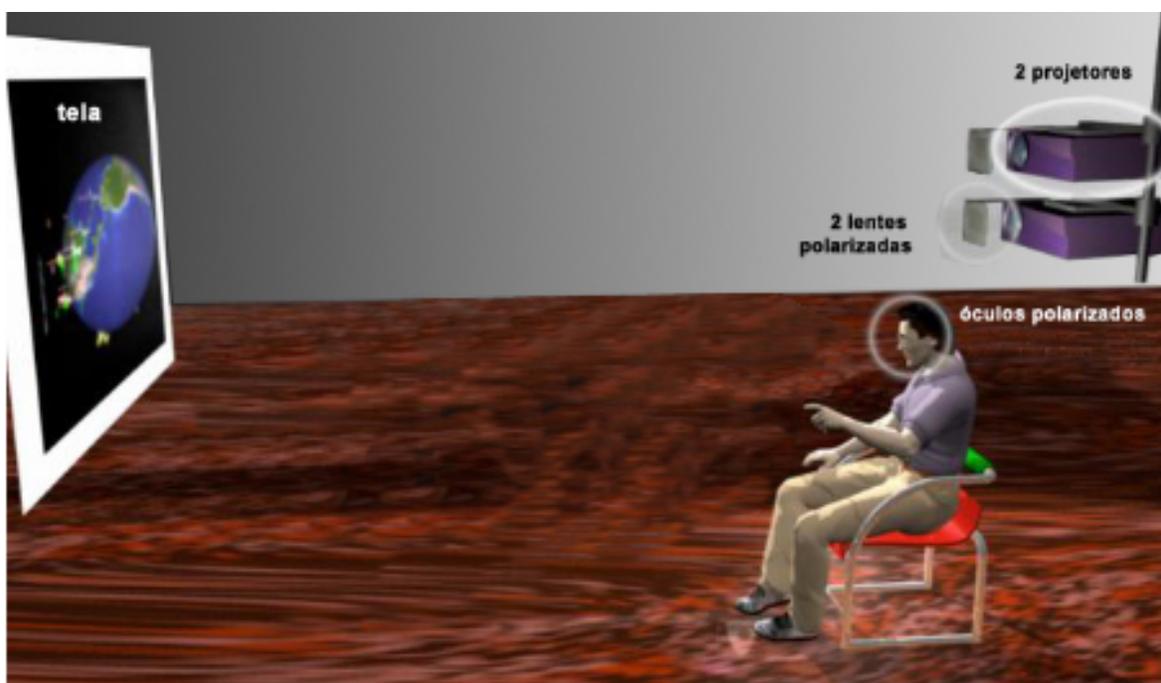


Figura 5.6. Projeção Polarizada

O uso de dispositivos interativos pode ser introduzido para flexibilizar o acesso desses usuários aos elementos da arquitetura. Um software verifica o que está sendo mostrado na tela e, em tempo real, envia comandos para a luva, dando ao usuário a sensação de

¹ [HTTP://lab3d.coppe.ufrj.br](http://lab3d.coppe.ufrj.br)

toque nos objetos visualizados.

Os dispositivos móveis sugeridos para serem utilizados com esta abordagem são a *tablet* PC e o PDA, devido à sua portabilidade. No entanto, a *tablet* PC é o mais adequado dos dois, devido à limitada capacidade da tela do PDA. Seu uso implicará na adoção de uma estratégia de exibição da aplicação direcionada a esta plataforma e a adaptação de conteúdo às suas restrições.

Outra oportunidade, na área de Educação, conquista do grupo e, em particular, deste projeto de tese, é o Projeto recém-aprovado, chamado TecEES: Tecnologias para Educação em Engenharia de Software. Trata-se de um projeto de colaboração CNPQ/NSF, entre a COPPE/UFRJ e a Universidade da Califórnia (UCI), em Irvine, incluindo ainda a Universidade Federal Fluminense (UFF). O objetivo deste projeto é investigar e desenvolver pesquisas em torno de abordagens, metodologias e estratégias de ensino de ES, com o intuito de facilitar e ampliar o conhecimento do estudante no desenvolvimento da sua capacidade de elaboração de projetos de software mais complexos e próximos aos utilizados na indústria. A idéia consiste em incrementar o processo de aprendizagem e a tomada de decisões, considerando a realização de simulações e o tratamento de questões de colaboração entre os estudantes e os professores, de modo prático e interessante para o aluno, por meio da aplicação de novas tecnologias, tais como Realidade Virtual, Realidade Aumentada e TV Digital e Interativa, e da exploração de estratégias, tais como Aprendizagem Colaborativa Apoiada por Computador, Design Criativo, Ambientes Virtuais de Aprendizagem, Aprendizagem Baseada em Problema e Educação à Distância, acarretando em uma melhoria na qualidade do ensino. A cooperação internacional formal permitirá um aporte financeiro importante para a compra de novos dispositivos para a realização desta pesquisa e, ainda, a disponibilização imediata de recursos sofisticados pela Universidade da Califórnia, que inclui *displays* grandes de alta resolução e monitores interativos que permitem a sua utilização por vários usuários, simultaneamente.

5.8. Exemplos de Aplicação da Abordagem

A integração com o ferramental de apoio da VisAr3D com o Ambiente Odyssey permite a utilização de algumas soluções relacionadas à AS como exemplo de abordagem de ensino. Ou seja, a VisAr3D aplica seus recursos e funcionalidades em exemplos de abordagens desenvolvidas no contexto do Odyssey, para atender ao seu principal objetivo

que é apoiar o ensino de AS. Os exemplos utilizados e descritos a seguir são dissertações de mestrado, uma concluída e outra em andamento, desenvolvidas pelo grupo de Reutilização da COPPE/UFRJ.

5.8.1 Abordagem ROOSC

A abordagem ROOSC (*Reengineering Object Oriented Software to Components*) (MOURA, 2009) é composta por duas etapas: a reestruturação dos modelos de pacotes e classes, que visa obter agrupamentos de classes (i.e. pacotes) candidatos a Componentes, de modo que atendam aos princípios de Desenvolvimento Baseado em Componentes (DBC); e a geração dos componentes propriamente, que visa obter um modelo de componentes com suas interfaces a partir dos agrupamentos de classes (i.e. pacotes) reestruturados. Trata-se de uma abordagem sistemática de apoio à reengenharia de software orientado a objetos (OO) para componentes baseada em métricas. Através desta abordagem, há a possibilidade de reaproveitar o conhecimento e esforço presentes em um software legado OO.

A fim de apoiar a reutilização, a ROOSC reestrutura os agrupamentos de classes (i.e. pacotes) com base em métricas adaptadas do contexto da OO para o contexto de DBC. Uma vez que pacotes são utilizados para agrupar elementos em OO segundo qualquer critério, a estratégia de reestruturação da abordagem ROOSC facilita a manutenção da aplicação ao reorganizar estes pacotes, tornando-os mais coesos e menos acoplados. Assim, ao final da reestruturação, os elementos que estão agrupados em cada pacote tendem a ser modificados em conjunto, facilitando a manutenção em pacotes localizados. A partir destes pacotes, que foram reestruturados com base nos princípios de DBC, são especificados os componentes e suas interfaces em uma arquitetura lógica.

Para apoiar a abordagem ROOSC, foram desenvolvidas duas ferramentas integradas ao ambiente Odyssey: a ORC (*Object Restructuring to Components*) e a GenComp (*Generating Components*).

Ao utilizar a VisAr3D como estratégia de apoio ao aprendizado deste conjunto de métricas e das reestruturações sugeridas pelo ROOSC para reorganização dos agrupamentos de classes (i.e. pacotes) candidatos a componentes, os recursos do ambiente virtual e, ainda, as seguintes funcionalidades poderão ser oferecidas:

- a visualização tridimensional dos diagramas antes e depois das reestruturações;
- a visualização das métricas em cores nos modelos 3D;

- um simulador para testar os efeitos das sugestões da ROOSC;
- uma pré-análise das reestruturações;
- a visualização de um número grande de elementos do modelo resultante.

Estas funcionalidades são descritas, de forma sucinta, a seguir:

a) A visualização tridimensional dos diagramas antes e depois das reestruturações

A VisAr3D identificará a arquitetura de software estudada e modelará automaticamente a mesma, em três dimensões. O aluno será capaz de visualizar em 3D os diagramas antes e depois das reestruturações sugeridas pelo ROOSC.

b) A visualização individual das métricas em cores nos modelos 3D

O modelo importado do ambiente do Odyssey apresentará métricas e sugestões de algumas reestruturações deste modelo. Estas métricas serão visualizadas individualmente em cores nas classes (ou pacotes) correspondentes no modelo tridimensional, em relação ao seu valor de referência. Seus valores, suas descrições e interpretações podem estar também disponíveis.

c) Um simulador para testar os efeitos das sugestões da ROOSC

Esta funcionalidade proverá um exercício de simulação para testar os efeitos das sugestões da ROOSC, para que o aluno engenheiro de software compreenda cada etapa desta funcionalidade. Para cada sugestão de reestruturação ou formação de componentes, ter-se-á a opção da sua simulação imediata. O aluno poderá interagir com modelos exemplo (semelhante ao modelo importado) e exercitar aquele tipo de reestruturação, manipulando e movendo os objetos modelados. Ele poderá criar outras opções de diagramas semelhantes e, ainda, visualizar, imediatamente, a consequência da sua interação. Por exemplo, a alteração no valor daquela métrica.

Como exemplo deste simulador, o aluno poderá reunir um grupo de classes ligadas pelo relacionamento de herança (métrica – NOCC reestruturação – mover hierarquia) num novo pacote e visualizar, imediatamente, a vantagem numérica daquela sugestão de reestruturação. Isto possibilitará um entendimento maior do conjunto de métricas envolvidas e das sugestões propostas pela ferramenta.

d) Uma pré-análise das reestruturações

Outra funcionalidade da VisAr3D seria apoiar o acompanhamento e a pré-análise do conjunto de reestruturações realizadas passo-a-passo e seus efeitos interligados. Ela possibilitará a sua visualização numa linha do tempo, iluminando as métricas intermediárias. Esta funcionalidade visa apoiar o engenheiro de software no momento de decidir sobre desfazer ou não uma seqüência de reestruturações aplicadas. Desta forma,

aproveitando a facilidade de visualizar um conjunto grande de dados, a VisAr3D possibilitaria o entendimento de cada sugestão de reestruturação à medida que cada uma é aplicada isoladamente ou em conjunto com outras reestruturações do mesmo tipo. Isto facilitaria a percepção do engenheiro de software sobre quando uma métrica melhora, ou entenderia que aquela outra, não foi eficiente, quando outra métrica se afasta de seu valor de referência. A partir daí, o engenheiro de software poderá desfazer a reestruturação até algum ponto desejado.

e) A visualização de um número grande de elementos do modelo resultante

Uma vez que os modelos recuperados através da engenharia reversa tendem a ter um grande número de elementos, o engenheiro de software pode ter dificuldade em compreender as reestruturações que estão sendo realizadas e o modelo resultante da etapa de reestruturação da abordagem ROOSC. Neste contexto, a VisAr3D poderá contribuir, também, ao facilitar a visualização dos elementos do modelo resultante desta etapa da abordagem.

Na segunda etapa da abordagem ROOSC, a geração de componentes propriamente, o modelo de componentes resultante pode apresentar um grande número de componentes e interfaces, tornando a visualização do resultado uma atividade difícil para o engenheiro de software. Para facilitar a navegação entre os componentes e interfaces do modelo de componentes resultante, o engenheiro de software poderá utilizar a VisAr3D, enfatizando uma área específica do modelo, assim como, navegar pelo modelo apoiado por toda facilidade provida pela visualização em 3D.

Como contribuição adicional, a VisAr3D poderá ainda atender a um trabalho futuro previsto na dissertação da abordagem do ROOSC que é o refinamento do modelo de componentes. Segundo MOURA (2009), o modelo de componentes ainda pode ser refinado através de métricas para o agrupamento de componentes, métricas que avaliem a complexidade do componente, reusabilidade, entre outras.

5.8.2. Abordagem PREVia

Um outro exemplo de aplicação da VisAr3D poderia ser no contexto da abordagem PREViA - Procedimento para a Representação da Evolução por meio da Visualização de Arquiteturas (SCHOTS *et al.*, 2009). Esta abordagem tem como objetivo criar um mecanismo de visualização da evolução de modelos arquiteturais de software, obtendo informações sobre esta evolução a partir da comparação de suas versões armazenadas em fontes de dados versionados, tais como repositórios de gerência de configuração.

Com seu uso, pretende-se que seja possível obter uma melhor compreensão e percepção da aderência da arquitetura emergente com relação à arquitetura conceitual e das diferenças entre versões das arquiteturas, o que pode vir a auxiliar na tomada de decisões de projeto e de implementação do software. Esta abordagem permite a “sobreposição” da arquitetura emergente sobre a arquitetura conceitual no decorrer do tempo para facilitar a compreensão. Uma das possíveis formas de apresentação propostas pela abordagem PREVia é a exibição da arquitetura conceitual em um nível alto de transparência (que possa ser configurado dentro de um conjunto de escalas de cores, sujeito a restrições de visibilidade), como se fosse uma marca d’água (*watermark*) na tela. Na medida em que houver um aumento no nível de similaridade de um elemento descrito na arquitetura conceitual e encontrado (em implementação) na arquitetura emergente, esta irá “preenchendo” a arquitetura conceitual, realçando possíveis divergências encontradas. No que diz respeito às diferenças entre versões de arquiteturas, serão realçados os elementos em que for detectada alguma diferença. Este trabalho constitui uma dissertação de mestrado ainda em desenvolvimento. Ao utilizar a abordagem VisAr3D neste contexto, propõe-se explorar a capacidade de percepção do aluno a partir das relações espaciais apresentadas, permitindo a interpretação e a compreensão da comparação entre as arquiteturas num ambiente 3D. A VisAr3D tem a intenção de facilitar a análise e contribuir para a absorção de novos conhecimentos, ao utilizar o potencial deste ambiente tridimensional, suas funcionalidades e vantagens e, principalmente, através de uma visualização diferenciada.

5.9. Considerações Finais

Pelas suas características ligadas especificamente à Realidade Virtual (RV) e a Realidade Aumentada (RA), a abordagem VisAr3D possui em si, imensas potencialidades que, na área da Educação, podendo transformá-la num poderoso instrumento a serviço da Academia, que procura a mudança e a evolução no ensino de Engenharia de Software em geral, e de Arquitetura de Software, em especial. Estas tecnologias reúnem em si várias especificidades e atributos que podem ajudar nas múltiplas situações e contextos de pesquisa e aprendizagem.

Com a RV e a RA, a assimilação do conhecimento pode se tornar muito mais simples. Imersos no mundo virtual, os alunos participam de uma viagem exploratória que suscita a curiosidade, o interesse, o desafio e, conseqüentemente, a motivação, fundamental em

qualquer contexto educativo. Por esta razão, a sua adaptação é rápida e há, portanto, uma maior probabilidade de um maior engajamento, por parte dos alunos, na aula.

Ao interagir com este mundo envolvente, o estudante pode obter dele, em tempo real, a resposta adequada às suas ações. Com o simulador, a experiência prática pode ser vivida inúmeras vezes, facilitando a compreensão de erros cometidos, a aprendizagem das tarefas e de procedimentos complexos. Ele cria a oportunidade de transformação do erro em aprendizagem.

Deve-se ressaltar que durante todo o processo de utilização da VisAr3D, o papel do professor continua a ser fundamental. Ele é visto como um orientador, que trabalha conjuntamente com os seus alunos e os auxilia, aprendendo com eles, simultaneamente.

Este capítulo apresentou a abordagem proposta nesta pesquisa de tese, baseada na caracterização do problema e numa lista de requisitos. Uma solução foi apresentada para atendê-los, utilizando o potencial que as tecnologias de RV e RA têm para oferecer como suporte à educação.

Em resumo, apostando na crescente demanda pela análise visual do software, a VisAr3D se propõe a apoiar o desenvolvimento e a participação dos alunos em projetos complexos, reduzir a distância entre a teoria e a prática e apoiar a dinâmica em sala de aula, utilizando recursos que incluem a visualização 3D como diferencial facilitador e motivador para os alunos ao aprenderem AS.

Uma característica importante da ferramenta é a geração automática de modelos 3D, a partir de arquivos XMI exportados do Ambiente Odyssey. Esta funcionalidade cria oportunidades de ampliação da VisAr3D, à medida que as pesquisas realizadas no contexto do Odyssey são desenvolvidas. A VisAr3D poderá ser utilizada como uma ferramenta 3D de ensino e prática de AS destes projetos de pesquisa.

No capítulo seguinte, serão apresentados os trabalhos relacionados, as contribuições esperadas, o estágio atual da pesquisa e a metodologia de pesquisa adotada, assim como o cronograma das atividades necessárias para o desenvolvimento da proposta apresentada.

6. Conclusões

O objetivo deste capítulo é concluir esta monografia de proposta de tese, apresentando algumas semelhanças de alguns trabalhos relacionados, descrevendo as contribuições esperadas em função dos objetivos estabelecidos e apresentando o planejamento para a realização das atividades, assim como um cronograma, para o desenvolvimento desta pesquisa.

O capítulo é organizado da seguinte forma: a Seção 6.1 apresenta os Trabalhos Relacionados à abordagem proposta. A Seção 6.2 mostra as Contribuições Esperadas com esta pesquisa. Na Seção 6.3, é apresentado um panorama da Situação Atual da pesquisa. A Seção 6.4 descreve as atividades necessárias para a sua realização, dispostas em um cronograma proposto.

6.1. Trabalhos Relacionados

Os trabalhos desenvolvidos por BAKER *et al.* (2005), DE LUCENA *et al.* (2006), TRONT *et al.* (2006) e MOHRENSCHILDT e PETERS (1998) foram direcionados a apoiar o ensino de ES, não focando especificamente em AS, no entanto, tiveram grande influência no desenvolvimento desta proposta. As semelhanças destes trabalhos com a abordagem VisAr3D começam pelas suas propostas de serem atrativos ao aluno, e continuam nas seguintes características:

- *Problems and Programmers* (BAKER *et al.*, 2005): Com o foco na experiência prática, os alunos interagem face-a-face, trocando idéias e as registrando, construindo um raciocínio coletivo para o entendimento de uma situação real. A utilização de jogos e simulações também incentivam o ensino de conceitos e teorias.
- Projeto Unibral (DE LUCENA *et al.*, 2006) e *Draw-Bot* (MOHRENSCHILDT & PETERS, 1998): Além de expor aos estudantes os mesmos problemas que são encontrados em situações profissionais reais, a utilização de um robô, como uma plataforma de hardware existente, é um fator que contribui para um interesse dos alunos. Este recurso pode ser comparado à utilização da RA através de dispositivos móveis, presentes na abordagem VisAr3D.
- *WriteOn* (TRONT *et al.*, 2006): Seu foco, assim como na VisAr3D, é manter uma grande interatividade entre o professor e os alunos dentro da sala de aula.

Através da revisão sistemática conduzida neste trabalho, foram encontradas algumas

ferramentas que apóiam o ensino de AS, que são destacadas a seguir como trabalhos que contribuíram para esta abordagem:

- A ferramenta de modelagem, proposta por WANG e SCANNELL (2005), ajuda os estudantes a aprenderem e a conduzirem análises para avaliar a qualidade das alternativas de projetos arquiteturais. Ela se assemelha a esta abordagem, ao investir na facilidade de utilização da interface gráfica para incentivar os alunos e diminuir a curva de aprendizado.
- O ArchKriti, descrito em (VALLIESWARAN & MENEZES, 2007), se destaca, ao criar um framework que seja extensível, permitindo a adição de *plugins* que o torna capaz de integrá-lo com outras ferramentas de desenvolvimento. Com a constante evolução do Odyssey em desenvolver soluções relacionadas à AS, e a facilidade de sua integração com o ferramental de apoio da VisAr3D, torna-se viável utilizar este ambiente virtual, suficientemente, flexível para atender às necessidades de ensino destas técnicas e abordagens.
- O ArchE (MCGREGOR *et al.*, 2007) é uma outra ferramenta de software que funciona como um assistente ao arquiteto de software, interagindo através de sugestões aos alunos iniciantes à medida que ele cria seu projeto arquitetural. A VisAr3D, por sua vez, ao se aplicada para o ensino da abordagem ROOSC, provê sugestões de reestruturações para apoiar o ensino das métricas e o entendimento de cada reestruturação através da visualização tridimensional e o uso de simulação.

A escolha por um trabalho desta natureza decorre da importância de se discutir e investigar aspectos relacionados à construção do conhecimento na área de abrangência do ensino de AS. Levou-se em conta, também, aspectos relacionados ao interesse e à participação do aluno em sala de aula, que se espera ser aumentado com a introdução das tecnologias de RV e RA. Segundo as abordagens encontradas e citadas nesta seção, nenhuma envolvia todas as características da VisAr3D, no entanto, contribuíram de alguma forma, para que as mesmas estivessem ali contempladas. Por esta razão, e por todo estudo e pesquisa realizados e apresentados nesta monografia, acredita-se que esta pesquisa de tese aborda um tema inédito.

6.2. Contribuições esperadas

As contribuições esperadas desta pesquisa envolvem, principalmente:

- Apoiar a prática e o ensino de arquitetura de software ao ampliar trabalhos desenvolvidos no ambiente Odyssey, como por exemplo as abordagens ROOSC e PREVia.

Ao ampliar a abordagem ROOSC, inclui:

- Favorecer o entendimento das métricas utilizadas na abordagem ROOSC.
- Favorecer o entendimento das reestruturações sugeridas pela abordagem ROOSC e das combinações das aplicações destas reestruturações.
- Dar suporte à tomada de decisão do engenheiro de software (aluno) ao utilizar a abordagem ROOSC.

Ao ampliar a abordagem PREVia, inclui favorecer a percepção e a compreensão na comparação entre arquiteturas de software em 3D.

- Disponibilizar a visualização 3D de arquitetura de software, usualmente exibida em 2D, com fins educativos.
- Disponibilizar um ambiente virtual com recursos que apoiam o ensino de arquitetura de software.
- Inovar ao utilizar a Realidade Aumentada no apoio ao ensino de disciplinas, reconhecidas como virtuais.
- Suportar a comunicação entre alunos, favorecendo o trabalho em grupo e a tomada de decisão em seus projetos.
- Incentivar a utilização da documentação gerada durante o desenvolvimento do projeto arquitetural ou durante a utilização da VisAr3D, que representam as informações compartilhadas entre os alunos e professor.
- Contribuir para o aumento do interesse do aluno na prática da disciplina de arquitetura de software.
- Dar suporte à prática do ensino de arquitetura de software.
- Dar suporte à prática do ensino de arquitetura de software em projetos grandes.
- Apoiar ao ensino de arquitetura de software ao utilizar recursos de Realidade Virtual, Realidade Aumentada e Estereoscopia.
- Incentivar a dinâmica da sala de aula da disciplina de arquitetura de software, através da participação dos alunos.

6.3. Situação Atual

6.3.1. Revisão da Literatura

A revisão da literatura foi realizada até agora com base na bibliografia em anexo e serviu de base para a redação desta monografia. Pretende-se continuar a execução desta revisão bibliográfica, com o objetivo de estar atenta aos avanços da área, ou aprofundar algum tema específico envolvido na pesquisa.

Uma revisão *quasi*-sistemática foi realizada em abril de 2009 e serviu para alcançar resultados mais significativos ao identificar publicações pertinentes que contribuíram para a elaboração desta proposta. Devido à característica da revisão sistemática, ela permitirá a reprodução de seu protocolo em um outro momento, com a intenção de obter resultados mais atuais.

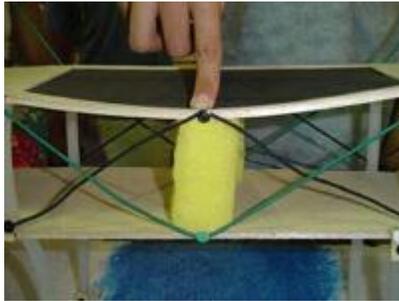
6.3.2. Estudo de linguagens e ambientes

A utilização de novas tecnologias como RV e RA demandam um investimento significativo no estudo de linguagens, ferramentas e ambientes de desenvolvimento de software. Esta Seção relata o esforço realizado neste sentido, para elaboração desta proposta de tese. O desenvolvimento dos dois primeiros protótipos teve como objetivo servir como base para a exploração das tecnologias atualmente disponíveis e como fonte de motivação no desenvolvimento deste trabalho.

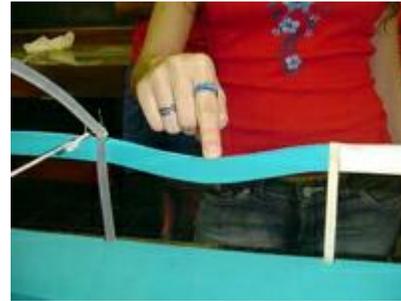
6.3.2.1. Protótipo de RA

Um protótipo de RA foi feito, utilizando a biblioteca ARToolkit. Este protótipo foi desenvolvido com a finalidade de facilitar a aprendizagem do comportamento estrutural de modelos qualitativos no ensino básico de arquitetura e engenharia (RODRIGUES *et al.*, 2008). O primeiro período do curso de Arquitetura e Urbanismo da UFRJ oferece uma disciplina que tem como objetivo mostrar ao aluno qual a função da estrutura na Arquitetura, introduzindo os sistemas estruturais existentes e seus comportamentos quando submetidos a determinados carregamentos, de forma qualitativa. Esta disciplina utiliza, obrigatoriamente, modelos reduzidos de sistemas estruturais, confeccionados com materiais flexíveis como o silicone, a borracha e o elástico, sem a preocupação com fatores de escala, nem de estética, para através da observação, propiciar uma melhor

compreensão do comportamento das estruturas. Simulando, portanto, os efeitos de tração, compressão, flexão, flambagem e torção. Utilizando-se o sentimento e a intuição. O aluno deve conceber sistemas estruturais, construir modelos e analisar qualitativamente o comportamento estrutural dos mesmos (figuras 6.1a e 6.1b).



(a)



(b)

Figura 6.1. Alunos interagindo com modelos de sistemas estruturais

Ao interagirem com os modelos, os alunos simulam a ação das cargas em determinadas estruturas e conseguem concluir qual é o tipo de configuração deformada associada ao esforço solicitante, pertinente a cada elemento estrutural observado. A proposta do protótipo foi desenvolver uma aplicação de RA, utilizando a biblioteca ARToolkit, acrescentando elementos virtuais, associados aos modelos dos alunos, que representem informações úteis sensíveis àquela interação, que os ajudem a compreender melhor: onde ficam as reações de apoio e como são as configurações deformadas. As figuras 6.2 e 6.3 mostram o protótipo em execução. O aluno, utilizando o marcador preso ao dedo, interage com o modelo aplicando uma força na viga e a ferramenta acrescenta informações virtuais, como as reações de apoio em relação à carga aplicada e a distância ao ponto de aplicação da carga e mostra, também, esquematicamente, as regiões tensionadas, ou seja, onde ocorre a tração e a compressão, representadas pelas letras “T” e “C”, respectivamente.

6.3.2.2. Protótipo de RV

Um protótipo de RV foi desenvolvido utilizando o mesmo tipo de aplicação descrito na Seção anterior, contudo voltado agora para a RV. A figura 6.4 mostra a aplicação de RV que simula a deformada de uma viga e as reações de apoio de acordo uma força “P”

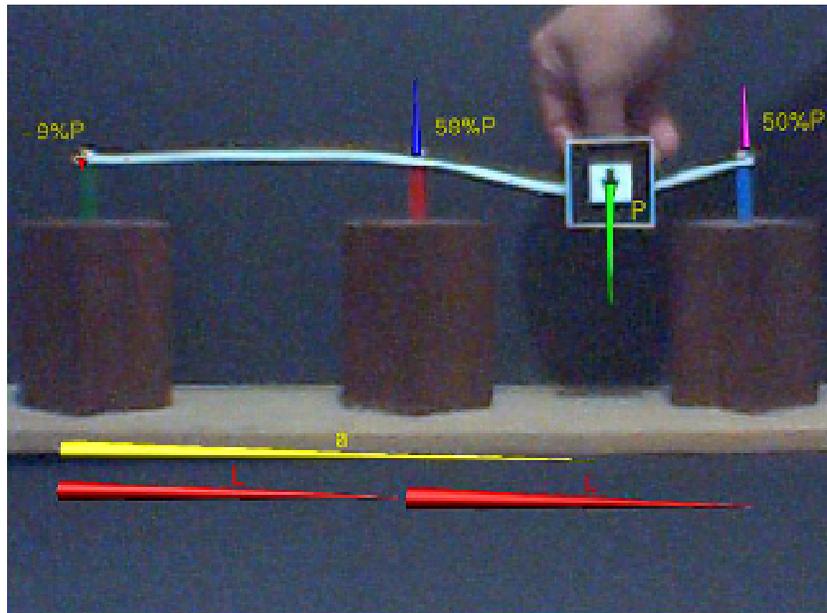


Figura 6.2. Protótipo com RA mostrando reações de apoio em um modelo de viga

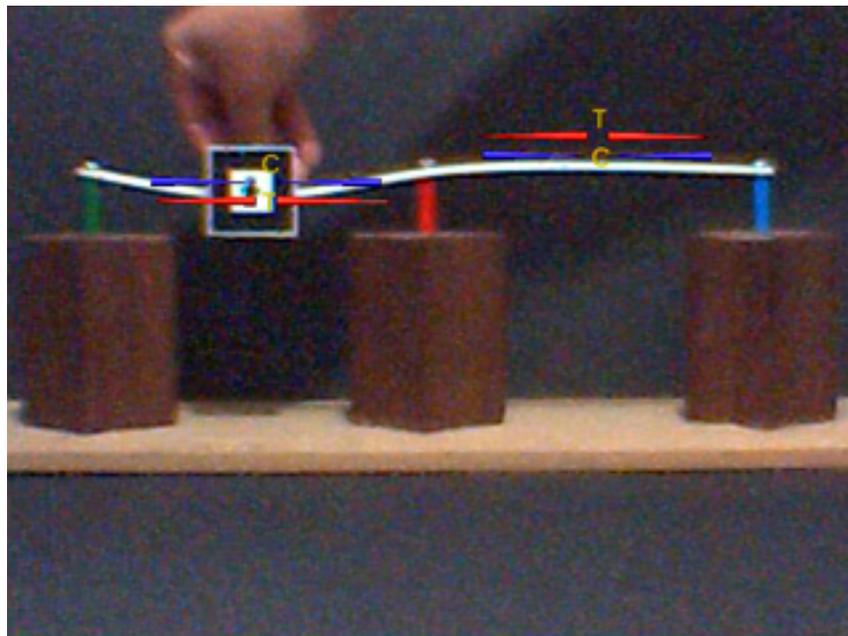


Figura 6.3. Protótipo com RA mostrando as regiões tensionadas em um modelo de viga

aplicada. O aluno, utilizando o clique do mouse, interage com o modelo aplicando uma força constante na viga e a ferramenta responde apresentando as reações de apoio. O ambiente virtual disponibiliza a exploração do ambiente 3D pelo usuário, acarretando novos ponto de vista da cena.

6.3.2.3. Evolução do Odyssey

A VisAr3D é uma ferramenta educativa integrada ao ambiente Odyssey. Esta integração se dá através da importação de diagramas modelados no editor do Odyssey e de documentação produzida com o FrameDoc, e da incorporação de funcionalidades da abordagem ROOSC e de funcionalidades da abordagem PREViA. Portanto, algumas adaptações e evoluções são necessárias para a integração destas ferramentas.

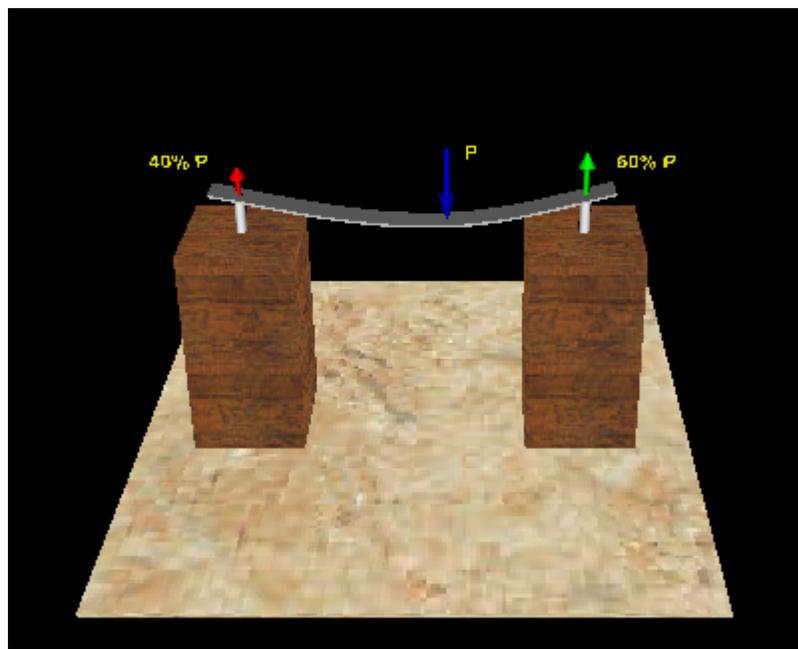


Figura 6.4. Protótipo de Aplicação de Realidade Virtual

6.3.2.4. Odyssey-XMI

A Visar3D modela, automaticamente, em 3D, a AS que foi construída no ambiente do Odyssey e, previamente, salva num arquivo XMI (XML Model Interchange) (OMG, 2005). Por isso, através de um estudo preliminar do Odyssey, observou-se que este precisaria

ser estendido para atender aos requisitos da VisAr3D, no que diz respeito à exportação de suas informações. Neste sentido, está sendo evoluído um *plugin* do Odyssey, denominado Odyssey-XMI (MURTA, 2006). Este *plugin* tem como finalidade possibilitar a interoperabilidade do Odyssey com outras ferramentas de modelagem. O Odyssey-XMI utiliza o XMI para a exportação e importação de modelos UML (BOOCH *et al.*, 2005).

O Odyssey-XMI, atualmente, utiliza a especificação UML 1.5 e, está sendo adaptado para também suportar o uso da UML 2.2. Para possibilitar que o Odyssey-XMI pudesse importar e exportar tanto modelos UML 1.5 como modelos UML 2.2, as classes do software foram refatoradas. Em função das particularidades da UML 2.2, os métodos receberam novos parâmetros, necessários para a viabilização da construção dos modelos durante a importação e exportação.

Para a integração com a VisAr3D, o Odyssey-XMI deverá importar e exportar modelos e diagramas UML, assim como informações adicionais presentes nas abordagens ROOSC e PREVia.

6.3.2.5. Estudo do X3D

A principal dificuldade encontrada no desenvolvimento do protótipo de RV, apresentado na Seção 6.3.2.2, foi a falta de documentação sobre a programação com VRML mais avançada. Por esta razão, para a implementação do ambiente virtual desta proposta, foi utilizada a linguagem sucessora do VRML, o X3D (X3D, 2009). O X3D é um padrão aberto para distribuir conteúdos de RV em 3D, em especial pela Internet. Esta linguagem é capaz de representar e comunicar cenas tridimensionais e objetos, desenvolvidos com a sintaxe XML. Como o formato XMI é usado para exportar modelos UML em XML, a interoperabilidade entre a VisAr3D e o ' está garantida.

Para que a visualização dos arquivos X3D seja possível no browser, fez-se necessário a instalação de um *plugin* específico, o Flux Player da Media Machines (FLUX DEVELOPER WIKI, 2009). Depois de escolhida a linguagem X3D, foi realizado um estudo mais profundo dessa linguagem através de tutoriais e testes.

6.3.3. Redação de Artigos e Relatórios Técnicos

Durante a fase de pesquisa, foram produzidos e publicados textos com a intenção de compartilhar a produção e o atual estágio de pesquisa com a comunidade acadêmica,

divulgando os resultados iniciais em conferências nacionais e internacionais. Estes são listados a seguir:

- RODRIGUES, C. S. C., RODRIGUES, P. F. N., WERNER, C. M. L., 2008, “An Application of Augmented Reality in Architectural Education for Understanding Structural Behavior through Models”, In: *Proceedings of the X Symposium on Virtual and Augmented Reality (SVR 2008)*, pp. 163-166, João Pessoa, May.
- RODRIGUES, C. S. C., WERNER, C. M. L., 2008, *Realidade Aumentada e Engenharia de Software*, In: Relatório técnico 1/2008, COPPE/UFRJ, Rio de Janeiro.
- RODRIGUES, C. S. C., WERNER, C. M. L., 2009, “VisAr3D: Uma abordagem baseada em Realidade Aumentada para o Ensino de Arquitetura de Software”, In: *Proceedings of the XII Conferencia Iberoamericana de Ingeniería de Requisitos Ambientes de Software (IDEAS 2009)*, pp. 361-372, Medellín, Abr.
- RODRIGUES, C. S. C., WERNER, C. M. L., 2009, “Software Architecture Teaching: A Systematic Review”, In: *Proceedings of the 9th World Conference on Computers in Education (WCCE 2009)*, pp. 1- 10, Bento Gonçalves, Jul.
- RODRIGUES, C. S. C., WERNER, C. M. L., 2009, *Uma Revisão Sistemática sobre as Iniciativas Realizadas no Ensino de Arquitetura de Software*, In: Relatório Técnico ES-728/09, COPPE/UFRJ, Rio de Janeiro.
- WERNER, C. M. L., RODRIGUES, C. S. C., SANTOS, R. P., COSTA, H. L. C., SANTO, R. E, CASTRO, W. S., 2009, “Projeto Tec3ES: Tecnologias e Estratégias para Educação em Engenharia de Software”, In: *Proceedings of the XVII Congresso Iberoamericano de Educação Superior em Computação (CIESC'2009)*, Pelotas, Set.

6.3.4. Avaliação da Abordagem VisAr3D

No início do processo de avaliação da abordagem VisAr3D pretende-se conduzir um estudo de viabilidade. Através de um estudo com esse objetivo, procura-se identificar se a abordagem realmente possui capacidade em produzir resultados práticos e se é viável continuar a despendar recursos para desenvolvê-la (SHULL et al., 2001).

O estudo visará identificar a viabilidade do mecanismo no apoio ao aprendizado do aluno. Com este estudo, pretende-se observar como os alunos numa aula de AS aprendem, com o suporte da VisAr3D. Mais especificamente, o estudo terá como foco a utilização da ferramenta como suporte ao ensino aplicados nas abordagens ROOSC e PREVia, enquanto este explora e interage com o ambiente virtual, ou troca mensagens entre outros alunos.

Feitos alguns refinamentos em função das lições aprendidas no primeiro estudo, em seguida, pretende-se realizar a comparação entre dois estudos observacionais no suporte ao ensino de AS. O termo “observacional” é usado para definir estudos onde o participante realiza alguma tarefa enquanto é observado por um experimentador (SHULL et al., 2001). Este tipo de estudo tem como finalidade coletar dados sobre como determinada tarefa é realizada. Através destas observações, pode-se obter uma compreensão de como uma ferramenta ou um processo novo é utilizado. O primeiro estudo será realizado utilizando a abordagem ROOSC original e o segundo, utilizando a abordagem ROOSC com o auxílio da VisAr3D. Pretende-se fazer uma comparação entre estes estudos para determinar se, na prática, esta última resulta num menor esforço para atender ao seu objetivo, que é apoiar os alunos no entendimento das métricas e das reestruturações dos modelos de pacotes e classes e, ainda, ser atrativo.

Inicialmente, nestas comparações, serão considerados componentes de atratividade, a capacidade de atrair alunos, a pouca desistência entre os alunos na hora de cumprir uma tarefa e o aumento no número de contribuições de documentação feitas pelos participantes.

Está prevista, também, a realização de um estudo similar, utilizando a abordagem PREVia, além de outros que possam evidenciar as contribuições da VisAr3D para o ensino de AS.

6.4. Fases para o Desenvolvimento e Cronograma

Nove fases compõem o desenvolvimento da tese proposta. Algumas dessas fases já foram encerradas, mas poderão ser revisitadas caso haja necessidade, outras encontram-se em andamento, ou estão previstas para começarem, conforme pode ser visto no cronograma da Tabela 6.1. As fases previstas, detalhadas a seguir, são:

1. Revisão da Literatura
2. Estudo de linguagens e ambientes
3. Redação e Apresentação da Monografia de Qualificação
4. Implementação da Ferramenta VisAr3D
5. Realização de Estudos de Caso
6. Redação de Artigos
7. Realização de Estudos Observacionais
8. Redação e Defesa de tese

Tabela 6.1. Cronograma

Atividades	Período-Mês/Ano												
	Jan	Abr	Jul	Out	Jan	Abr	Jul	Out	Jan	Abr	Jul	Out	Jan
	Mar /08	Jun /08	Set /08	Dez /08	Mar /09	Jun /09	Set /09	Dez /09	Mar /10	Jun /10	Set /10	Dez /10	Mar /11
1. Revisão da Literatura													
2. Estudo de linguagens e ambientes													
3. Redação e Apresentação da Monografia de Qualificação													
4. Implementação da Ferramenta VisAr3D													
5. Realização de Estudos de Caso													
6. Redação de Artigos													
7. Realização de Estudos Observacionais													
8. Redação e Defesa de tese													

7. Referências

- ACME: The acme architectural description language home-page, 1998. Disponível em: <<http://www.cs.cmu.edu/~acme/>>, Acesso em: 16 Set 2009.
- ALEXANDER, C., ISHIKAWA, S., SILVERSTEIN, M., LACOBSON, M., FIKSDAHLKING, I., ANGEL, S., 1977, **A Pattern Language**, Nova York, Oxford University Press.
- AL-TAHAT, K. S., SEMBOK, T. M. T., IDRIS, S. B., 2001, "Using design patterns in the development of a planner-based courseware system", In: *Proceedings of the IEEE Region 10 International Conference on Electrical and Electronic Technology*, pp. 873–876, Bangi, Malaysia, Aug.
- AURÉLIO, 1999, **Dicionário Aurélio Eletrônico – século XXI versão 3.0**, Lexicon Informática Ltda.
- AZUMA, R. T., BAILLOT, Y., BEHRINGER, R., FEINER, S., JULIER, S., MACINTYRE, B., 2001, "Recent Advances in Augmented Reality", **IEEE Computer Graphics and Applications**, v. 21, n. 6 (Nov.), pp. 34-47.
- BAKER, A., NAVARRO, E. O., VAN DER HOEK, A., 2005, "An experimental card game for teaching software engineering processes", **Journal of Systems and Software**, v. 75, n. 1-2 (Feb.), Software Engineering Education and Training, pp. 3-16 .
- BARCELOS, R. F., 2006, *Uma Abordagem para Inspeção de Documentos Arquiteturais Baseada em Checklist*, Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BASS, L., CLEMENTES, P. KAZMAN, R., 2003, **Software Architecture in Practice - Second Edition**, Boston, Pearson Education Inc.
- BASTOS, N. C., 2005, *Arquitetura para Dispositivos Não-Convencionais de Interação Utilizando Realidade Aumentada: Um Estudo de Caso*, Trabalho de Conclusão de Curso de Graduação em Ciência da Computação, Universidade Federal de Pernambuco, PE, Brasil.
- BAUER, F.L., 1972, **Software Engineering - Information Processing**, Amsterdam: North Holland Publishing.
- BAUER, M., HILLIGES, A., MACWILLIAMS, A., SANDOR, C., WAGNER, G., KLINKER, G., NEWMAN, J., REITMAYR, G., PINTARIC, T., FAHMY, T., SCHALSTIEG, D., 2003, "Integrating Studierstube and DWARF", In: *Proceedings of the International Workshop on Software Technology for Augmented Reality Systems (STARS 2003)*, pp. 1–5, Tokyo, Japan, Oct.

- BELL, J., FOGLEL, H. S., 1995, "The Investigation and Application of Virtual Reality as an Educational Tool", In: *Proceedings of the American Society for Engineering Education Annual Conference*, pp. 1718-1728, Anaheim, USA, Jun.
- BIOLCHINI, J., MIAN, P. G., NATALI, A. C. C., TRAVASSOS, G. H., 2005, Systematic Review in Software Engineering, Technical Report ES 679/05, COPPE/UFRJ, Rio de Janeiro.
- BLOIS, A. P. T. B., 2006, *Uma Abordagem de Projeto Arquitetural Baseado em Componentes no Contexto de Engenharia de Domínio*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BOER, R. C., FARENHORST, R., VAN VLIET, H., 2009, "A Community of Learners Approach to Software Architecture Education", In: *Proceedings of the 22nd Conference on Software Engineering Education and Training (CSEET '09)*, pp. 190-197, Hyderabad, India, Feb.
- BOOCH, G., RUMBAUGH, J., JACOBSON, I., 2005, *UML - Guia do Usuário*, 2 ed., Editora Campus.
- BRAGA, M., 2001, "Realidade Virtual e Educação", *Revista de Biologia e Ciências da Terra*, v. 1, n. 1, pp. 5-20.
- BRAGA, R. M. M., 2000, *Busca e Recuperação de Componentes em Ambientes de Reutilização de Software*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BUCCI, P., LONG, T. J., WEIDE, B. W., 1998, "Teaching software architecture principles in CS1/CS2". In: *Proceedings of the Third International Workshop on Software Architecture*, pp. 9-12, Orlando, USA, Nov.
- BUTZ, A., HÖLLERER, T., FEINER, S., MACINTYRE, B., BESHES, C., 1999, "Enveloping users and computers in a collaborative 3d augmented reality", In: *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pp. 35-44, San Francisco, USA, Oct.
- CARDOSO, A., LAMOUNIER, E. A., 2008, "Aplicações na Educação e Treinamento", In: *Realidade Virtual e Aumentada - Uma Abordagem Tecnológica*, v. 1, Editora SBC - Sociedade Brasileira de Computação, Porto Alegre, pp. 343-357.
- CHENOWETH, S., ARDIS, M., DUGAS, C., 2007, "Adapting cooperative learning to teach software architecture in multiple role-teams", In: *Proceedings of the ASEE Annual Conference and Exposition*, pp. 1-12, Honolulu, USA, Jun.
- CONN, R., 2002, "Developing software engineers at the C-130J software factory", *IEEE Software*, v. 19, n. 5 (Oct.), pp 25-29.

- COPLIEN, J. O., SCHMIDT, D. C., 1995, *Pattern Languages of Program Design*, Addison-Wesley Publishing, Reading.
- CREIGHTON, O., SINGER, M., 2008, "Who leads our future leaders? on the rising relevance of social competence in software development", In: *Proceedings of the International Conference on Software Engineering*, pp. 23–26, Leipzig, Germany, May.
- DARWIN: An Architectural Description Language, Disponível em: <<http://www-dse.doc.ic.ac.uk/Software/Darwin/>>, Acesso em 22 Ago 2009.
- DE LUCENA, V. F., BRITO, A., GOHNER, P., 2006, "A Germany-Brazil Experience Report on Teaching Software Engineering for Electrical Engineering Undergraduate Students", In: *Proceedings of the 19th Conference on Software Engineering Education & Training (CSEET'06)*, pp. 69-76, Turtle Bay, USA, Apr.
- DENZLER, C., GRUNTZ, D., 2008, "Design patterns: Between programming and software design", In: *Proceedings of the 30th International Conference on Software Engineering*, pp. 801–804, Leipzig, Germany, May.
- DIKEL, D., KANE, D., ORNBURN, S., LOFTUS, W., WILSON, J., 1997, "Applying software product-line architecture", **Computer**, v. 30, n. 8 (Aug.) pp. 49-55.
- DONGSUN, K., SUNTAE, K., SEOKHWAN, K., SOOYONG, P., "Software engineering education toolkit for embedded software architecture design methodology using robotic systems", In: *Proceeding of the 15th Asia-Pacific Software Engineering Conference*, pp. 317–324, Beijing, China, Dec.
- FLUX DEVELOPER WIKI, Disponível em: <<http://wiki.developer.flux.com>>, Acesso em: 02 Out 2009.
- FREITAS, C. M. D. S., CHUBACHI, O. M., LUZZARDI, P. R. G., CAVA R. A., 2001, **Visualização de Informações**, RITA – Revista de Informática Teórica e Aplicada, v, 8, n. 2 (Oct.), pp. 143-158.
- FUHRMANN, A., PURGATHOFER, W., 2001, "Studierstube: An Application Environment for Multi-User Games in Virtual Reality", In: *Proceedings of the 31 Jahrestagung der Gesellschaft für Informatik*, pp. 1185-1190, Viena, Áustria, Sep.
- GAMMA, E., Helm, R., Johnson, R., Vlissides, F. M., 1994, **Design Patterns: Elements of Reusable Object-Oriented Software**, Baarn, Addison-Wesley.
- GARLAN, D, PERRY, D., 1995, "Introduction to the special issue on software architecture". **IEEE Transactions on Software Engineering**, v. 21, n. 4 (Apr.), pp. 269-274.

- GARLAN, D., SHAW, M., 1994, *An Introduction to Software Architecture*, In: Technical report CS-94-166, Carnegie Mellon University.
- GAST, H., 2008, "Patterns and traceability in teaching software architecture", In: *Proceedings of the 6th International Conference on Principles and Practice of Programming in Java (PPPJ 2008)*, pp. 23–31, Modena, Italy, Sep.
- GRIMSON, W., ETTINGER, G., KAPUR, T., LEVENTON, M., WELLS, W., KIKINIS, R., 1997, "Utilizing Segmented MRI Data in Image-guided Surgery", *International Journal of Pattern Recognition and Artificial Intelligence*, 11(8), pp. 1367-1397.
- GRISHAM, P. S., HAWTHORNE, M. J., PERRY, D. E., 2007, "Architecture and design intent: An experience report", In: *Proceedings of the Second Workshop on Sharing and Reusing architectural Knowledge Architecture, Rationale, and Design Intent (SHARK-ADI'07)*, pp 12-12, Minneapolis, USA, May.
- HEER, J., AGRAWALA, M., 2006, "Software design patterns for information visualization", *IEEE Transactions on Visualization and Computer Graphics*, v. 12, n. 5 (Sep.), pp. 853–860.
- HEIDRICH, F. E., CLARO, A., PEREIRA, A., 2003, "Simulação Computacional de Iluminação Natural através de Ambientes em VRML", In: *Anais do VII Congresso Ibero-americano de Gráfica Digital (Sigradi 2003)*, pp. 117-120, Rosario, Nov.
- HILBURN, T. B., TOWHIDNEJAD, M., 2007, "A Case for Software Engineering", In: *Proceedings of the 20th Conference on Software Engineering Education and Training (CSEET'2007)*, pp. 107-114, Dublin, Ireland, Jul.
- HOFMEISTER, C., NORD, R. L., SONI, D., 2005, "Global Analysis: Moving from software requirements specification to structural views of the software architecture". *IEE Proceedings Software*, v. 152, n. 4 (Aug.), pp. 187-197.
- HUANG, S., DISTANTE, D., 2006, "On Practice-Oriented Software Engineering Education" In: *Proceedings of the 19th Conference on Software Engineering Education and Training Workshops (CSEETW'06)*, pp. 15-15, North Shore Oahu, USA, Apr.
- JABREF versão 2.4.2, Disponível em: <<http://jabref.sourceforge.net/index.php>>, Acesso em: 04 Mai 2009.
- KATO, H., "ARToolkit", Human Interface Technology Laboratory, University of Washington, Disponível em: <<http://www.hitl.washington.edu/artoolkit>>, Acesso em: 05 Out 2009.
- KAZMAN, R., ABOWD, G., BASS, L., CLEMENTS, P., 1996, "Scenario-based analysis of software architecture", *IEEE Software*, v. 13, n. 6 (Nov.), pp. 47-55.

- KAZMAN, R., BASS, L., ABOWD, G., WEBB, M., 1994, "SAAM: a method for analyzing the properties of software architectures". In: *Proceedings of the 16th International Conference on Software Engineering (ICSE)*, pp. 81-90, Sorrento, Italy, May.
- KAZMAN, R., KLEIN, M., CLEMENTS, P., 2000, *ATAM: Method for Architecture Evaluation*, In: Technical Report CMU/SEI-2000-TR-004, Pittsburgh, PA.
- KIRNER, C., SISCOUTTO, R., 2007, "Fundamentos de Realidade Virtual e Aumentada", In: *Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações*, Editora SBC - Sociedade Brasileira de Computação, Porto Alegre, pp. 2-21.
- KIRNER, C., TORI, R., 2006, "Fundamentos de Realidade Aumentada". In: *Fundamentos e Tecnologia de Realidade Virtual e Aumentada*, v. 1, Editora SBC - Sociedade Brasileira de Computação, Porto Alegre, pp. 22-38.
- KÜMMEL, G. Z., 2007, *Uma Abordagem para a Criação de Arquiteturas de Referência de Domínio a partir da Comparação de Modelos Arquiteturais de Aplicações*, Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- LAGO, P., VAN VLIET, H., 2005, "Teaching a Course on Software Architecture" In: *Proceedings of the 18th Conference on Software Engineering Education and Training (CSEE&T2005)*, pp. 35-42, Ottawa, Canada, Apr.
- LEE, A. H., 2003, "A manageable web software architecture: Searching for simplicity", ***ACM SIGCSE Bulletin***, v. 35, n. 1 (Jan.), pp. 229-233.
- LIBÂNIO, J. C., 1990, Didática, São Paulo, Editora Cortez.
- LOPES, L. F. B., 2005, *O Estudo e a Implementação de Interfaces para Utilização em Sistemas de Realidade Aumentada*, Dissertação de M.Sc, Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, SP, Brasil.
- LUCKHAM, D., KENNEY, J., AUGUSTIN, L., VERA, J., BRYAN, D., MANN, W., 1995, "Specification and Analysis of System Architecture Using Rapide", ***IEEE Transactions on Software Engineering***, v. 21, n. 4 (Apr.), pp. 336-355.
- MAENNISTOE, T., SAVOLAINEN, J., MYLLÄRNIEMI, V., 2008, "Teaching software architecture design", In: *Proceedings of the 7th IEEE/IFIP Working Conference on Software Architecture (WICSA 2008)*, pp. 117-124, Vancouver, Canada, Feb.
- MAFRA, S. N., TRAVASSOS, G. H., 2006, *Estudos Primários e Secundários apoiando a busca por Evidência em Engenharia de Software*, Relatório Técnico ES-687/06, COPPE/UFRJ, Rio de Janeiro.

- MCGREGOR, J. D., BACHMAN, F., BASS, L., BIANCO, P., KLEIN, M., 2007, "Using an architecture reasoning tool to teach software architecture", In: *Proceedings of the 20th Conference on Software Engineering Education & Training*, pp. 275–282, Dublin, Ireland, Jul.
- MEDVIDOVIC, N., TAYLOR, R.N., 2000, "A Classification and Comparison Framework for Software Architecture Description Languages". **IEEE Transactions on Software Engineering**, v. 26 n. 1 (Jan.), pp. 70-93.
- MEIGUINS, S. B., BEHRENS, H. F., 1999, "Laboratório Virtual para Experiências de Eletrônica", In: *Anais do II Workshop Brasileiro de Realidade Virtual (WRV'99)*, pp. 56-67, Marília, Nov.
- MENDES, A., 2002, **Arquitetura de Software: Desenvolvimento Orientado para Arquitetura**, Rio de Janeiro, Editora Campus.
- MEYER, B., 2001, "Software engineering in the academy", **Computer**, v. 34, n. 5 (May.), pp. 28-35.
- MILER, N., 2000, *A Engenharia de Aplicações no Contexto da Reutilização baseada em Modelos de Domínio*, Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- MILGRAM, P., KISHINO, F., 1994, "A Taxonomy of Mixed Reality Visual Displays", **IEICE Transactions on Information Systems**, v. E77-D, n. 12 (Dec.), pp. 1321-1329.
- MONTEIRO, E – Nativos Digitais já estão dominando o mundo e transformando a forma como o ser humano se comunica. Reportagem do O Globo de 18/05/2009. Disponível em: <<http://oglobo.globo.com/tecnologia/mat/2009/05/18/nativos-digitais-ja-estao-dominandomundo-transformando-forma-como-ser-humano-se-comunica-755911408.asp>>. Acesso em: 11 set. 2009.
- MOHRENSCHILDT, M. V., PETERS, D. K., 1998, "Draw-Bot: A project for teaching software engineering", In: *Proceedings of the 28th Annual Frontiers in Education Conference*, pp. 1022-1027, Tempe, USA, Jun.
- MORAN, J. M., 2000, "Ensino e Aprendizagem Inovadores com Tecnologia", **Informática na Educação: Teoria & Prática**, v. 3, n.1 (Set.), pág. 137-144.
- MOURA, A. M. M., 2009, *ROOSC: Uma Abordagem de Reengenharia de Sistemas Orientados a Objetos para Componentes Baseada em Métricas*, Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- MURTA, L. G. P., 2006, *Gerência de Configuração no Desenvolvimento Baseado em Componentes*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

- MURTA, L. G. P., BARROS, M. O., WERNER, C. M. L., 2001, "FrameDoc: Um Framework para a Documentação de Componentes Reutilizáveis", In: *Proceedings of the IV International Symposium on Knowledge Management/Document Management (ISKM/DM'2001)*, pp.241-259, Curitiba, Brazil, Ago.
- NAVEDA, J. F., 1999, "Teaching architectural design in an undergraduate software engineering curriculum", In: *Proceedings of the 29th Annual Frontiers in Education Conference (FIE '99)*, pp. 12B1/1-12B1/4, San Juan, Puerto Rico, Nov.
- ODYSSEY, Projeto Odyssey, Disponível em: <<http://reuse.cos.ufrj.br/odyssey>>, Acesso em: 11 set. 2009.
- OMG, 2005, XML Metadata Interchange (XMI) Specification. Version 2.0, Object Management Group.
- PAI, M., MCCULLOCH, M., GORMAN, J., PAI, N., ENANORIA, W., KENNEDY, G., THARYAN, P., COLFORD Jr., J., 2004, "Systematic reviews and meta-analyses: An illustrated step-by-step guide", **The National Medical Journal of India**, v. 17, n. 2 (Mar.), pp: 86-95.
- PAVLOVIC, V.I., SHARMA, R., HUANG, T.S., 1997, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review", **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 19, n. 7 (Jul.), pp. 677-695.
- PERRY, D. E., WOLF, A. L., 1992, "Foundations for the Study of Software Architecture", **ACM SigSoft Software Engineering Notes**, v. 17, n. 4 (Oct.), pp. 40-52.
- PIEKARSKI, W., THOMAS, B., 2002, "ARQuake: The Outdoor Augmented Reality Gaming System", **Communications of the ACM**, v. 45, n. 1 (Jan.), pp. 36-38.
- PILATTI, L., PRIKLADNICKI, R., AUDY, J. L. N. , 2007, "Avaliando os Impactos dos Aspectos Não-Técnicos da Engenharia de Software em Ambientes de Desenvolvimento Global de Software: Um Caso Prático", In: *Proceeding of the III Workshop Um Olhar Sócio-Técnico sobre a Engenharia de Software (WOSES 07), VI SBQS - Simpósio Brasileiro de Qualidade de Software*, pp. 85-96, Porto de Galinhas, Brazil.
- PINHO, M., 2000, "Interação em Ambientes Tridimensionais", In: *Tutorial of the 3rd Workshop on Virtual Reality (WRV'2000)*, Gramado, Out.

- POUR, G., 2006, "Software Engineering for Pervasive Computing: An Outlook for Educational Reform," In: *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering - Software Architecture and Reuse (ICIS-COMSAR'06)*, pp. 168-173, Honolulu, USA, Jul.
- REUSE. Equipe de Reutilização de Software, Disponível em: <<http://reuse.cos.ufrj.br>>. Acesso em: 11 set. 2009.
- RODRIGUES, C. S. C., RODRIGUES, P. F. N., WERNER, C. M. L., 2008, "An Application of Augmented Reality in Architectural Education for Understanding Structural Behavior through Models", In: *Proceedings of the X Symposium on Virtual and Augmented Reality (SVR2008)*, pp. 163-166, João Pessoa, USA, May.
- RODRIGUES, C. S. C., WERNER, C. M. L., 2009a, *Uma Revisão Sistemática sobre as Iniciativas Realizadas no Ensino de Arquitetura de Software*, Relatório Técnico, ES-728/09, COPPE/UFRJ, Rio de Janeiro.
- RODRIGUES, C. S. C., WERNER, C. M. L., 2009b, "VisAr3D: Uma abordagem baseada em Realidade Aumentada para o Ensino de Arquitetura de Software", In: *Proceedings of the XII Conferencia Iberoamericana de Ingeniería de Requisitos Ambientales de Software (IDEAS 2009)*, pp. 361-372, Medellín, Abr.
- SCHAUER, R., KELLER, R. K., 1998, "Pattern visualization for software comprehension", In: *Proceedings of the 6th International Workshop on Program Comprehension*, pp. 4-12, Ischia, Italy, Jun.
- SCHMALSTIEG, D., "Studierstube: Augmented Reality Project". Disponível em: <<http://www.studierstube.org>>, Acesso em: 5 Out 2009.
- SCHOTS, M., MURTA, L. G. P., WERNER, C. M. L., 2009, "PREVIA: Uma Abordagem para a Representação Visual da Evolução de Modelos Arquiteturais de Software", In: *Anais do Workshop de Teses e Dissertações (WTES) do Simpósio Brasileiro de Engenharia de Software (SBES)*, Fortaleza, Out.
- SECOND LIFE, "Second Life: Your World. Your Imagination", Disponível em: <<http://secondlife.com>>, Acesso em: 05 Out 2009.
- SEI – Software Engineering Institute, Disponível em: <<http://www.sei.cmu.edu/>>, Acesso em: 17 Set 2009.
- SHAW, M., 2000, "Software Engineering Education: A Roadmap". In: *Proceedings of the Conference on The Future of Software Engineering*, pp. 371-380, Limerick, Ireland, Jun.
- SHAW, M., GARLAN, D., 1996, **Software Architecture: Perspectives on an Emerging Discipline**, Upper Saddle River, NJ, USA, Prentice Hall Publishing.

- SHULL, F., CARVER, J., TRAVASSOS, G. H., 2001, "An empirical methodology for introducing software processes", SIGSOFT Software Engineering Notes, v. 26, n. 5 (Set.), pp. 288-296.
- SISCOUITTO, R. A., RAPOSO, A. B., TORI, R., SZENBERG, F., CELES, W., GATTASS, M., 2006, Estereoscopia, In: *Fundamentos e Tecnologia de Realidade Virtual e Aumentada*, v. 1, Editora SBC - Sociedade Brasileira de Computação, Porto Alegre, pp. 221-245.
- STANEK, W. R., 1996, HTML, Java, CGI, VRML, SGML Web Publishing, Sams.net Publishing.
- SUTHERLAND, I., 1968, "A head-mounted three dimensional display", In: *Proceedings of the AFIPS Fall Joint Computer Conference*, pp.757-764, San Francisco, USA, Dec.
- SVAHNBERG, M., MAERTENSSON, F., 2007, "Six years of evaluating software architectures in student projects," *Journal of Systems and Software*, v. 80, n. 11 (Nov.), pp. 1893-1901.
- THOMPSON, J. B., EDWARDS, H., 2009, "Preparing Graduate Students for Industry and Life Long Learning: - A Project Based Approach", In: *Education and Technology for a Better World*, New York: Springer, pp. 292-30.
- TRONT, J. G., ELIGETI, V., PREY, J., 2006, "WriteOn: A tool to support teaching software engineering", In: *Proceedings of the 19th Conference on Software Engineering Education and Training Workshops (CSEETW'06)*, pp. 8-8, Turtle Bay, USA, Apr.
- UNICON: An architectural description language, Disponível em: <<http://www.cs.cmu.edu/afs/cs/project/vit/www/unicon/index.html>>, Acesso em: 19 Set 2009.
- VALLIESWARAN, V., MENEZES, B., 2007, "ArchKriti: A software architecture based design and evaluation tool suite", In: *Proceedings of the Fourth International Conference on Information Technology-New Generations (ITNG '07)*, pp. 701-706, Las Vegas, USA, Apr.
- VARMA, V., GARG, K., 2005, "Case studies: The potential teaching instruments for software engineering education", *Proceedings of the Fifth International Conference on Quality Software (QSIC 2005)*, pp. 279-284, Melbourne, Australia, Sep.
- VASCONCELOS, A., 2007, *Uma Abordagem de apoio à Criação de Arquiteturas de Referência de Domínio baseadas na Análise de Sistemas Legados*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

- VESTAL, S., 1994, "Mode Changes in a Real-Time Architecture Description Language", *Proceedings of the 2nd. International Workshop on Configurable Distributed Systems*, pp. 136-146, Pittsburgh, USA, Mar.
- VRML97, Disponível em: <<http://www.web3d.org/x3d/specifications/vrml/>>, Acesso em: 06 Out 2009.
- WAGNER, D., PINTARIC, T., LEDERMANN, F., SCHMALSTIEG, D., 2005, "Towards Massively Multi-User Augmented Reality on Handheld Devices", In: *Proceedings of the 3rd International Conference on Pervasive Computing*, pp. 208-219, Munich, Germany, May.
- WANG, A. I., ARISHOLM, E., JACCHERI, L., 2007, "Educational approach to an experiment in a software architecture course", In: *Proceedings of the 20th Conference on Software Engineering Education & Training*, pp. 291-298, Dublin, Ireland, Jul.
- WANG, A. I., SOERENSEN, C. F., 2006, "Writing as a tool for learning software engineering", In: *Proceedings of the 19th Conference on Software Engineering Education & Training*, pp. 35-42, Turtle Bay, USA, Apr.
- WANG, A. I., STAELHANE, T., 2005, "Using post mortem analysis to evaluate software architecture student projects", In: *Proceedings of the 18th Conference on In Software Engineering Education and Training*, pp. 43-50, Ottawa, Canada, Apr.
- WANG, W. L.; SCANNELL, D., 2005, "An architecture-based software reliability modeling tool and its support for teaching", In: *Proceedings of the Frontiers in Education Conference (FIE)*, pp. T4C-15, Indianapolis, USA, Oct.
- WERNER, C., MANGAN, M., MURTA, L., PINHEIRO, R., MATTOSO, M., BRAGA, R., BORGES, M., 2003, "OdysseyShare: an Environment for Collaborative Component-Based Development", In: *Proceedings of the IEEE Conference on Information Reuse and Integration (IRI)*, pp. 61-68, Las Vegas, USA, Oct.
- WRIGHT: The Wright Architecture Description Language, Disponível em: <<http://www.cs.cmu.edu/afs/cs/project/able/www/wright/index.html>>, Acesso em: 19 Set 2009.
- X3D, Disponível em: <<http://www.web3d.org/x3d/>>, Acesso em: 06 Out 2009.
- XADL: XADL 2.0, Disponível em: <<http://www.isr.uci.edu/projects/xarchuci/>>, Acesso em: 19 Set 2009.
- XAVIER, J. R., 2001, *Criação e Instanciação de Arquiteturas de Software Específicas de Domínio no Contexto de uma Infraestrutura de Reutilização*, Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.