

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO

**Meta-modelo para o Processo de Sistemas com RV
Pautado por Enfoque no Usuário, Iteratividade de Projeto
e Critérios de Usabilidade**

Milena Marquezin Olher

**São Carlos
Maio/2004**

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

O45mm

Olher, Milena Marquezin.

Meta-modelo para o processo de sistemas com RV
pautado por enfoque no usuário, iteratividade de projeto e
critérios de usabilidade / Milena Marquezin Olher. -- São
Carlos : UFSCar, 2004.

98 p.

Dissertação (Mestrado) -- Universidade Federal de São
Carlos, 2004.

1. Engenharia de software. 2. Interação homem-máquina.
3. Realidade virtual. 4. Processo de desenvolvimento para
ambientes virtuais. 5. Usabilidade. I. Título.

CDD: 005.1(20^a)

“Quem tem coragem para pensar tem coragem para fazer tudo.”

Alfred Sherman

Agradecimentos

Agradeço a Deus por me fazer capaz de realizar este trabalho e a minha história.

Agradeço aos meus pais, Eduardo e Celina, por me ensinar que posso não apenas escolher os caminhos a percorrer em meio a aqueles que surgem logo à frente, mas posso também criar os caminhos a percorrer em meio aos sonhos que quero realizar, tornando real este trabalho.

Agradeço aos meus irmãos, Du e Me, por me mostrar o significado da cumplicidade.

Agradeço ao meu lindinho, Marcellus, por compreender e compartilhar carinhosamente importantes momentos deste trabalho e meus, tornando esses momentos verdadeiramente marcantes.

Agradeço à minha amiga Lu, por me ensinar o valor da amizade.

Agradeço ao meu amigo Eugênio, por incentivar minhas decisões.

Agradeço às minhas amigas Taci e Dri e aos meus amigos Lucas e Matheus por responder minhas dúvidas.

Agradeço ao meu exemplo profissional e pessoal, Júnia, por assumir e apoiar pacientemente importantes desafios deste trabalho e meus, tornando esses desafios inquestionavelmente menores.

Enfim, todos vocês fazem parte não apenas deste trabalho, mas da minha história.

Muito obrigada!

Resumo

O crescimento contínuo da área de Interação Humano-Computador tem causado o surgimento de abordagens para melhor corresponder às necessidades dos usuários, concernentes não somente aos fatores de funcionalidade dos sistemas interativos, os quais são o principal foco da área de Engenharia de Software, mas também aos fatores de interatividade desses sistemas.

Nesse contexto, a tecnologia de Realidade Virtual se apresenta como uma proposta de mudança para o desenvolvimento de interfaces mais avançadas, no que diz respeito à interação humano-computador, oferecendo modos de interação mais inerentes aos seres humanos para atender às expectativas dos usuários de sistemas interativos por uma interação mais natural e intuitiva, bem como envolvendo recursos e estudos nos mais diversos ramos do conhecimento.

Dessa forma, embora tenha evoluído significativamente, seu emprego em sistemas interativos ainda apresenta desafios a suplantar, dentre eles analisar e melhor formalizar o processo de sistemas com Realidade Virtual nas práticas das áreas de Interação Humano-Computador e Engenharia de Software. Porém, pesquisas que detalhem o processo de sistemas com Realidade Virtual de tal forma que seus benefícios sejam explorados são pouco encontradas na literatura.

Este trabalho propõe um meta-modelo para o processo de sistemas com Realidade Virtual, identificando as especificidades desses sistemas e as implicações existentes no seu processo, apresentando e analisando criticamente as metodologias citadas na literatura para o processo de sistemas com Realidade Virtual, bem como identificando os modelos de processo da Engenharia de Software e as abordagens da Interação Humano-Computador que melhor atendem às implicações identificadas no processo desses sistemas, as quais se apresentam como enfoque no usuário, iteratividade de projeto e usabilidade.

Abstract

The continuous growth of the Human-Computer Interaction area has been causing the appearance of approaches to better correspond to the users' needs, concerning not only the functionality factors of the interactive systems, which are the main focus of the Software Engineering area, but also the interaction factors of those systems.

In that context, the Virtual Reality technology comes as a change proposal for the development of more advanced interfaces, with regard to the human-computer interaction, offering means of interaction that are more natural to humans to supply to the interactive system users' expectations by a more natural and intuitive interaction, as well as involving resources and studies in the most several fields of the knowledge.

In that way, although it has evolved significantly, its usage in interactive systems still presents challenges to overcome, among which a better analysis and formalization of the process of systems with Virtual Reality in the practices of the Human-Computer Interaction and Software Engineering areas. However, researches that detail the process of systems with Virtual Reality so that its benefits are explored are little found in the literature.

This work proposes a meta-model for the process of systems with Virtual Reality, identifying the differential of those systems and the existing implications in your process, presenting and critically analyzing the methodologies mentioned in the literature for the process of systems with Virtual Reality, as well as identifying the process models in Software Engineering and the approaches in Human-Computer Interaction that best assist the identified implications in the process of those systems, which focus on the user, project iteration and usability.

Sumário

1. Introdução	1
1.1. Motivações e Objetivos deste Trabalho.....	1
1.2. Definições	2
1.3. Organização deste Trabalho	3
2. Implicações no Processo de Sistemas com RV	4
2.1. Considerações Iniciais	4
2.2. Conceitos Básicos de RV.....	4
2.3. Sistemas com RV.....	6
2.4. Enfoque no Usuário no Contexto de RV	9
2.5. Iteratividade de Projeto no Processo de Sistemas com RV.....	11
2.6. Usabilidade no Contexto de RV	12
2.6.1. Critérios de Usabilidade para RV.....	13
2.7. Considerações Finais	15
3. Metodologias para Sistemas com RV	16
3.1. Considerações Iniciais	16
3.2. Modelo e Metodologia VRID	17
3.3. Metodologia CLEVR.....	20
3.4. Metodologia de Stuart.....	22
3.5. Metodologia de Scaife e Rogers	24
3.6. Análise Crítica e Comparativa das Metodologias	27
3.7. Considerações Finais	30
4. Discussão das Metodologias para Sistemas com RV com seus Proponentes.....	31
4.1. Considerações Iniciais	31
4.2. Modelo e Metodologia VRID	32
4.3. Metodologia CLEVR.....	33
4.4. Metodologia de Stuart.....	36
4.5. Metodologia de Scaife e Rogers	40
4.6. Considerações Finais	42
5. Meta-modelo para o Processo de Sistemas com RV	44
5.1. Considerações Iniciais	44
5.2. Meta-modelo com Enfoque no Usuário e Iteratividade de Projeto	44

5.2.1. Contribuições da Metodologia VRID.....	44
5.2.2. Contribuições da Metodologia CLEVR.....	45
5.2.3. Contribuições da Metodologia de Stuart.....	47
5.2.4. Contribuições da Metodologia de Scaife e Rogers	48
5.2.5. Interpretação e Representação do Meta-modelo com Enfoque no Usuário e Iteratividade de Projeto	50
5.3. Meta-modelo com Enfoque no Usuário, Iteratividade de Projeto e Critérios de Usabilidade.....	58
5.3.1. Critérios de Usabilidade Incorporados à Etapa de Análise de Requisitos.....	59
5.3.2. Critérios de Usabilidade Incorporados à Etapa de Projeto (Baseado em Prototipação).....	59
5.3.3. Critérios de Usabilidade Incorporados às Etapas de Avaliação de Requisitos e Projeto, de Implementação e de Avaliação de Sistema	61
5.3.4. Interpretação e Representação do Meta-modelo com Enfoque no Usuário, Iteratividade de Projeto e Critérios de Usabilidade.....	61
5.4. Considerações Finais	62
6. Conclusões	64
6.1. Considerações Iniciais	64
6.2. Resultados Obtidos	64
6.3. Desafios a Suplantar	66
6.4. Trabalhos Futuros	68
Referências Bibliográficas	70
Apêndice A	76
Anexo A	77
Anexo B.....	80
Anexo C	86
Anexo D	96

Lista de Figuras

Figura 2.1 - Sistema com RV	6
Figura 3.1 - Modelo VRID (Extraída de [Tanriverdi et al 2001]).....	17
Figura 3.2 - Metodologia VRID (Adaptada de [Tanriverdi et al 2001]).....	19
Figura 3.3 - Metodologia CLEVR (Adaptada de [Seo et al 2002]).....	21
Figura 3.4 - Metodologia de Stuart (Adaptada de [Stuart 1996]).....	23
Figura 3.5 - Metodologia de Scaife e Rogers (Adaptada de [Scaife et al 2001]).....	25
Figura 4.1 - Nova Interpretação e Representação para a Metodologia CLEVR	35
Figura 4.2 - Nova Interpretação e Representação para a Metodologia de Stuart	39
Figura 4.3 - Nova Interpretação e Representação para a Metodologia de Scaife e Rogers.....	41
Figura 5.1 - Etapas Iniciais do Meta-modelo para Sistemas com RV com Enfoque no Usuário e Iteratividade de Projeto	54
Figura 5.2 - Etapas Finais do Meta-modelo para Sistemas com RV com Enfoque no Usuário e Iteratividade de Projeto	56
Figura 5.3 - Meta-modelo para Sistemas com RV com Enfoque no Usuário e Iteratividade de Projeto.....	58
Figura 5.4 - Meta-modelo para Sistemas com RV com Enfoque no Usuário, Iteratividade de Projeto e Critérios de Usabilidade	62
Figura 6.1 - Etapas de Pesquisa com seus Artefatos	65

Lista de Tabelas

Tabela 2.1 - Tabela Comparativa das Características Visuais, Físicas, Comportamentais, de Comunicações e Interações de Interfaces Convencionais e de Interfaces com RV.....	7
Tabela 2.2 - Tabela Comparativa das Características de Diálogos de Entrada e Saída, de Requisitos e de Monitoramento de Comandos de Interfaces Convencionais e de Interfaces com RV	8
Tabela 3.1 - Tabela Comparativa das Fundamentações no Processo das Metodologias para Sistemas com RV.....	28
Tabela 3.2 - Tabela Comparativa das Implicações no Processo das Metodologias para Sistemas com RV.....	29
Tabela 5.1 - Contribuições da Metodologia VRID para Compor o Meta-modelo.....	45
Tabela 5.2 - Contribuições da Metodologia CLEVR para Compor o Meta-modelo.....	46
Tabela 5.3 - Contribuições da Metodologia de Stuart para Compor o Meta-modelo	47
Tabela 5.4 - Contribuições da Metodologia de Scaife e Rogers para Compor o Meta-modelo	49
Tabela 5.5 - Síntese das Contribuições das Metodologias para Sistemas com RV para Compor o Meta-modelo para Sistemas com RV	51

Lista de Acrônimos

2D	Bidimensional
3D	Tridimensional
AV	Ambiente Virtual
CLEVR	<i>Concurrent and Level by Level Development of VR Systems</i>
DP	Design Participativo
ES	Engenharia de Software
GaCIV	Gabaritos Configuráveis para Interfaces Virtuais
HMD	<i>Head-Mounted Display</i>
IHC	Interação Humano-Computador
PCU	Projeto Centrado no Usuário
RV	Realidade Virtual
VRID	<i>Virtual Reality Interface Design</i>
WIMP	<i>Windows, Icons, Menus and a Pointing Device</i>

1. Introdução

1.1. Motivações e Objetivos deste Trabalho

A área de Interação Humano-Computador, IHC (em inglês, HCI, *Human-Computer Interaction*), está direcionada à concepção, projeto, implementação e avaliação de sistemas computacionais interativos, referenciados neste trabalho como sistemas interativos, para uso humano, bem como ao estudo dos principais fenômenos nos quais estão envolvidos [Hewett et al 1996].

O crescimento contínuo dessa área tem motivado o aparecimento de diversas abordagens para melhor corresponder às necessidades dos usuários, relacionadas não somente com fatores de funcionalidade de sistemas interativos, os quais são o principal foco da área de Engenharia de Software, ES (em inglês, SE, *Software Engineering*), mas também com fatores de interatividade desses sistemas. Considerando que a usabilidade em sistemas interativos é caracterizada, basicamente, pela forma como o usuário interage com o sistema, acredita-se que sua interface com o usuário, referenciada neste trabalho como interface, é o principal componente responsável por tornar a interação humano-computador, referenciada neste trabalho como interação, satisfatória e consistente, sendo fator determinante para seu sucesso ou fracasso [Assis 2001] [Assis et al 2000] [Nielsen 1993] [Smith et al 2001].

Nesse contexto, a tecnologia de Realidade Virtual, RV (em inglês, VR, *Virtual Reality*), se apresenta como uma proposta de mudança para o desenvolvimento de interfaces mais avançadas no que diz respeito à interação. Com aplicação na maioria das áreas do conhecimento, essa tecnologia objetiva oferecer modos de interação mais inerentes aos seres humanos, para, desse modo, atender às expectativas dos usuários de sistemas interativos por uma interação mais natural e intuitiva [Appel et al 1999] [Pinho et al 1997].

A tecnologia de RV envolve, em seu desenvolvimento, recursos e estudos nos mais diversos ramos do conhecimento [Myers et al 1996] [Seo et al 2002].

Desse modo, embora tenha evoluído significativamente nos últimos anos, seu emprego em sistemas interativos ainda apresenta inúmeros desafios a suplantarem, dentre eles analisar e melhor formalizar o processo de sistemas com RV. Da mesma forma que a tecnologia de RV atinge maior nível de maturidade, existe a necessidade de entender como essa tecnologia pode ser acomodada na prática da ES [Smith et al 1999a] e da IHC [Olher et al 2003]. Porém, pesquisas que detalhem o processo de sistemas com RV de tal forma que seus potenciais

benefícios sejam realmente explorados, são pouco encontradas na literatura [Scaife et al 2001] [Smith et al 2001].

A partir dessa motivação, este trabalho tem como objetivo construir um meta-modelo para o processo de sistemas com RV, verificando as especificidades desses sistemas, identificando as implicações existentes no processo de sistemas com RV, apresentando e analisando criticamente as metodologias citadas na literatura para o processo desses sistemas e verificando os modelos de processo da ES e as abordagens da IHC nas quais estão fundamentadas, bem como identificando os modelos de processo da ES e as abordagens da IHC que melhor atendem às implicações existentes no processo de sistemas com RV, as quais se apresentam como enfoque no usuário, iteratividade de projeto e usabilidade.

1.2. Definições

Diversos termos concernentes às áreas de ES, IHC e RV são bastante abrangentes e amplamente utilizados por pesquisadores, acadêmicos e desenvolvedores, os quais, muitas vezes, definem esses termos baseados em suas próprias experiências.

Neste trabalho, de acordo com as considerações realizadas por Brown (1997) e Pressman (2001), adota-se as seguintes definições:

- Sistema de software, referenciado neste trabalho como sistema, é a junção de programas de computador que quando executados provêm as funções e o desempenho desejados, estruturas de dados que permitem que os programas manipulem adequadamente as informações e documentos que descrevem as operações e a utilização dos programas de computador.
- Processo de sistemas é a estrutura de atividades da ES comuns e aplicáveis para todos os sistemas, composto pelas fases genéricas de definição, desenvolvimento e manutenção.
- Modelos de processo de sistemas da ES são estratégias de desenvolvimento que contemplam as fases genéricas que compõem o processo de sistema.
- Abordagens da IHC são fundamentos de desenvolvimento que apresentam o enfoque em aspectos que compõem a interação humano-computador.
- Metodologia é uma descrição, geralmente através de fases, passos ou etapas, para o desenvolvimento de sistemas, a qual pode estar fundamentada sobre modelos de processo da ES e ou em abordagens da IHC.

- Meta-modelo para sistemas com RV é uma estrutura de etapas e atividades comuns e aplicáveis para todos os modelos de processo de sistemas com RV, composta através do estudo das metodologias para esses sistemas citadas na literatura e dos modelos de processo da ES e das abordagens da IHC nas quais essas metodologias estão fundamentadas, que quando seguidas direcionam para a obtenção de modelos de processo de sistemas com RV ou desses sistemas de tal forma que apresentem qualidade, atendendo não somente às exigências da ES encontradas no processo de sistemas com RV, mas também às exigências da IHC.

Este estudo possui como seu principal objeto de pesquisa o processo de sistemas de software interativos com interfaces baseadas na tecnologia de RV, referenciados neste trabalho como sistemas com RV.

No item 1.2 são discutidos as motivações e os objetivos deste estudo. O item 1.3 apresenta a organização deste trabalho.

1.3. Organização deste Trabalho

Este trabalho está organizado em sete capítulos, a contar com este, os quais procuram abranger desde as implicações que surgem no processo de sistemas com RV à construção de um meta-modelo para o processo de sistemas com RV que contemple essas implicações.

O capítulo 2 deste trabalho é destinado à apresentação das implicações no processo de sistemas com RV, sendo realizada a contextualização da tecnologia de RV e sendo identificada a importância do enfoque no usuário, da iteratividade de projeto e dos critérios de usabilidade para o processo de sistemas com RV. No capítulo 3 são mostradas e analisadas as metodologias para sistemas com RV encontradas na literatura. O capítulo 4 deste trabalho é destinado à discussão com os proponentes das metodologias para sistemas com RV mostradas e analisadas, sendo realizada a sugestão de novas interpretações e representações para essas metodologias. No capítulo 5 são compiladas as contribuições das metodologias para sistemas com RV discutidas para a construção do meta-modelo para sistemas com RV com base nessas metodologias e nos modelos de processo da ES e nas abordagens da IHC nas quais estão fundamentadas, bem como nas exigências encontradas no processo de sistemas com RV e nos modelos de processo da ES e nas abordagens da IHC que melhor atendam a essas exigências.

2. Implicações no Processo de Sistemas com RV

2.1. Considerações Iniciais

Sendo o processo de sistemas com RV o principal objeto de pesquisa deste trabalho, este capítulo se destina à contextualização da tecnologia de RV, considerando a problemática envolvida no processo de sistemas com RV através da identificação de exigências em seu processo.

Inicialmente, no item 2.2 é apresentada a conceituação da tecnologia de RV, bem como as principais características de sistemas com RV. O enfoque no usuário no contexto de RV é discutido no item 2.3. No item 2.4 é apresentada a necessidade de iteratividade de projeto em sistemas com RV. A usabilidade no contexto de RV é discutida no item 2.5. Por fim, no item 2.6 são realizadas algumas considerações finais.

2.2. Conceitos Básicos de RV

Segundo Pimentel et al (1995), Stuart (1996) e Vince (1995), o termo “realidade virtual” é creditado a Jaron Lanier, que no final dos anos 1980 sentiu a necessidade de um termo para distinguir os mundos digitais imersivos que estava tentando criar das simulações tradicionais geradas por computador. Por ser um termo bastante abrangente e amplamente utilizado, pesquisadores, acadêmicos e desenvolvedores muitas vezes procuram definir RV baseados em suas próprias experiências.

Como uma definição genérica, RV refere-se à experiência imersiva e interativa baseada em imagens gráficas 3D geradas por computador em tempo real [Pimentel et al 1995].

Em uma definição mais refinada, RV refere-se ao uso de sistemas de computação gráfica combinados com dispositivos de interface para prover a sensação de imersão em um ambiente tridimensional interativo gerado por computador, chamado ambiente virtual, referenciado neste trabalho como AV (em inglês, VE, *Virtual Environment*), no qual os objetos virtuais têm presença espacial [Earnshaw et al 1995]. Por imersão entende-se como a sensação de que o ponto de vista do usuário ou alguma parte de seu corpo está contido dentro do espaço gerado por computador, ou seja, é a sensação provida ao usuário para que ele se sinta parte do AV [Burdea et al 1994] [Earnshaw et al 1995] [Vince 1995]. Por presença

espacial entende-se que os objetos gerados por computador no AV têm uma posição definida no espaço tridimensional em relação ao usuário [Earnshaw et al 1995]. Nota-se que um ambiente é aquilo que cerca o usuário, o conjunto de condições e objetos que são percebidos por ele e com os quais ele pode interagir [Stuart 1996] e, dessa forma, um AV pode ser projetado para simular tanto ambientes reais quanto ambientes imaginários [Pinho et al 1997].

De acordo com as definições realizadas por Burdea et al (1994) e Pimentel et al (1995), a tecnologia de RV pode ser considerada como uma integração de três características fundamentais: imersão, interação e imaginação. Além dessas características, outras três citadas em [Bowman et al 2001], [Stuart 1996] e [Vince 1995], podem ser incorporadas a esse conjunto: envolvimento, navegação e resposta ao usuário. Nota-se que essas idéias, quando isoladas, não são exclusivas da RV, mas no contexto dessa tecnologia elas devem coexistir.

As características que fundamentam a tecnologia de RV são descritas a seguir:

- **Imersão:** relacionada à sensação do usuário de fazer parte do AV [Burdea et al 1994];
- **Interação:** referente à capacidade do computador em detectar as entradas do usuário e modificar o AV e as possíveis ações sobre ele em tempo real [Burdea et al 1994] [Pimentel et al 1995];
- **Imaginação:** relacionada à criatividade humana para ajustar a RV nos mais diversos ramos do conhecimento [Burdea et al 1994];
- **Envolvimento:** referente ao grau de motivação para o engajamento do usuário na realização de uma determinada tarefa no AV [Stuart 1996] [Vince 1995];
- **Navegação:** relacionada à liberdade do usuário para caminhar no AV [Bowman et al 2001];
- **Resposta ao usuário:** relacionada ao retorno, geralmente através da visão e audição, provido pelo sistema em resposta às ações do usuário. Esse retorno também pode ocorrer através do tato, olfato e ou paladar [Bowman et al 2001].

Devido às suas características, a tecnologia de RV possibilita que o usuário entre no AV e visualize, manipule e explore os dados da aplicação em tempo real de forma mais diversificada quanto à utilização dos sentidos humanos, permitindo que o conhecimento intuitivo do usuário a respeito do mundo real possa ser transferido para manipular o AV [Pinho et al 1997].

Desse modo, a tecnologia de RV configura-se como uma proposta de mudança para o desenvolvimento de interfaces avançadas. Com aplicação na maioria das áreas do conhecimento, e com um grande investimento das indústrias na produção de hardware,

software e dispositivos de interface, essa tecnologia objetiva oferecer modos de interação inerentes aos seres humanos, atendendo as expectativas dos usuários de sistemas interativos por uma interação mais natural e intuitiva [Appel et al 1999] [Pinho et al 1997].

2.3. Sistemas com RV

Sistemas com RV podem ser considerados sob diferentes pontos de vista e níveis de detalhamento. De acordo com as considerações realizadas por Pinho et al (1997) e Tanriverdi et al (2001), neste estudo considera-se que um sistema com RV, representado na figura 2.1, consiste basicamente de quatro componentes: usuário, interface, controle de diálogo e aplicação.

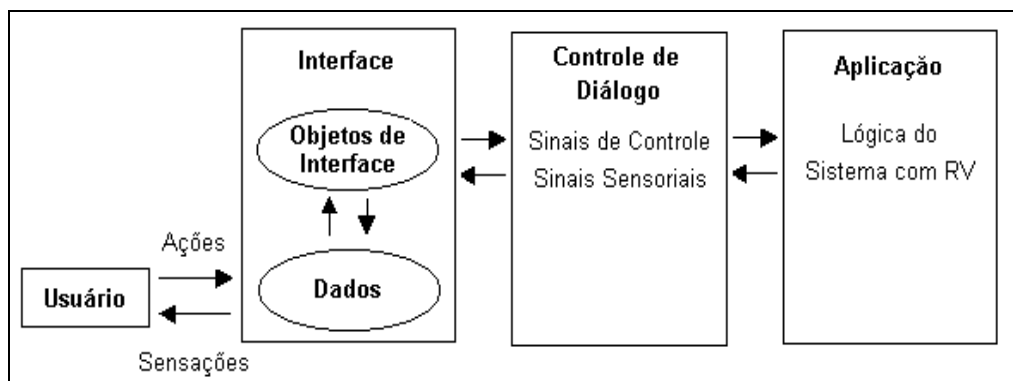


Figura 2.1 - Sistema com RV

Conforme a figura 2.1, através da interface o usuário e outras entidades externas (como outros sistemas) manipulam e trocam informações com o sistema com RV. A interface consiste de objetos de interface e dados. Objetos de interface referem-se às entidades na interface que possuem regras e identidades bem definidas. Dados referem-se às entradas recebidas do usuário ou de outras entidades externas e às saídas geradas pelo sistema. O controle de diálogo permite a comunicação entre a interface e a aplicação. A aplicação contém características, regras e conhecimento que definem a lógica do sistema com RV [Tanriverdi et al 2001].

Sistemas com RV apresentam interfaces semelhantes às interfaces convencionais no sentido em que são os meios pelos quais os usuários manipulam e trocam informações com esses sistemas. Entretanto, imagens gráficas tridimensionais geradas por computador associadas a dispositivos de interface utilizados em interfaces de sistemas com RV,

referenciadas neste trabalho como interfaces com RV, permitem que a interação seja realizada de forma mais inerente aos usuários, ou seja, no plano tridimensional e de forma mais diversificada quanto à utilização dos sentidos humanos.

Considerando os recursos que interfaces com RV devem apresentar para prover a interação proposta, diferentes características visuais, físicas, comportamentais, de comunicações e de interações, como interpretadas por este trabalho com base no trabalho de Tanriverdi et al (2001) e apresentadas na tabela 2.1, são exibidas por essas interfaces se comparadas com interfaces convencionais.

Tabela 2.1 - Tabela Comparativa das Características Visuais, Físicas, Comportamentais, de Comunicações e Interações de Interfaces Convencionais e de Interfaces com RV

Características	Interfaces Convencionais	Interfaces com RV
Visuais	Dispositivos e Objetos <i>2D</i>	Dispositivos e Objetos <i>2D</i> e <i>3D</i>
Físicas	Objetos <i>Virtuais</i>	Objetos <i>Virtuais</i> e <i>Físicos</i>
Comportamentais	Objetos <i>Passivos</i>	Objetos <i>Passivos</i> e <i>Ativos</i>
Comunicações	<i>Simples</i> entre Objetos	<i>Complexas</i> entre Objetos
Interações	<i>Explícitas</i>	<i>Explícitas</i> e <i>Implícitas</i>

Pela tabela 2.1, nota-se que enquanto interfaces convencionais apresentam principalmente objetos 2D, utilizando dispositivos de exibição 2D, interfaces com RV apresentam objetos 2D e, principalmente, 3D, utilizando dispositivos de exibição 3D. Interfaces convencionais contêm tipicamente objetos virtuais gerados por computador, ao passo que interfaces com RV contêm objetos virtuais e físicos, que podem coexistir nessas interfaces trocando, de modo mais complexo, informações entre si. Quanto às suas características comportamentais, enquanto objetos de interfaces convencionais normalmente exibem comportamentos passivos (comportamentos predeterminados que são ativados em resposta às ações dos usuários), objetos de interfaces com RV exibem comportamentos passivos e ativos (comportamentos autônomos e comunicações mais complexas entre si que podem mudar seus estados e afetar seus comportamentos). Ainda pela tabela 2.1, observa-se que interfaces convencionais suportam interações explícitas (como comandos do usuário através do uso de teclados), ao passo que interfaces com RV suportam interações explícitas e, principalmente, implícitas (como movimentos das mãos do usuário através do uso de luvas). Segundo Tanriverdi et al (2001), dispositivos e objetos 2D e 3D, tipos de objetos virtuais e físicos, comportamentos de objetos passivos e ativos, comunicações mais complexas entre objetos e interações explícitas e implícitas apresentados em interfaces com RV, apesar de

permitir que o processo de interação entre usuário e sistema seja mais natural e intuitivo, são mais complexos de ser combinados, projetados e implementados. Desse modo, a construção de interfaces com RV se torna mais difícil para projetistas e desenvolvedores.

Sistemas com RV apresentam interfaces consideradas Pós-WIMP (em inglês, *Post-WIMP, Post Windows, Icons, Menus and a Pointing Device*) [Van Dam 1997] ou Não-WIMP (em inglês, *Non-WIMP, Non Windows, Icons, Menus and a Pointing Device*) [Jacob et al 1999] que provêm estilo de interação humano-computador paralelo e contínuo com o usuário (interações contínuas entre usuário e sistema por meio de dispositivos paralelos como capacete e luvas), enquanto sistemas convencionais apresentam interfaces consideradas WIMP [Van Dam 1997] [Jacob et al 1999], que provêm estilo de interação serial e discreto com o usuário (interações discretas entre usuário e sistema por meio de dispositivos seriais como teclado ou mouse). Através de seu estilo de interação, interfaces Pós-WIMP (como interfaces de sistemas com RV) permitem maior grau de interatividade entre usuários e sistemas se comparadas com interfaces WIMP, oferecendo a promessa de serem mais fáceis para aprender e usar, uma vez que o projeto de interação é definido de acordo com as habilidades dos usuários com o mundo real [Jacob et al 1999].

Ainda considerando os recursos que interfaces com RV devem apresentar para prover a interação proposta, diferentes características de diálogos de entrada e saída, de requisitos e de monitoramento de comandos, como interpretadas por este trabalho com base no trabalho de Jacob et al (1999) e apresentadas na tabela 2.2, são exibidas por essas interfaces em contraste com interfaces de convencionais.

Tabela 2.2 - Tabela Comparativa das Características de Diálogos de Entrada e Saída, de Requisitos e de Monitoramento de Comandos de Interfaces Convencionais e de Interfaces com RV

Interfaces Convencionais (WIMP)	Interfaces com RV (Pós-WIMP ou Não-WIMP)
Diálogos de entrada/saída através de fluxo único <i>serial</i> (síncrono)	Diálogos de entrada/saída através de diversos fluxos <i>paralelos</i> (assíncronos porém relacionados)
Diálogos de entrada/saída <i>discretos</i>	Diálogos de entrada/saída <i>discretos e contínuos</i>
Entradas <i>precisas</i> (fácil interpretação)	Entradas <i>probabilísticas</i> (podem apresentar difícil interpretação)
Requisitos de <i>seqüência</i> (e não tempo) são importantes	Requisitos de <i>tempo</i> (computações baseadas em tempo real) são importantes
Monitoramento de comandos <i>explícitos</i> do usuário	Monitoramento de comandos <i>passivos</i> do usuário

Conforme a tabela 2.2, algumas características fundamentais diferem interfaces de sistemas convencionais, interfaces WIMP, de interfaces de sistemas com RV, interfaces Pós-WIMP, tornando modelos, linguagens e ferramentas de software de interfaces direcionados à construção de interfaces convencionais inadequados à construção de interfaces com RV. Segundo Jacob et al (1999), avanços na tecnologia de interfaces ultrapassaram avanços em modelos, linguagens e ferramentas de software de interfaces, que ainda baseados, em sua maioria, no estilo de interação serial e discreto, são bastante apropriados para a construção de interfaces WIMP, as quais se caracterizam por suas interações discretas e seriais com o usuário, e pouco adequados para a construção de interfaces Pós-WIMP, as quais se caracterizam por suas interações contínuas e paralelas com o usuário. Dessa forma, apesar de serem mais inerentes aos usuários, interfaces Pós-WIMP são mais difíceis de ser construídas [Jacob et al 1999].

Por fim, além de considerar a natureza ainda experimental do projeto de interação multimodo, sistemas com RV devem apresentar desempenho em tempo real, no qual é preciso garantir níveis aceitáveis de imersão, interação e navegação [Seo et al 2002].

Dessa forma, percebe-se que a partir de suas características, sistemas com RV possuem maior grau de complexidade em seu processo, o qual deve estar pautado por metodologias mais específicas as quais, para alcançar a interação proposta pela tecnologia de RV, devem adotar enfoque no usuário, iteratividade de projeto e usabilidade.

2.4. Enfoque no Usuário no Contexto de RV

O processo de sistemas com RV, bem como os desenvolvedores desses sistemas devem estar voltados para a utilização da tecnologia de RV de tal modo que os sistemas desenvolvidos apoiem e atendam aos requisitos de seus usuários atendendo critérios de qualidade [Bevan 1999]. Contudo, artigos ou pesquisas que detalhem o desenvolvimento de sistemas com RV de tal forma que os potenciais benefícios do emprego dessa tecnologia em sistemas interativos sejam realmente explorados, dificilmente são encontrados [Scaife et al 2001] [Smith et al 2001]. Observa-se que, segundo este trabalho, analisar e formalizar amplamente o processo de sistemas com RV, em relação à sua abrangência nas áreas de ES e IHC, se apresenta como importante desafio na utilização dessa tecnologia.

Segundo Scaife et al (2001), as características de sistemas com RV, principalmente no que se refere à alta interação entre esses sistemas e seus usuários, têm levado à utilização de

metodologias para sistemas com RV baseadas em abordagens da IHC [Brown 1997] como o Projeto Centrado no Usuário [Preece 1993], referenciado neste trabalho por PCU (em inglês, UCD, *User-Centered Design*) e o Design Participativo [Kuhn et al 1993], referenciado neste trabalho por DP, (em inglês, PD, *Participatory Design*), as quais estão fundamentadas no enfoque no usuário durante o processo de projeto [Goransson et al 1999] [Gulliksen et al 1999]. Observa-se que o envolvimento de usuários deve, idealmente, ser integrado em todo o processo de projeto [Gulliksen et al 1999] [Neale et al 2001].

A abordagem do PCU pode implicar ou não na participação do usuário junto à equipe de desenvolvimento. Enquanto isso, a abordagem do DP necessariamente deve implicar na participação do usuário junto à equipe de desenvolvimento [Gulliksen et al 1999]. A importância da atividade de prototipação é reconhecida na abordagem PCU [Rocha et al 2000] e, portanto, também na abordagem DP, uma vez que essa abordagem é considerada como um modo específico do PCU [Gulliksen et al 1999]. A atividade de prototipação possibilita que avaliações iterativas de um sistema sejam realizadas com usuários, desempenhando um importante papel na identificação e modelagem de requisitos de sistemas interativos no que diz respeito à sua interface, usabilidade e funcionalidade [Pressman 2001]. A atividade de prototipação permite que importantes informações de projeto sejam obtidas desde o início do processo de projeto [Hall 2001]. A utilização do PCU e, segundo este trabalho, do DP, é realizada em conjunto com o modelo de processo de Prototipação [Pressman 2001] da ES, pois esse modelo de processo se adapta às exigências dessas abordagens [Murray et al 1997]. Dessa forma, considera-se que o PCU e DP estão baseados no modelo de processo de Prototipação da ES. Algumas abordagens restringem as abordagens PCU e DP apenas ao desenvolvimento de interfaces, não explicitando o desenvolvimento do componente funcional do sistema [Shneiderman 1998] [Nielsen 1993]. Contudo, outras abordagens evidenciam que a contribuição do PCU e do DP pode ser mais abrangente do que somente no desenvolvimento de interfaces, pautando o desenvolvimento de sistemas interativos como um modelo de processo completo da ES [Brown 1997]. No entanto, nota-se que para que o PCU e o DP se apliquem além do desenvolvimento de interfaces, o conceito de independência de diálogo [Lucena et al 1997] deve ser considerado através do componente de controle de diálogo, o qual permite a comunicação entre os componentes de interface e de aplicação em sistemas de informação. Esse conceito separa aspectos concernentes a interface dos relativos a aplicação, uma vez que apresentam funcionalidades distintas. A aplicação se preocupa com o processamento de dados em um sistema. A interface se preocupa em obter

esses dados do usuário, bem como exibir os resultados de seus processamentos [Lucena et al 1997].

Conforme afirma Shneiderman (1998), uma abordagem da IHC com enfoque nos usuários integrada com um modelo de processo da ES pode englobar todo o processo de sistemas interativos, nos quais se incluem sistemas com RV.

Desse modo, embora alguns autores considerem que as abordagens PCU e DP estejam voltadas somente para o desenvolvimento de interfaces, acredita-se que uma abordagem da IHC com enfoque nos usuários integrada com um modelo de processo da ES possa englobar todo o processo de sistemas interativos, nos quais se incluem sistemas com RV.

Vale lembrar que devido à alta integração necessária entre esses sistemas e seus usuários, acredita-se que abordagens com enfoque no usuário se tornam requisito essencial para que sistemas com RV atendam a padrões de usabilidade esperados [Smith et al 2001] e, desse modo, devem estar inseridas no processo desses sistemas.

2.5. Iteratividade de Projeto no Processo de Sistemas com RV

Segundo Pressman [2001], as fases genéricas da ES são encontradas em todos os processos de sistemas, independentemente do modelo de processo seguido e da área de aplicação, tamanho ou complexidade do sistema desenvolvido.

Conforme Lee et al (2002), o processo de sistemas com RV é composto pelas etapas de especificação, implementação e avaliação, exigindo iteratividade de projeto [Kim et al 1998] [Lee et al 2002] [Seo et al 2002] [Smith et al 2001] [Stuart et al 1996] [Tanriverdi et al 2001].

Vale notar que é possível correlacionar as fases genéricas da ES e as etapas do processo de sistemas com RV com base em suas atividades. A etapa de especificação corresponde à fase genérica de definição da ES. As etapas de implementação e avaliação correspondem à fase genérica de desenvolvimento da ES. Nota-se que, de acordo com o escopo deste trabalho, não há etapas correspondentes à fase genérica de manutenção da ES para contemplar o processo de sistemas com RV, pois nessa fase são reaplicados os passos das fases de definição e desenvolvimento.

Dessa forma, acredita-se que qualquer metodologia proposta para sistemas com RV deve apresentar, ao menos, etapas iterativas correspondentes as fases genéricas de definição e desenvolvimento da ES, bem como suas atividades, uma vez que essas fases são encontradas

em todos os processos de sistemas e iteratividade de projeto é exigida para o processo de sistemas com RV.

2.6. Usabilidade no Contexto de RV

Como na maioria dos sistemas de computação, um importante fator para o sucesso de sistemas com RV é a garantia quanto à satisfação de seus usuários em relação à usabilidade desses sistemas, ou seja, em relação às características que refletem o potencial de funcionalidade, proporcionando um adequado equilíbrio entre a facilidade de aprendizagem e a de uso dos sistemas [Pereira et al 2000] [Smith et al 2001].

Nesse contexto, acredita-se que desenvolver sistemas com RV, de tal modo que apresentem valor estético, funcional e que atendam aos anseios de seus usuários em relação à sua usabilidade, também se apresenta como importante desafio na utilização dessa tecnologia, uma vez que ainda precisam ser ponderadas inúmeras questões, dentre elas como melhor analisar e formalizar o processo desses sistemas [Scaife et al 2001] [Seo et al 2002] [Smith et al 2001] [Tanriverdi et al 2001] [Willians et al 2001], considerando não apenas enfoque no usuário e iteratividade de projeto, mas também usabilidade através de critérios de usabilidade para sistemas com RV.

Nota-se que não somente ferramentas e abordagens mais detalhadas para o processo de sistemas com RV dificilmente são encontradas ou amplamente aceitas [Scaife et al 2001] [Seo et al 2002] [Smith et al 2001] [Tanriverdi et al 2001] [Willians et al 2001], mas também abordagens mais detalhadas para o processo de sistemas com RV que visem critérios de usabilidade mais específicos distribuídos ao longo da concepção, projeto e implementação desses sistemas.

Segundo Stanney et al (2003), usabilidade em sistemas com RV está apenas começando a receber o foco de atenção necessário para identificação de uma taxonomia de critérios de usabilidade específicos esses sistemas.

Princípios de usabilidade tradicionais não consideram características específicas de sistema com RV. Além de não considerarem essas características, critérios de usabilidade para sistema com RV dificilmente são encontrados e empiricamente derivados ou validados [Gabbard et al 1997] [Stanney et al 2003].

Métodos de avaliação de usabilidade tradicionais têm sido aplicados em sistemas com RV. No entanto, esses métodos podem não representar características específicas de sistemas

com RV, apresentando limitações quando aplicados a esses sistemas [Gabbard et al 1997] [Stanney et al 2003].

Nesse contexto, trabalhos mais recentes [Bowman et al 2000] [Gabbard et al 1997] [Gabbard et al 1999] [Kalawsky 1999] [Stanney et al 2003] se desenvolvem tentando melhorar sistemas com RV a partir do ponto de vista dos usuários, buscando uma abordagem para avaliação de usabilidade em sistemas com RV.

É interessante notar que além da necessidade de desenvolvimento de abordagens específicas para avaliação de usabilidade em sistemas com RV, surge também a necessidade de estabelecer critérios de usabilidade específicos para esses sistemas.

No que diz respeito aos critérios de usabilidade para sistema com RV, Gabbard et al (1997) apresentam uma taxonomia das características de usabilidade em sistemas com RV, alguns métodos de avaliação de usabilidade para sistemas interativos, e suas aplicações e limitações para sistemas com RV.

No que se refere aos métodos para desenvolver e avaliar critérios de usabilidade em sistemas com RV, Stanney et al (2003) apresentam uma abordagem derivada dos trabalhos de Gabbard et al (1999), Kalawsky (1999), Kennedy et al (1993) e Witmer et al (1993), a qual identifica diretivas de usabilidade em sistema com RV e apresenta uma hierarquia de critérios de usabilidade para esses sistemas, a qual é adotada neste estudo.

Desse modo, sugere-se que os critérios de usabilidade para sistemas com RV sejam incorporados ao processo desses sistemas.

2.6.1. Critérios de Usabilidade para RV

De acordo com Gabbard et al (1999), dois domínios compõem considerações de desenvolvimento de software interativo: 1) domínio de construção, ou seja, a visão dos desenvolvedores de software e o projeto da aplicação e 2) domínio comportamental, ou seja, a visão dos usuários e suas interações com a aplicação.

Segundo Stanney et al (2003), usabilidade em sistema com RV está relacionada a duas entidades: 1) interface do sistema, ou seja, considerações de hardware, software e projeto de interação e 2) interface do usuário, ou seja, considerações fisiológicas, psicológicas e psicossociais.

A abordagem MAUVE (em inglês, *Multi-criteria Assessment of Usability for Virtual Environments*) [Stanney et al 2003] adota essas visões, identificando, categorizando e

integrando critérios associados com características da interface do sistema com RV e com as considerações do usuário dessa interface em uma abordagem sistemática para projetar e avaliar usabilidade em sistemas com RV, através do desenvolvimento de uma técnica de avaliação multicritérios, bem como de uma ferramenta de avaliação de sistemas com RV automatizada.

Conforme Stanney et al (2003), a MAUVE é composta por uma hierarquia de critérios de usabilidade para sistemas com RV:

1. Características do Sistema com RV

1.1. *Interação*

- 1.1.1. Navegação: usuários devem poder se localizar e se orientar prontamente.
- 1.1.2. Manipulação de Objetos: usuários devem poder selecionar e manipular objetos virtuais de forma fácil e natural.
- 1.1.3. Movimento do Usuário: usuários devem poder controlar seus movimentos de modo intuitivo e contínuo.

1.2. *Entrada e Saída Multimodais de Sistema*

- 1.2.1. Visual: deve-se otimizar campo de visão, enquanto deve-se minimizar retardação visual e distorções.
- 1.2.2. Auditivo: deve-se empregar eficazmente informações e retornos auditivos de modo oportuno e significativo.
- 1.2.3. Tátil: deve-se incorporar eficazmente apresentações táteis e de força para apoiar tarefas do usuário de forma confiável.

2. Considerações do Usuário do Sistema com RV

2.1. *Envolvimento*

- 2.1.1. Presença: presença, experiência subjetiva de que se está em um ambiente quando se está fisicamente em outro, deve ser promovida.
- 2.1.2. Imersão: imersão, percepção de que se está envolvido por, incluído em e interagindo com um ambiente que provê fluxo contínuo de estímulos e experiências, deve ser promovida.

2.2. *Efeitos Colaterais*

- 2.2.1. Conforto: deve-se minimizar desconfortos físicos, enquanto deve-se maximizar a segurança do usuário.
- 2.2.2. Males: deve-se minimizar náusea, desorientação e distúrbios visuais.
- 2.2.3. Efeitos Posteriores: deve-se minimizar efeitos deletérios posteriores.

Para avaliar os critérios de usabilidade para sistemas com RV, diversas considerações de projeto são representadas por perguntas específicas para cada um desses critérios na abordagem MAUVE, a qual é adotada neste trabalho, podendo aumentar a capacidade de desenvolvedores e avaliadores na maximização e avaliação sistemática de usabilidade nesses sistemas através da consideração de importantes aspectos de usabilidade não consideradas tradicionalmente [Stanney et al 2003].

Desse modo, inserir os critérios de usabilidade para sistemas com RV em seu processo se apresenta como importante desafio para a obtenção de sistemas com RV que atendam amplamente as necessidades de seus usuários.

2.7. Considerações Finais

Neste capítulo, foram apresentadas a contextualização da tecnologia de RV, as principais características de sistemas com RV e as implicações encontradas no processo desses sistemas.

As três principais implicações identificadas por este trabalho no processo de sistemas com RV, enfoque no usuário, iteratividade de projeto e usabilidade, são sugeridas como exigências que devem ser adotadas na concepção, projeto e implementação desses sistemas, para que, dessa forma, sistemas com RV possam atender aos anseios de seus usuários, no que concerne às características que refletem seu potencial de funcionalidade, facilidade de aprendizagem e de uso.

As principais características de sistemas com RV e as implicações encontradas no processo desses sistemas contribuíram significativamente para a análise crítica e comparativa sobre as metodologias para sistemas com RV citadas na literatura realizada neste trabalho no capítulo seguinte.

3. Metodologias para Sistemas com RV

3.1. Considerações Iniciais

A princípio, conforme afirmam Smith et al (1999a), devido aos avanços de hardware e software e devido à disponibilização de novos dispositivos de entrada e saída, muitas das limitações e restrições associadas ao desenvolvimento de sistemas com RV estão sendo removidas. Porém, da mesma forma que a tecnologia de RV atinge maior maturidade, sendo adotada em uma ampla gama de aplicações, existe a necessidade de se entender como essa tecnologia pode ser acomodada na prática da ES [Smith et al 1999a]. Artigos ou pesquisas que detalhem o processo de sistemas com RV dificilmente são encontrados [Smith et al 2001]. Em contrapartida, conforme afirmam Tanriverdi et al (2001), apesar de suas vantagens e potencialidades, a utilização de sistemas com RV ainda não se encontra amplamente difundida. Essa falta de proliferação de sistemas com RV pode ser parcialmente atribuída aos desafios encontrados para desenvolver essas aplicações, principalmente no que se refere à necessidade de metodologias específicas para o processo de sistemas com RV [Tanriverdi et al 2001]. A partir dessas duas visões, nota-se que, apesar de divergirem sobre a abrangência da adoção e utilização da tecnologia de RV nas aplicações, Smith e Tanriverdi sugerem e alertam sobre a necessidade de pesquisas direcionadas ao processo de sistemas com RV.

Vale ressaltar que essa necessidade é ratificada por diversos outros trabalhos. Além de Smith e Tanriverdi, Seo et al (2002) afirmam que pouca atenção tem sido dada a métodos e ferramentas para o desenvolvimento estruturado de sistemas com RV. Da mesma forma, Scaife et al (2001) afirmam que poucos direcionamentos são encontrados sobre quais métodos utilizar para desenvolver sistemas com RV de tal forma que seus benefícios sejam verdadeiramente explorados. E ainda, Willians et al (2001) afirmam que não existem metodologias de desenvolvimento de sistemas com RV amplamente aceitas. A partir dessas demais visões, conclui-se que ferramentas direcionadas ao desenvolvimento de sistemas com RV, bem como abordagens que detalhem o processo desses sistemas são dificilmente encontradas ou amplamente aceitas.

Desse modo, este capítulo é destinado ao levantamento, à apresentação e análise crítica das metodologias para sistemas com RV citadas na literatura.

No item 3.2 é retratada a metodologia VRID. O item 3.3 discute a metodologia CLEVR. No item 3.4 é discutida a metodologia de Stuart. O item 3.5 retrata a metodologia de Scaife e

Rogers. No item 3.5 são analisadas criticamente as metodologias para sistemas com RV VRID, CLEVR, de Stuart e de Scaife e Rogers. O item 3.6 apresenta algumas considerações finais quanto às metodologias para sistemas com RV citadas na literatura.

3.2. Modelo e Metodologia VRID

O modelo e metodologia VRID (em inglês, *Virtual Reality Interface Design*) [Tanriverdi et al 2001] contém direcionamentos para o projeto de interfaces de sistemas com RV. Essa abordagem objetiva apoiar projetistas de interfaces de sistemas com RV a pensar de modo compreensível o projeto de interface, decompor as tarefas de projeto em tarefas menores, mais simples e conceitualmente distintas e comunicar a estrutura de projeto entre projetistas e desenvolvedores de sistemas com RV.

Para segmentar e diferenciar conceitualmente a maior complexidade presente em objetos de interfaces de sistemas com RV, o modelo VRID, mostrado na figura 3.1, é organizado em torno de uma arquitetura de objetos multicomponentes.

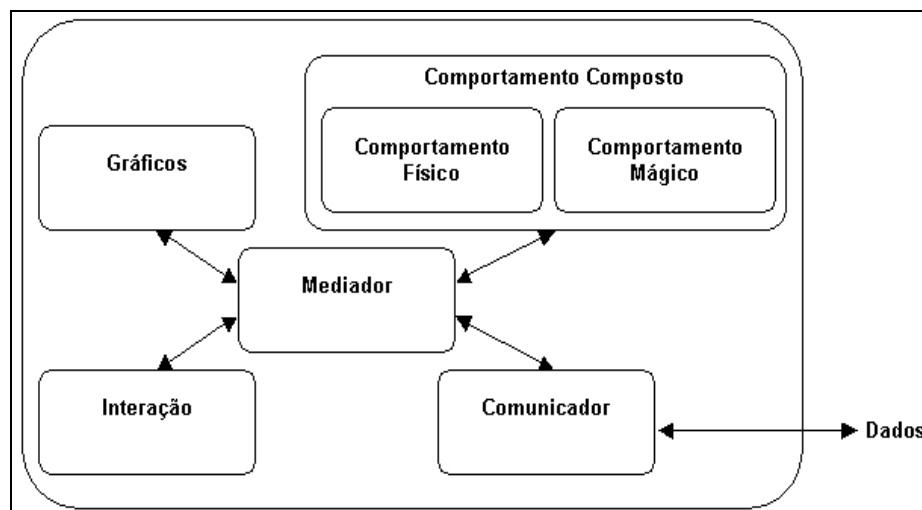


Figura 3.1 - Modelo VRID (Extraída de [Tanriverdi et al 2001])

Pela figura 3.1, a arquitetura de objetos multicomponente utilizada no modelo VRID é composta pelos seguintes componentes:

- **Componente gráfico:** utilizado para especificar os modelos gráficos necessários para a geração por computador da aparência e das animações dos objetos de interface. Como objetos de interfaces de sistemas com RV apresentam

comportamentos mais complexos que precisam ser representados por formas visuais também mais complexas, o objetivo é prover um direcionamento para que os projetistas de interfaces de sistemas com RV possam especificar os modelos gráficos desses objetos em um alto nível de abstração.

- **Componente de comportamento:** componente utilizado para especificar os comportamentos dos objetos de interface. Divisões de comportamento, em físicos ou mágicos, conforme sua possível existência ou não no mundo real, e simples ou compostos, conforme sua complexidade, permitem aos projetistas entender, decompor e projetar de forma simplificada os comportamentos dos objetos. Essa simplificação possibilita que os componentes de comportamento projetados sejam mais facilmente reutilizados na criação de novos comportamentos.
- **Componente de interação:** utilizado para especificar as entradas do sistema com RV e como essas entradas mudam os comportamentos dos objetos de interface. O componente de interação recebe uma entrada, interpreta seu significado, analisa sua implicação para o comportamento de um objeto e a comunica ao componente de comportamento, o qual realiza a mudança de comportamento desejada.
- **Componente comunicador:** componente utilizado para especificar as comunicações de um objeto de interface com outros objetos, elementos de dados ou com o componente de aplicação. Nesse componente, os projetistas de interfaces de sistemas com RV especificam as fontes e os destinos de comunicação de um objeto de interface, bem como os mecanismos de troca de mensagem utilizados. As comunicações internas e externas são analisadas em alto nível de abstração e especificadas separadamente. O objetivo é auxiliar os projetistas na decomposição da complexidade associada com o projeto de comunicação, facilitando os processos de análise, projeto, codificação e manutenibilidade dos componentes.
- **Componente mediador:** utilizado para especificar os mecanismos de controle e de coordenação para comunicações entre os demais componentes de um objeto de interface. Esse componente permite aos projetistas identificar e gerenciar quais pedidos de comunicação podem gerar conflitos entre os comportamentos de um objeto de interface. A partir do momento que os componentes de um objeto precisam ter conhecimento apenas sobre si mesmos e sobre o componente mediador, o componente mediador também assegura a perda de acoplamento entre os demais componentes.

Para aplicar sistematicamente o modelo VRID no projeto de interfaces de sistemas com RV, a metodologia VRID, ilustrada na figura 3.2, é especificada e utilizada.

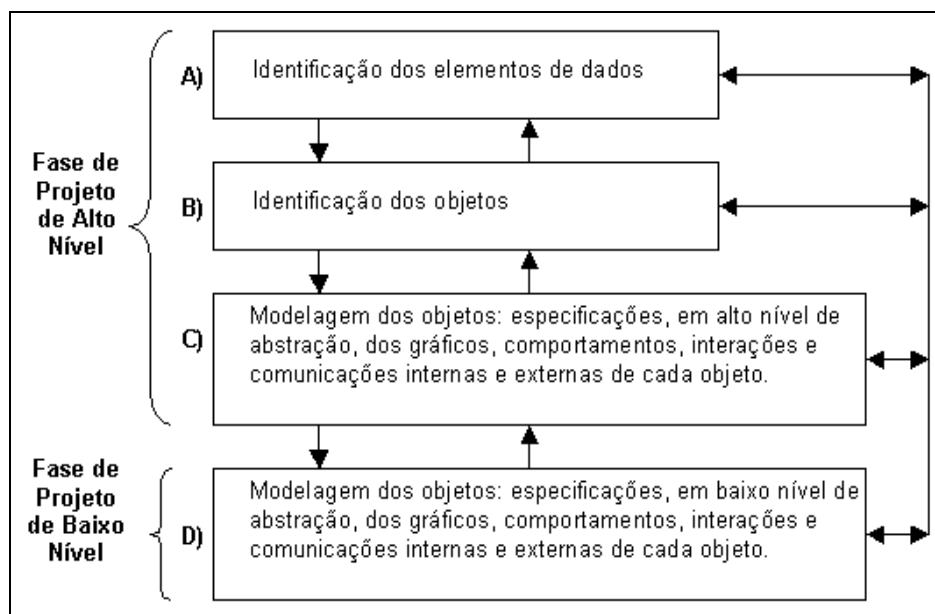


Figura 3.2 - Metodologia VRID (Adaptada de [Tanriverdi et al 2001])

Conforme a figura 3.2, a metodologia VRID é composta pela fase de projeto de alto nível de abstração e pela fase de projeto de baixo nível de abstração.

Na fase de projeto de alto nível de abstração, o objetivo é especificar uma solução de projeto utilizando o modelo VRID como direcionamento conceitual. A entrada para essa fase é uma descrição funcional da interface do sistema com RV, utilizada por três passos para identificar e especificar, em alto nível de abstração, os elementos de dados e objetos de interface:

- A) Identificação dos elementos de dados:** o papel dos elementos de dados, como usuários, dispositivos físicos ou outros sistemas, é permitir a comunicação entre a interface e as entidades externas ao sistema com RV. Os elementos de dados devem ser identificados a partir da análise da descrição funcional da interface do sistema com RV.
- B) Identificação dos objetos:** objetos têm papéis e identidades bem definidas na interface do sistema com RV. A realização desse passo envolve identificação dos possíveis objetos mencionados na descrição funcional da interface do sistema com RV, decisão sobre a legitimidade desses objetos e distinção entre objetos físicos e virtuais.

C) Modelagem dos objetos: objetos identificados como físicos não são abordados nesse passo, uma vez que não precisam ser modelados e suas entradas para o sistema com RV já foram identificadas como elementos de dados. Desse modo, somente objetos virtuais devem ser abordados através da especificação de seus gráficos, comportamentos, interações e características de comunicação interna e externa. O modelo VRID deve ser utilizado para modelar os aspectos de cada objeto virtual a partir de uma nova análise da descrição da interface do sistema com RV.

A saída dessa fase de projeto é uma representação em alto nível de abstração dos elementos de dados e dos objetos de interface identificados.

Na fase de projeto de baixo nível de abstração, o objetivo é prover detalhes refinados das representações de alto nível. Essa fase tem como entrada a representação obtida na fase de projeto anterior, utilizada por um único passo para modelar novamente os objetos de interface, porém em baixo nível de abstração:

D) Modelagem dos objetos: as especificações dos gráficos, comportamentos, interações e características de comunicação interna e externa dos objetos obtidas no passo anterior são refinadas de modo detalhado.

Essa fase de projeto tem como saída um conjunto de especificações de projeto em baixo nível de abstração, representadas em terminologias orientadas à implementação.

Nessa metodologia, uma abordagem *top-down* é adotada para o processo de projeto, partindo de abstrações de alto nível para detalhes de baixo nível de abstração. Contudo, esse não é um processo linear. Inúmeras iterações entre as fases de projeto de alto e baixo nível de abstração, e refinamentos recíprocos em ambos os níveis, são realizados até que se tenha um projeto de interface do sistema com RV conceitualmente seguro e realmente implementável.

Um estudo de caso com o modelo e metodologia VRID aplicado na especificação de interface de um sistema com RV pode ser encontrado em [Tanriverdi et al 2001].

3.3. Metodologia CLEVR

A metodologia CLEVR (em inglês, *Concurrent and Level by Level Development of VR Systems*) [Seo et al 2002] segue como princípios para o processo de sistemas com RV a consideração simultânea da forma, função e comportamento dos objetos de interface, a realização de um processo incremental e hierárquico, e a utilização de um modelo de processo

de desenvolvimento espiral. Segundo Seo et al (2002), o processo de sistemas com RV deve considerar, tão cedo quanto possível, algumas características como desempenho e usabilidade.

A metodologia CLEVR, baseada no modelo de processo Espiral da ES [Pressman 2001] de acordo com seus autores, é formada por três etapas, onde iterações de análise de requisitos, projeto e validação são realizadas continuamente a cada etapa, representadas na figura 3.3.

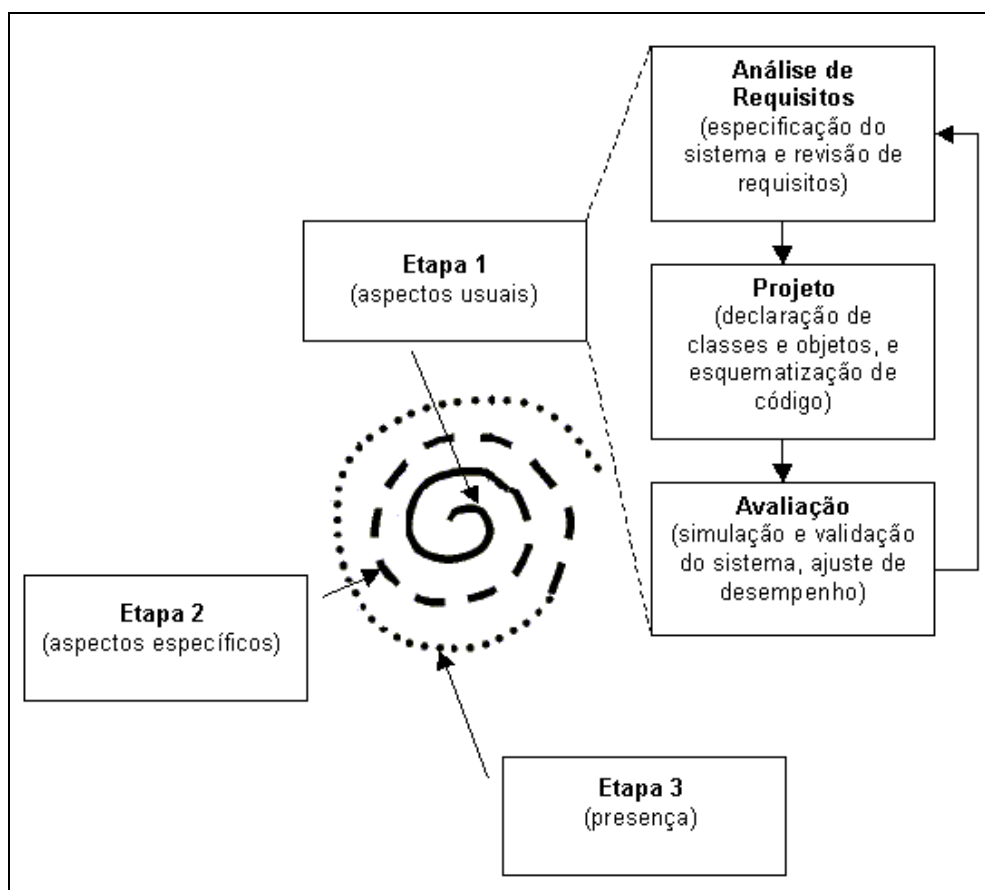


Figura 3.3 - Metodologia CLEVR (Adaptada de [Seo et al 2002])

Pela figura 3.3, as etapas que compõem a metodologia CLEVR são descritas a seguir:

- 1. Etapa 1:** nessa etapa as iterações são focadas em aspectos usuais, como análise de requisitos, identificação dos objetos de interface e cenários, hierarquia de classes, modelagem das tarefas do usuário e comportamento e arquitetura geral do sistema com RV.
- 2. Etapa 2:** a segunda etapa de desenvolvimento envolve aspectos mais relacionados à tecnologia de RV, abordando assuntos como modelagem das formas, funções e comportamentos dos objetos de interface identificados, refinamento dos modelos

especificados, e simulações e execuções incrementais para validação e revisão dos comportamentos dos objetos, estimativa de desempenho e apoio às decisões de projeto.

- 3. Etapa 3:** nessa etapa efeitos especiais e elementos como aumento das formas de entradas e saídas, variação de tipos de interação e minimização da distração, podem ser utilizados para melhorar a característica de presença do sistema com RV.

Nessa metodologia, as especificações e modelagens são representadas em terminologias orientadas à implementação. As segunda e terceira etapas de iteração distinguem o processo de sistemas com RV do processo de sistemas interativos comuns, ou seja, a maior parte das características que diferenciam sistemas com RV de outros sistemas é abordada durante essas duas últimas etapas. A abordagem incremental elimina a necessidade de se ter um conjunto completo de especificações de requisitos para um sistema antes que sua implementação possa ser iniciada. Enquanto isso, a abordagem hierárquica promove maior conhecimento quanto ao desempenho do projeto, induzindo o desenvolvedor a considerar, de modo *top-down*, as características mais críticas referentes à forma, função e ao comportamento dos objetos.

Um conjunto de ferramentas para apoio aos desenvolvedores no processo de sistemas com RV, chamado P-VoT (em inglês, *POSTECH - Virtual reality system development Tools*), bem como um estudo de caso são apresentados em [Seo et al 2002].

3.4. Metodologia de Stuart

A metodologia de Stuart [Stuart 1996] segue princípios utilizados na abordagem geralmente conhecida como Projeto Iterativo, os quais são recomendados por Gould et al (1985) e surgem da premissa que um sistema projetado para pessoas deve ser útil e de fácil uso e aprendizagem. No caso de sistemas com RV, a garantia de segurança no uso também deve ser adicionada a essa premissa.

Para alcançar os princípios recomendados, o Projeto Iterativo, o qual se mostrou aplicável ao processo de diversos tipos de interfaces, propõe as seguintes considerações:

- Os projetistas entendam quem são os usuários, em termos de suas atitudes, comportamentos e características cognitivas e em termos do trabalho que eles realizarão utilizando o sistema;

- A equipe de projetistas trabalhe próxima aos usuários representativos desde o início do processo de projeto;
- Os protótipos sejam criados o mais breve possível no processo, sendo utilizados para avaliar o projeto de acordo com o seu desempenho e as reações dos usuários;
- Os resultados da avaliação através dos protótipos sejam utilizados para projetar, prototipar e avaliar novamente o sistema.

A metodologia de Stuart, ilustrada na figura 3.4, é composta pelos passos de definição dos requisitos, projeto, protótipo e avaliação do sistema, os quais são realizados de modo iterativo. A cada iteração os requisitos são revisados e novamente projetados, prototipados e avaliados. Esse ciclo deve se repetir até que a avaliação seja satisfatória, ou seja, de acordo com todos os requisitos e critérios especificados, no que diz respeito ao desempenho do sistema com RV e ao comportamento do usuário frente à aplicação.

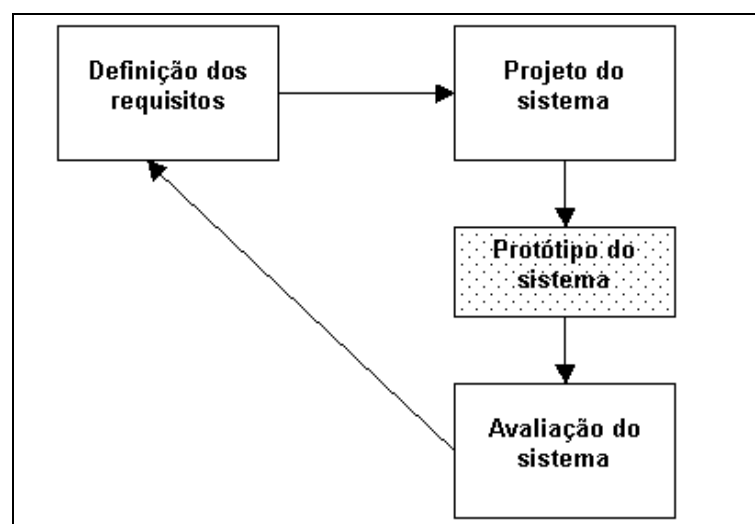


Figura 3.4 - Metodologia de Stuart (Adaptada de [Stuart 1996])

Conforme a figura 3.4, os passos que compõem a metodologia de Stuart são descritos a seguir:

- 1. Definição dos requisitos:** esse passo envolve a definição clara dos requisitos do sistema com RV, os quais incluem além de requisitos funcionais, capacidades humanas como perceptual, motora, cognitiva e afetiva, desempenho do sistema e natureza dos usuários, das tarefas e do ambiente no qual o sistema com RV será inserido.
- 2. Projeto do sistema:** nesse passo o sistema com RV deve ser projetado com base nos requisitos definidos. Para isso as tecnologias de entrada, saída e de

computação, a arquitetura do sistema, os objetos e seus comportamentos e interações no sistema com RV devem ser escolhidos e projetados de acordo com os requisitos definidos.

3. **Protótipo do sistema:** esse passo envolve a construção do protótipo do sistema com RV conforme o projeto especificado para o sistema.
4. **Avaliação do sistema:** nesse passo o desempenho e a usabilidade do sistema com RV devem ser avaliados através do protótipo construído. Um desempenho muito ruim do protótipo pode alertar tanto para a possibilidade de que o sistema projetado realmente venha a ter um desempenho similarmente inaceitável, quanto para a possibilidade de que o projeto não tenha sido bem prototipado, ou seja, a avaliação pode mostrar problemas na definição de requisitos, no projeto do sistema e ou na construção do protótipo. Para que os problemas identificados possam ser efetivamente corrigidos, é importante que suas causas também sejam identificadas e corrigidas. Avaliar a usabilidade de sistemas com RV através de usuários representativos realizando as possíveis tarefas com protótipos dentro do contexto no qual eles realmente utilizarão as aplicações, permite verificar o valor real dos sistemas com RV na execução das tarefas. O resultado encontrado na avaliação do sistema determina se novas iterações no processo de projeto serão ou não necessárias.

O processo de um sistema com RV bem sucedido se apresenta como uma tarefa difícil, uma vez que problemas em qualquer momento de projeto podem prejudicar a usabilidade e o valor da aplicação. Para amenizar essa questão, o processo de sistemas com RV pode exigir uma abordagem, ao mesmo tempo, *top-down* e *bottom-up*. De acordo com Stuart (1996), o processo de Projeto Iterativo possibilita que essa abordagem seja alcançada, na qual além da abordagem *top-down*, o ciclo garante muitas das vantagens da abordagem *bottom-up*.

Uma descrição completa sobre a metodologia de Stuart é apresentada em [Stuart 1996].

3.5. Metodologia de Scaife e Rogers

A metodologia de Scaife e Rogers [Scaife et al 2001] contém atividades esperadas para o processo de sistemas com RV [Smith et al 2001]. Essa abordagem objetiva explorar de que modo diferentes atividades de pesquisa como construções teóricas e análises de

aplicações, estudos exploratórios e experimentais, prototipações e testes com usuários, podem ser utilizadas como ferramentas para apoiar o desenvolvimento de sistemas com RV. Segundo Scaife et al (2001), a filosofia para o desenvolvimento dessa metodologia surgiu de uma apreciação das formas como os projetistas trabalham e dos problemas por eles encontrados, tendo sido derivada de inúmeras interações com os mesmos em pesquisas e projetos de desenvolvimento.

A metodologia de Scaife e Rogers, baseada na abordagem de PCU [Preece 1993] da IHC segundo seus autores, que por sua vez está baseada no modelo de processo de Prototipação [Pressman 2001] da ES, é formada por cinco etapas de desenvolvimento, realizadas de forma progressiva, representadas na figura 3.5. Por sua vez, cada etapa é formada por um certo número de atividades, as quais também são realizadas de forma progressiva ou em paralelo. Os vários resultados obtidos em uma determinada etapa de desenvolvimento são utilizados em conjunto para informar a etapa seguinte.

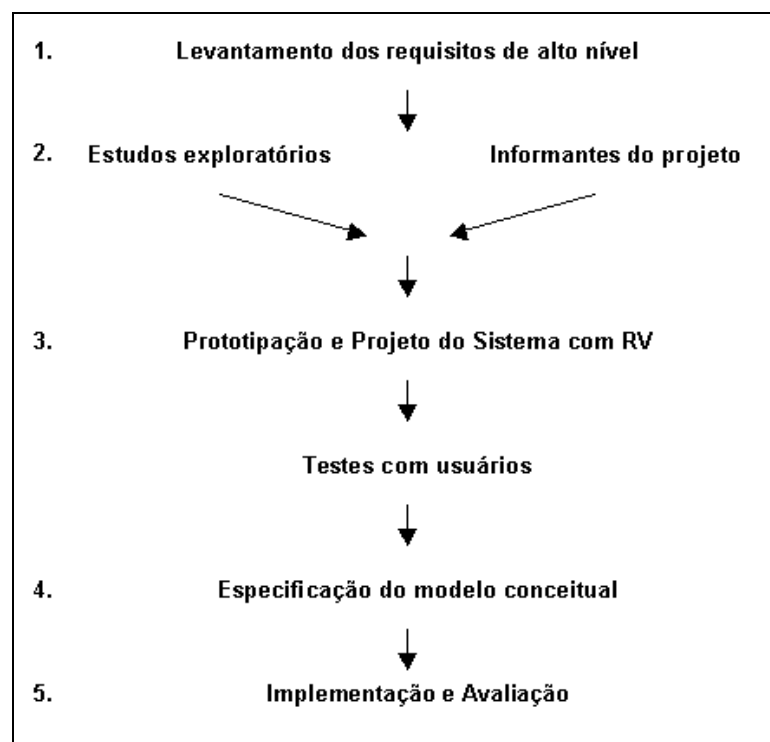


Figura 3.5 - Metodologia de Scaife e Rogers (Adaptada de [Scaife et al 2001])

Pela figura 3.5, as etapas que compõem a metodologia de Scaife e Rogers são descritas a seguir:

- 1. Levantamento dos requisitos de alto nível:** essa etapa envolve a definição de qual ambiente está sendo desenvolvido e porque. Essa definição permite que uma

nítida visão do espaço do problema seja obtida antes que qualquer solução de projeto seja proposta. O desenvolvimento de sistemas com RV é um processo onde um alto consumo de tempo, trabalho e custo são necessários para que mudanças de projeto sejam feitas caso a codificação tenha sido iniciada. Uma clara definição do espaço do problema antes de se iniciar o processo de prototipação pode evitar mudanças dispendiosas no projeto.

2. **Estudos exploratórios e Informantes do projeto:** nessa etapa são verificadas as práticas atuais e as possibilidades futuras no espaço do problema delimitado na primeira fase. Essa etapa envolve a realização de estudos exploratórios e a utilização de informantes do projeto. O resultado dessa etapa é identificar restrições sobre as diversas idéias de projeto, através da identificação das necessidades, preferências e práticas existentes.
3. **Prototipação, Testes com usuários e Projeto do sistema com RV:** essa etapa envolve o início do projeto do sistema com RV conforme os resultados obtidos nas etapas anteriores. A construção e utilização de protótipos permitem que testes preliminares com usuários possam ser realizados, avaliando e descartando idéias de projeto e solucionando possíveis problemas encontrados. O resultado dessa fase é um conjunto de implicações específicas do projeto.
4. **Especificação do modelo conceitual:** nessa etapa é escrito um modelo conceitual do projeto com base nas implicações encontradas na etapa anterior. Esse modelo é essencialmente um conjunto de especificações para funcionalidade, aparência e finalidade do sistema com RV. O modelo deve ser detalhado de tal forma que possa ser utilizado como importante material de apoio para os desenvolvedores.
5. **Implementação e Avaliação:** essa etapa envolve a implementação do sistema com RV na ferramenta de desenvolvimento escolhida e a avaliação desse sistema nas várias etapas como é iterado, no que diz respeito aos seus objetivos. Os critérios para a avaliação do sistema com RV podem ser obtidos através da identificação de um conjunto de objetivos específicos na fase inicial de projeto.

Um estudo de caso com a metodologia de Scaife e Rogers aplicada no desenvolvimento de um sistema com RV pode ser encontrado em [Scaife et al 2001].

3.6. Análise Crítica e Comparativa das Metodologias

O modelo e metodologia VRID se apresenta como uma abordagem apenas para o projeto de interfaces de sistemas com RV e de forma restrita ao contexto da IHC, não considerando o enfoque no usuário ou em usabilidade em seu processo. A metodologia VRID associada ao modelo VRID, através de iterações e refinamentos entre suas fases de projeto, permite obter projetos de interface de sistema com RV conceitualmente seguros e implementáveis na prática. No entanto, a metodologia VRID não apresenta algumas atividades esperadas para abranger todo o processo de sistemas com RV (como análise de requisitos, implementação e teste), bem como outros componentes que consistem esses sistemas (como usuário e aplicação), os quais tornariam essa metodologia mais completa.

A metodologia CLEVR se mostra como uma abordagem para todo o processo de sistemas com RV se comparada com a metodologia VRID, porém também de forma restrita ao contexto da IHC, ou seja, não incorporando o enfoque no usuário ou em usabilidade em seu processo proposto. O processo proposto nessa metodologia, conforme seus autores, está baseado no modelo de processo Espiral da ES. No entanto, percebe-se que não é possível identificar claramente como essa metodologia está fundamentada sobre esse modelo de processo, uma vez que não são apresentadas algumas das atividades características desse modelo de processo (como análise de riscos) durante todo o processo proposto. As atividades a serem realizadas, bem como o momento de sua realização, são bem definidos na metodologia CLEVR, possibilitando que a complexidade associada ao processo de sistemas com RV seja melhor gerenciada. Porém, devido ao desenvolvimento incremental e ao desenvolvimento hierárquico adotados nessa metodologia, segundo seus autores, ou baseado no modelo de processo Evolucionário da ES, conforme este estudo, suas etapas e iterações são realizadas de tal modo que não é possível determinar precisamente quais atividades pertencem a que etapa e ou iteração. Acredita-se que para que a metodologia CLEVR se torne mais completa, no âmbito das exigências no processo de sistemas com RV, é fundamental direcionar seu processo para o enfoque no usuário e em usabilidade.

A metodologia de Stuart se apresenta como uma abordagem apenas para a análise e para o projeto de sistemas com RV e de modo mais amplo ao contexto da IHC, ou seja, incorporando o enfoque no usuário e em usabilidade. O processo proposto nessa metodologia, segundo esta pesquisa, está baseado na abordagem PCU da IHC, que por sua vez está baseado no modelo de processo Prototipação da ES, considerando, desse modo, o enfoque no usuário. A preocupação com usabilidade é considerada na metodologia de Stuart durante as avaliações

de protótipos realizadas com usuários. Os passos que consistem essa metodologia são abrangentes, descrevendo características de sistemas com RV importantes somente em um nível mais baixo de abstração no processo de sistemas com RV. No entanto, a metodologia de Stuart não apresenta claramente alguns passos (como implementação, avaliação e correção de possíveis erros) esperados para contemplar todo o processo de sistemas com RV, os quais tornariam essa metodologia mais completa.

A metodologia de Scaife e Rogers se mostra como uma abordagem para todo o processo de sistemas com RV se comparada com a metodologia de Stuart, porém também de modo mais amplo ao contexto da IHC, ou seja, incorporando o enfoque em usabilidade e no usuário. O processo proposto nessa metodologia, conforme seus autores, está baseado na abordagem de PCU da IHC, que por sua vez está baseado no modelo de processo Prototipação da ES, considerando, desse modo, o enfoque no usuário. A preocupação com usabilidade é ponderada na metodologia de Scaife e Rogers durante as avaliações de protótipos realizadas com usuários. No entanto, ciclos que permitem iteratividade de projeto em seu processo não são encontrados claramente nessa metodologia. Acredita-se que para que a metodologia de Scaife e Rogers se torne mais completa, no âmbito das exigências no processo de sistemas com RV, é fundamental inserir iteratividade em seu processo.

Como resultado da análise crítica e comparativa entre as metodologias para sistemas com RV citadas na literatura, é apresentada a tabela 3.1, que mostra sobre quais modelos de processo da ES e sobre quais abordagens da IHC, as metodologias para sistemas com RV estão fundamentadas segundo seus proponentes e segundo este trabalho.

Tabela 3.1 - Tabela Comparativa das Fundamentações no Processo das Metodologias para Sistemas com RV

Metodologia para Sistemas com RV	Modelo de Processo da ES		Abordagem da IHC	
	Segundo seus proponentes	Segundo este trabalho	Segundo seus proponentes	Segundo este trabalho
VRID	Nenhum	Nenhum	Nenhuma	Nenhuma
CLEVR	Espiral	Evolucionário	Nenhuma	Nenhuma
Stuart	Nenhum	Prototipação	Projeto Iterativo	PCU
Scaife e Rogers	Nenhum	Prototipação	PCU	PCU

Pela tabela 3.1, de acordo com seus autores e com este estudo, a metodologia VRID não está fundamentada sobre nenhum modelo de processo da ES e sobre nenhuma abordagem da IHC. A metodologia CLEVR está pautada, segundo seus autores, pelo modelo de processo Espiral da ES. No entanto, segundo este trabalho, essa metodologia está pautada pelo modelo

de processo Evolucionário da ES, uma vez que apresenta em seu processo as características de desenvolvimento sugeridas por esse modelo. A metodologia CLEVR não está pautada por nenhuma abordagem da IHC conforme seus autores e conforme este trabalho. A metodologia de Stuart não está fundamentada sobre nenhum modelo de processo da ES segundo seu autor. Contudo, essa metodologia está fundamentada sobre o modelo de processo Prototipação da ES segundo este trabalho, pois apresenta em seu processo as características de desenvolvimento sugeridas por esse modelo. A metodologia de Stuart está fundamentada, conforme seu autor, sobre a abordagem de Projeto Iterativo da IHC. No entanto, conforme este trabalho, essa metodologia está fundamentada sobre a abordagem do PCU da IHC, uma vez que apresenta em seu processo as características de desenvolvimento sugeridas por essa abordagem. A metodologia de Scaife e Rogers não está pautada por nenhum modelo de processo da ES segundo seus autores. No entanto, essa metodologia está pautada pelo modelo de processo Prototipação da ES segundo este trabalho, pois apresenta em seu processo as características de desenvolvimento sugeridas por esse modelo. A metodologia de Scaife e Rogers está pautada, de acordo com seus autores e com este trabalho, pela abordagem do PCU da IHC.

Ainda como resultado da análise crítica e comparativa entre as metodologias para sistemas com RV citadas na literatura, é mostrada a tabela 3.2, que mostra sobre quais exigências do processo de sistemas com RV as metodologias para sistemas com RV estão fundamentadas, segundo este trabalho.

Tabela 3.2 - Tabela Comparativa das Implicações no Processo das Metodologias para Sistemas com RV

Metodologia para Sistemas com RV	Enfoque no Usuário	Iteratividade de Projeto	Crítérios de Usabilidade	Abrangência
VRID	Não	Sim	Não	Projeto de interfaces com RV
CLEVR	Não	Sim	Não	Todo o processo de sistemas com RV
Stuart	Sim	Sim	Não	Análise e projeto de sistemas com RV
Scaife e Rogers	Sim	Não	Não	Todo o processo de sistemas com RV

Conforme a tabela 3.2, dentre as metodologias analisadas, apenas as metodologias de Scaife e Rogers e de Stuart consideram o enfoque no usuário e em usabilidade para o processo de sistemas com RV, realizando prototipações e avaliações com usuários. Porém, dentre essas

metodologias analisadas, a metodologia de Stuart não explicita claramente atividades de implementação, avaliação e correção de possíveis erros que complementem o projeto gerado em seu processo, as quais são esperadas para completar o processo de sistemas com RV, ao passo que a metodologia de Scaife e Rogers não explicita claramente ciclos que gerem iteratividade em seu processo, a qual é esperada para o processo de sistemas com RV.

3.7. Considerações Finais

Este capítulo apresentou as metodologias para sistemas com RV encontradas na literatura e a análise crítica e comparativa dessas metodologias.

Conforme a análise realizada, nota-se que dentre as metodologias analisadas, apenas duas, a de Stuart e a de Scaife e Rogers, consideram o enfoque no usuário para o processo de sistemas com RV, realizando prototipações e avaliações com usuários. Porém, dentre essas duas metodologias, a de Scaife e Rogers não explicita claramente ciclos que permitam iteratividade de projeto em seu processo, a qual é esperada para o processo de sistemas com RV, enquanto a metodologia de Stuart não explicita claramente atividades de implementação, avaliação e correção de possíveis erros que complementem o projeto gerado em seu processo, as quais são esperadas para completar o processo de sistemas com RV. Ainda conforme a análise crítica realizada, percebe-se que todas as quatro metodologias analisadas não incorporam critérios de usabilidade para sistemas com RV em seu processo proposto.

No capítulo a seguir, a análise crítica sobre as metodologias para sistemas com RV é discutida com seus autores e são propostas novas representações e interpretações para essas metodologias.

4. Discussão das Metodologias para Sistemas com RV com seus Proponentes

4.1. Considerações Iniciais

Diversos contatos com os proponentes das metodologias VRID, CLEVR, de Stuart e de Scaife e Rogers foram feitos, através de e-mails, para que a análise crítica realizada sob a ótica deste trabalho pudesse ser validada e adequada sob a ótica dos autores dessas metodologias.

Dentre o conteúdo desses e-mails, devem ser destacados diversos questionários específicos, conforme anexos os A, B, C e D, elaborados por esta pesquisadora e enviados aos proponentes das metodologias para sistemas com RV analisadas, uma vez que permitiram confirmar não apenas a análise crítica decorrente deste estudo, mas também novas interpretações e representações sugeridas por este trabalho para essas metodologias. Esses questionários específicos, contendo questões referentes especificamente a cada metodologia, foram desenvolvidos a partir de um questionário base, conforme o apêndice A, contendo questões referentes a todas as metodologias para sistemas com RV analisadas. Esse questionário base foi composto por questões das áreas de RV, ES e IHC.

Dessa forma, este capítulo se destina à apresentação das discussões com os proponentes das metodologias para sistemas com RV citadas na literatura.

Inicialmente, no item 4.2 é apresentada a discussão com os proponentes do modelo e metodologia VRID com base no questionário específico enviado. O item 4.3 discute a metodologia CLEVR, apresentando uma nova interpretação e representação para essa metodologia. No item 4.4 é apresentada a discussão com o proponente da metodologia de Stuart com base no questionário específico enviado, retratando uma nova interpretação e representação. O item 4.5 discute a metodologia de Scaife e Rogers, apresentando uma nova interpretação e representação para essa metodologia. Por fim, no item 4.6 são realizadas as considerações finais sobre as discussões com os proponentes das metodologias para sistema com RV analisadas.

4.2. Modelo e Metodologia VRID

Para a discussão do modelo e metodologia VRID, apenas um questionário, apresentado no anexo A, foi elaborado e enviado por esta pesquisadora e respondido pelos autores desse modelo e metodologia.

De acordo com a análise crítica realizada neste estudo e a discussão com os proponentes do modelo e metodologia VRID, através desse questionário, confirma-se que, segundo este trabalho e segundo os autores dessa metodologia:

- A abordagem VRID é proposta como um modelo e metodologia apenas para projetar interfaces com RV.
- Dentre as fases genéricas de definição, desenvolvimento e manutenção previstas pela ES para o processo de sistemas, a abordagem VRID foca somente o passo de projeto de sistemas com RV, mais especificamente, de interfaces com RV e, desse modo, não é possível que seja feita a correspondência entre as fases genéricas da ES e as fases da metodologia VRID.
- O projeto de interfaces de sistemas com RV exige uma abordagem iterativa.
- A necessidade de novas iterações e refinamentos entre as fases e os níveis de detalhamentos previstos pela metodologia VRID é determinada através de avaliação de projeto (como através de avaliação heurística) e de discussão entre projetistas de interface e desenvolvedores de sistema com RV.
- Nessa abordagem, projetistas de interface com RV trabalham o projeto de interface e produzem especificações prontas que possam ser implementadas pelos desenvolvedores de sistemas com RV. As especificações produzidas pelos projetistas através da utilização do modelo e metodologia VRID auxiliam os desenvolvedores na implementação de sistemas com RV. Apesar de ser possível que uma mesma pessoa desempenhe ambos os papéis, de projetista e de desenvolvedor, se esses papéis forem desempenhados por pessoas distintas, as especificações de projeto podem prover um bom mecanismo de comunicação entre projetista e desenvolvedor.
- Nessa metodologia, quando usuários finais são envolvidos no processo de projeto, é importante que projetistas utilizem a metodologia VRID em conjunto com metodologias de projeto e avaliação centradas no usuário. As especificações de projeto produzidas na fase Projeto de Baixo Nível de Abstração da metodologia VRID objetivam melhorar a comunicação entre projetistas e desenvolvedores de sistemas com RV, e quando usuários finais são envolvidos no processo de projeto,

objetivam também auxiliar a comunicação com esses usuários, ajustando-se, nesse caso, aos princípios do DP.

- As estruturas e linguagens propostas para as especificações de projeto do modelo e metodologia VRID podem envolver ou motivar a atividade de prototipação.
- O projeto de interfaces com RV proposto pela metodologia VRID não está fundamentado sobre um modelo de processo da ES.
- O projeto proposto por essa metodologia não está fundamentado sobre uma abordagem da IHC.

Vale notar que essa discussão direciona para a consideração do enfoque no usuário e da usabilidade no processo de projeto de interface sugerido pela metodologia VRID, sugerindo que metodologias de projeto e avaliação centradas no usuário sejam utilizadas para permitir o envolvimento do usuário nesse processo e que avaliações heurísticas sejam realizadas para a definir a necessidade de novas iterações e refinamentos, as quais, conforme a análise crítica e comparativa realizada neste trabalho, não estavam presentes na metodologia de VRID.

4.3. Metodologia CLEVR

Para a discussão da metodologia CLEVR, dois questionários, conforme anexo B, foram elaborados e enviados por esta pesquisadora e respondidos pelos autores dessa metodologia.

Conforme a análise crítica realizada neste estudo e a discussão com os proponentes da metodologia CLEVR, através do primeiro questionário, confirma-se que, segundo este trabalho e segundo os autores dessa metodologia:

- A abordagem CLEVR é proposta como uma metodologia.
- O processo de sistemas com RV exige uma abordagem iterativa.
- As iterações de requisito, projeto, e validação são realizadas continuamente em cada etapa prevista pela metodologia CLEVR, ocorrendo na primeira etapa, na segunda etapa e na terceira etapa de seu processo.
- O processo proposto por essa metodologia segue os princípios do desenvolvimento incremental e do desenvolvimento hierárquico.
- O processo de sistemas com RV proposto pela metodologia CLEVR não está fortemente fundamentado sobre o modelo de processo Espiral da ES.
- O processo proposto por essa metodologia não está fundamentado sobre uma abordagem da IHC.

Ainda conforme a discussão com os proponentes da metodologia CLEVR através do primeiro questionário, apenas conforme os autores dessa metodologia e não conforme a análise deste trabalho, inicialmente sugeriu-se que:

- Dentre as fases genéricas de definição, desenvolvimento e manutenção previstas pela ES para o processo de sistemas, todas essas fases eram previstas pela metodologia CLEVR para o processo de sistemas com RV, considerando-se que a fase genérica de definição correspondia à etapa de Requisitos da metodologia CLEVR, a de desenvolvimento correspondia à de Projeto e a fase genérica de manutenção correspondia à etapa de Validação da metodologia CLEVR.
- O número de etapas e seus limites não eram importantes na metodologia CLEVR, uma vez que essa divisão de etapas foi utilizada apenas para explicar essa metodologia.
- Nessa metodologia, a atividade de prototipação e o desenvolvimento não eram diferenciados.

No entanto, a partir da análise crítica realizada neste estudo, uma nova interpretação sobre a abrangência da metodologia CLEVR e uma nova representação para as etapas dessa metodologia foram sugeridas por este trabalho aos autores da metodologia CLEVR, os quais as confirmaram.

Conforme a nova discussão com os proponentes da metodologia CLEVR, através do segundo questionário, confirma-se finalmente que, de acordo com este trabalho e com a confirmação dos autores dessa metodologia:

- A metodologia CLEVR abrange apenas as fases genéricas de definição e de desenvolvimento previstas pela ES, considerando-se que a fase de definição corresponde à etapa de Requisitos da metodologia CLEVR e a fase de desenvolvimento corresponde às etapas de Projeto e Validação da metodologia CLEVR. A fase genérica de manutenção para o processo de sistemas ocorre em sistemas já existentes e a etapa de Validação da metodologia CLEVR ocorre em sistemas em desenvolvimento e, desse modo, faz parte da fase genérica de desenvolvimento.
- Apesar da metodologia CLEVR ser proposta inicialmente sobre o modelo de processo Espiral da ES, essa metodologia pode ser representada por uma estrutura iterativa de processo independente desse modelo de processo, com número de etapas e limites mais definidos. Observa-se que a atividade de prototipação não precisa estar diferenciada do desenvolvimento, ou seja, essa atividade pode ser considerada como

um passo do desenvolvimento da metodologia CLEVR. No entanto, a atividade de prototipação deve permitir avaliações com o usuário, para que princípios da IHC sejam inseridos no processo proposto por essa metodologia, tornando-a mais completa.

Desse modo, com base na análise crítica deste estudo e na análise das respostas dos autores da metodologia CLEVR às questões respectivas a essa metodologia, tem-se uma nova interpretação e representação para a metodologia CLEVR, mostrada na figura 4.1, sugerida por este trabalho e ratificada pelos autores dessa metodologia.

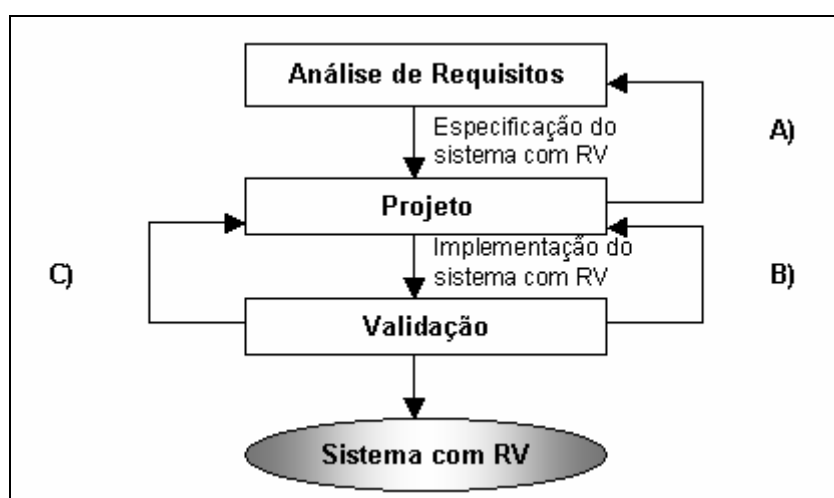


Figura 4.1 - Nova Interpretação e Representação para a Metodologia CLEVR

Pela figura 4.1, nessa nova interpretação e representação da metodologia CLEVR, suas iterações de requisito, projeto e validação são mantidas através da percepção de cada passo dessas iterações como uma etapa no processo dessa metodologia e três ciclos que permitem iteratividade são identificados e acrescentados. Esses ciclos eliminam a necessidade de se possuir um conjunto completo de especificações de requisitos para o sistema com RV antes do início de sua implementação, possibilitando que mudanças possam ser feitas de modo simples, mesmo que posteriormente no processo de sistema, mantendo o que é chamado de originalmente de desenvolvimento incremental pelos autores dessa metodologia, bem como consideram características críticas de forma, função e comportamento de objetos de maneira *top-down*, mantendo o que é chamado originalmente de desenvolvimento hierárquico pelos autores dessa metodologia, e de maneira *bottom-up* no processo de sistemas com RV. Nessa nova interpretação e representação da metodologia CLEVR, os resultados encontrados ao final de cada etapa são acrescentados.

Ainda pela figura 4.1, os três ciclos identificados e acrescentados à metodologia CLEVR são descritos a seguir:

- A) Análise de Requisitos:** iterações são focadas em aspectos usuais e da tecnologia de RV como identificação dos objetos e cenas, modelagem e refinamento da hierarquia de classes, das formas, funções e comportamentos dos objetos e das interações.
- B) Projeto:** envolve iterações que endereçam simulação e execução incremental para validar e revisar os modelos de especificação e prever o desempenho aproximado do sistema e para transformar esses modelos de especificação em um modelo de implementação que possa ser compilado e executado na plataforma alvo.
- C) Validação:** iterações são focadas na melhoria da característica de imersão, decidindo empregar ou não elementos adicionais conhecidos por afetar a sensação de presença como aumento das entradas e saídas, variação das formas para o processo de interação, minimização da distração e utilização de efeitos especiais.

Vale ressaltar que essa nova interpretação e representação reproduzem o processo de sistemas com RV sugerido pela metodologia CLEVR de modo fiel à sua real aplicação, apesar de não sugerir o modelo de processo Espiral da ES como fundamentação para essa metodologia, a qual, conforme a análise crítica e comparativa realizada neste trabalho, não está bem fundamentada sobre esse modelo de processo.

4.4. Metodologia de Stuart

Para a discussão da metodologia de Stuart, dois questionários, apresentados no anexo C, foram elaborados e enviados por esta pesquisadora e respondidos pelo autor dessa metodologia.

De acordo com a análise crítica realizada neste estudo a discussão com o proponente da metodologia de Stuart, através do primeiro questionário, confirma-se que, segundo este trabalho e segundo o autor dessa metodologia:

- A abordagem de Stuart é classificada como metodologia de projeto.
- O processo de sistemas com RV exige uma abordagem iterativa.
- A metodologia de Stuart envolve iterações de definição de requisitos, prototipação, avaliação com usuários e revisão de projeto e de protótipo até que o projeto seja validado. A partir dessa validação, o projeto deve ser implementado e o sistema deve ser avaliado e seus erros corrigidos.

- Em alto nível de abstração, algumas das descrições relacionadas mais especificamente à tecnologia de RV, e não ao processo de sistemas com RV, podem ser suprimidas nessa metodologia, para que, desse modo, somente suas implicações possam ser ponderadas no processo de sistemas com RV, bem como alguns dos detalhes dos sistemas perceptuais humanos não precisam receber o foco de desenvolvimento. Porém, em baixo nível de abstração, na especificação de requisitos para o projeto, esses detalhes devem ser considerados. É possível planejar questões gerais sobre o sistema que está sendo projetado antes que haja preocupação com dispositivos específicos de entrada e saída que serão usados no processo de interação entre usuário e o sistema. Porém, deve haver cuidado para que essa preocupação não ocorra em um momento muito tardio no processo de projeto do sistema. Observa-se que caso esteja sendo criado um sistema com RV sem nenhum componente de determinada percepção, detalhes desse sistema de percepção humano não são considerados em nenhum dos níveis de abstração.
- O processo de sistemas com RV proposto por essa metodologia está fortemente fundamentado sobre a abordagem do Projeto Iterativo.
- Algumas considerações para aplicar essa abordagem no processo de sistemas com RV devem ser ponderadas:
 - A falta de ferramentas de prototipação rápida de sistemas com RV;
 - A diversidade de possibilidades tecnológicas e de interação em interfaces com RV se comparadas com as demais interfaces;
 - Uma área de pesquisa relativamente nova, existindo menos estudos conclusivos que direcionem projetistas em suas escolhas de acordo com essa diversidade de possibilidades tecnológicas e de interação.

Ainda de acordo com a discussão com o proponente da metodologia de Stuart através do primeiro questionário, apenas conforme o autor dessa metodologia e não conforme a análise deste trabalho, inicialmente sugeriu-se que:

- Dentre as fases genéricas de definição, de desenvolvimento e de manutenção previstas pela ES para o processo de sistemas, apenas a fase de definição era prevista pela metodologia de Stuart para o processo de sistemas com RV, considerando-se que a fase genérica de definição correspondia ao passo de Definição de Requisitos da metodologia de Stuart, seguida pelos passos de Prototipação, Avaliação com usuários, Revisão de projeto e do protótipo e Implementação do sistema.

No entanto, a partir da análise crítica realizada neste estudo, uma nova interpretação para a abrangência da metodologia de Stuart foi sugerida ao autor dessa metodologia, que a confirmou.

Conforme a nova discussão com o proponente da metodologia de Stuart através do segundo questionário, confirma-se que, de acordo com este trabalho e com a confirmação do autor dessa metodologia:

- A metodologia de Stuart abrange além da fase de definição, a fase de desenvolvimento prevista pela ES, considerando-se que a fase de definição corresponde ao passo de Definição de Requisitos da metodologia de Stuart e a fase de desenvolvimento corresponde aos passos Projeto, Prototipação, Avaliação com Usuários, Revisão de Projeto e do Protótipo e Implementação da metodologia de Stuart.
- Algumas considerações para aplicar a metodologia de Stuart, considerando as fases genéricas de definição e desenvolvimento, no processo de sistemas com RV devem ser ponderadas:
 - O código criado na prototipação do sistema pode ser totalmente diferente do código implementado no sistema entregue;
 - Os passos da fase genérica de desenvolvimento (projeto de software, codificação, realização de testes de software) não devem sugerir um modelo convencional da ES, uma vez que o Projeto Iterativo deve ser seguido.
 - O processo apresentado inicialmente na metodologia de Stuart deve ser usado para gerar o projeto. A partir desse projeto devem ser realizadas sua implementação, avaliação e correção de possíveis erros.
- Além de estar fundamentado sobre a abordagem Projeto Iterativo, o processo de sistemas com RV proposto pela metodologia de Stuart também está fundamentado sobre a abordagem PCU da IHC, apresentando o enfoque no usuário. Contudo, não se pensou em adotar a abordagem PCU formalmente, ou seja, se pensou sobre manter o enfoque no usuário de modo informal (intuitivamente) quando essa metodologia foi desenvolvida por seu autor.
- Além de estar fundamentado também sobre o PCU, o processo proposto por essa metodologia, está conseqüentemente, conforme discutido neste trabalho, fundamentado sobre o modelo de processo de Prototipação da ES, apresentando suas características.

Dessa forma, com base na análise crítica deste trabalho e na análise das respostas do autor da metodologia de Stuart às questões respectivas a essa metodologia, tem-se uma nova interpretação e representação para a metodologia de Stuart, apresentada na figura 4.2, sugerida por este trabalho.

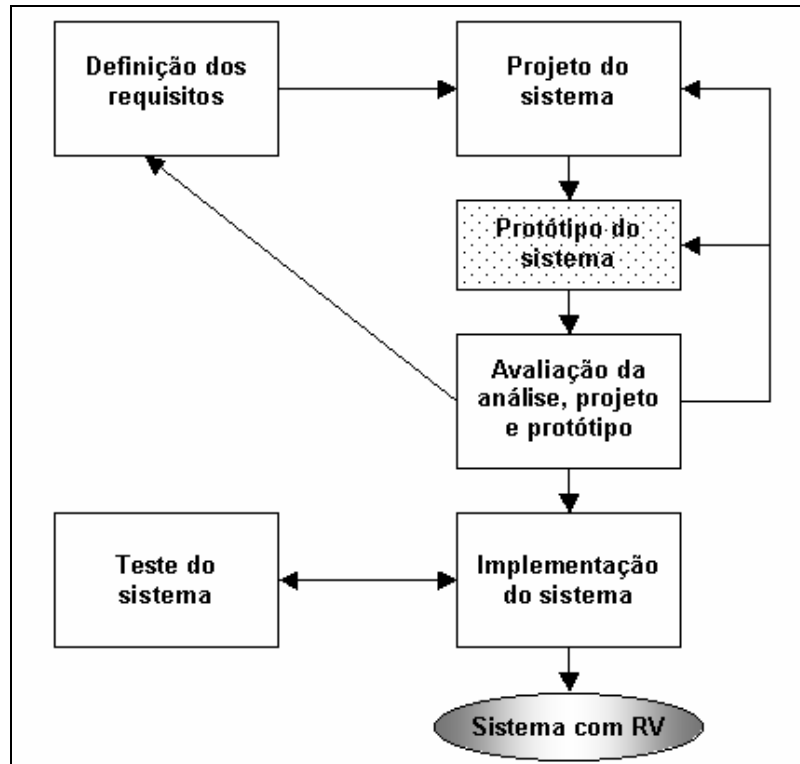


Figura 4.2 - Nova Interpretação e Representação para a Metodologia de Stuart

Conforme a figura 4.2, nessa nova interpretação e representação para a metodologia de Stuart, os passos de Definição de requisitos, Prototipação, Projeto e Avaliação são mantidos, dois outros passos e dois ciclos são acrescentados. As etapas de Implementação do sistema e Teste do sistema permitem que as atividades de implementação, testes e correções de possíveis erros estejam claramente inseridas e representadas nessa metodologia, a qual, desse modo, abrange todo o processo de sistemas com RV. Os dois ciclos acrescentados a partir do passo de Avaliação em direção aos passos de Prototipação e de Projeto permitem que as possíveis iterações decorrentes da necessidade de se ajustar não apenas a análise de requisitos, mas ainda o projeto e ou o protótipo, também estejam claramente inseridas e representadas nessa metodologia. Ainda pela figura 4.3, nessa nova interpretação e representação da metodologia de Stuart, os resultados encontrados ao final de cada etapa são acrescentados.

Vale ressaltar que essa nova interpretação e representação reproduzem o processo de sistemas com RV sugerido pela metodologia de Stuart de modo fiel à sua real aplicação, apesar de inserir ciclos nessa metodologia, os quais, conforme a análise crítica deste trabalho, não estavam representados na metodologia de Stuart.

4.5. Metodologia de Scaife e Rogers

Para a discussão da metodologia de Scaife e Rogers, um questionário, conforme anexo D, foi elaborado e enviado por esta pesquisadora e respondido pelos autores dessa metodologia.

Conforme a análise crítica realizada neste estudo e a discussão com os proponentes da metodologia de Scaife e Rogers, através desse questionário, confirma-se que, segundo os autores dessa metodologia e segundo este trabalho:

- A abordagem de Scaife e Rogers é classificada como metodologia.
- A metodologia de Scaife e Rogers pode ser utilizada para o processo de outras aplicações com RV além do contexto de aprendizado, o qual foi o principal domínio de aplicação utilizado durante o desenvolvimento dessa metodologia, sendo, dessa forma, genérica.
- Dentre as fases genéricas de definição, desenvolvimento e manutenção previstas pela ES para o processo de sistemas, define-se que as fases de definição e desenvolvimento são previstas pela metodologia de Scaife e Rogers para o processo de sistemas com RV, considerando-se que a fase de definição corresponde às etapas Levantamento dos requisitos de alto nível e Estudos exploratórios e Informantes de projeto da metodologia de Scaife e Rogers e que a fase de desenvolvimento corresponde às etapas de Prototipação, Testes com usuários e Projeto do sistema com RV, Especificação do modelo conceitual e Implementação e Avaliação da metodologia de Scaife e Rogers.
- O processo de sistemas com RV proposto por essa metodologia está fortemente fundamentado sobre o PCU.
- Além de estar fundamentado sobre o PCU, o processo de sistemas com RV proposto pela metodologia de Scaife e Rogers, está conseqüentemente, conforme discutido neste trabalho, fundamentado sobre o modelo de processo de Prototipação da ES, apresentando suas características.

- Apesar da metodologia de Scaife e Rogers ser proposta inicialmente sobre a estrutura linear de processo de sistemas, para atender as necessidades da estrutura iterativa de processo exigida por sistemas com RV, alguns ciclos podem ser adicionados a estrutura linear de processo da metodologia de Scaife e Rogers, para tornar seu processo iterativo e fiel à sua real aplicação.

Desse modo, com base na análise crítica deste estudo e na análise das respostas dos autores da metodologia de Scaife e Rogers às questões respectivas a essa metodologia, tem-se uma nova interpretação e representação para a metodologia de Scaife e Rogers, mostrada na figura 4.3, sugerida por este trabalho e ratificada pelos autores dessa metodologia.

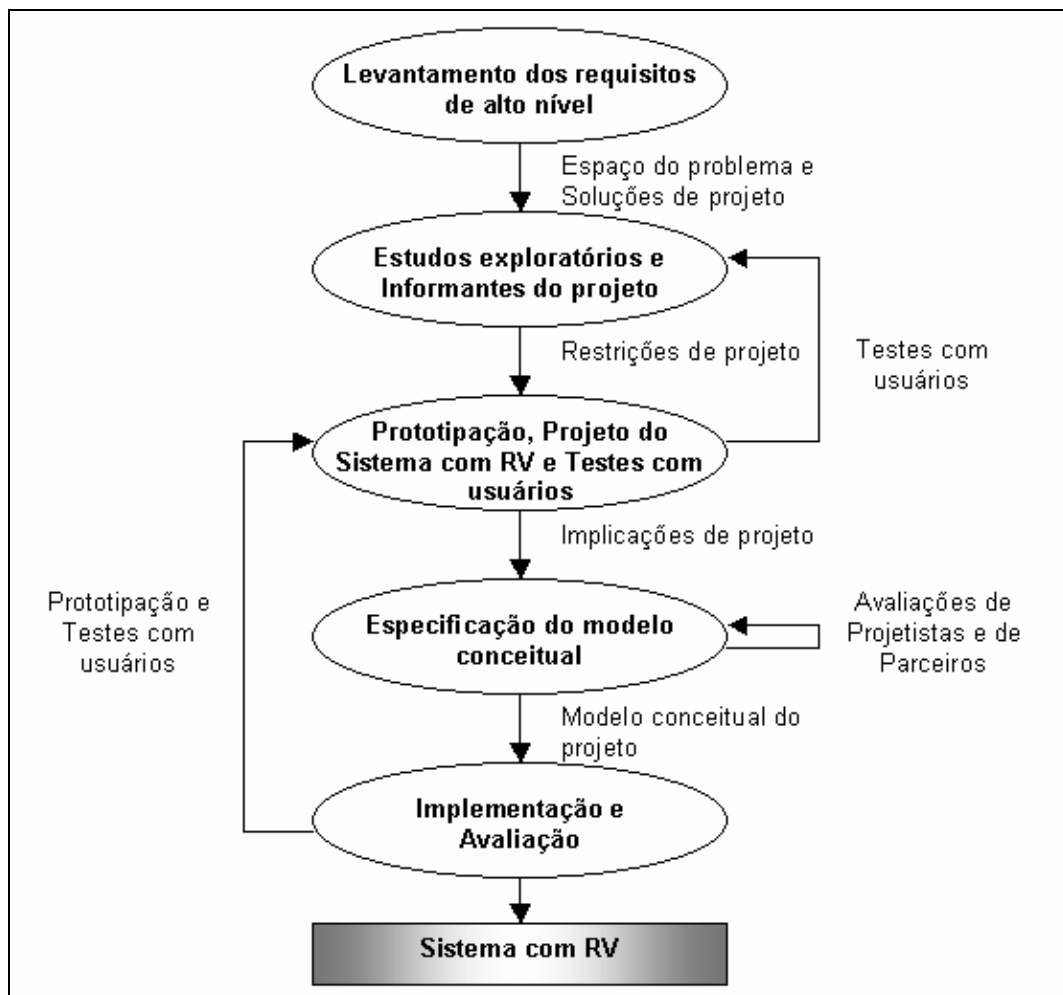


Figura 4.3 - Nova Interpretação e Representação para a Metodologia de Scaife e Rogers

Pela figura 4.3, nessa nova interpretação e representação da metodologia de Scaife e Rogers, suas cinco etapas são mantidas e três ciclos que permitem iteratividade são identificados e acrescentados. As atividades realizadas em cada etapa são mantidas na nova

interpretação dessa metodologia, porém suprimidas na nova representação da metodologia de Scaife e Rogers, pois a realização dessas atividades é fundamental no processo proposto pela metodologia de Scaife e Rogers, mas sua representação não contribui para o entendimento claro e objetivo dessa metodologia. Nessa nova interpretação e representação da metodologia de Scaife e Rogers, os resultados encontrados ao final de cada etapa são acrescentados.

Ainda pela figura 4.3, os três ciclos identificados e acrescentados à metodologia de Scaife e Rogers são descritos a seguir:

- **Testes com Usuários:** durante a terceira etapa de projeto, requisitos não especificados inicialmente podem ser descobertos nos testes com usuários, exigindo novos estudos exploratórios e novas entrevistas com informantes de projeto.
- **Avaliação de Projetistas e de Parceiros:** na quarta etapa de projeto, avaliações entre projetistas e parceiros de projeto são realizadas para que o modelo conceitual do sistema com RV seja especificado de acordo com interesses de ambas as partes.
- **Prototipação e Teste com usuários:** durante a quinta etapa de projeto, novos protótipos podem ser construídos e utilizados nos testes com usuários, possibilitando que requisitos de projeto de um sistema com RV possam ser novamente refinados.

Vale ressaltar que essa nova interpretação e representação reproduzem o processo de sistemas com RV sugerido pela metodologia de Scaife e Rogers de modo fiel à sua real aplicação, atendendo as necessidades de uma estrutura iterativa de processo, a qual, conforme a análise crítica deste trabalho, não estava presente na metodologia de Scaife e Rogers.

4.6. Considerações Finais

Neste capítulo, foram apresentadas as discussões realizadas entre esta pesquisadora e os proponentes das metodologias para sistemas com RV VRID, CLEVR, de Stuart e de Scaife e Rogers.

Todas as considerações realizadas por este trabalho sobre as metodologias para sistemas com RV analisadas, bem como as novas interpretações e representações para essas metodologias são sugeridas a fim de não apenas reproduzir o processo realizado durante a utilização dessas metodologias no processo de sistemas com RV de modo fiel à sua aplicação, mas, principalmente, a fim de melhor adequar essas metodologias às exigências no processo desses sistemas.

As discussões realizadas entre este estudo e os proponentes das metodologias para sistemas com RV VRID, CLEVR, de Stuart e de Scaife e Rogers contribuíram significativamente para a construção do meta-modelo para sistemas com RV proposto por este trabalho no capítulo seguinte.

5. Meta-modelo para o Processo de Sistemas com RV

5.1. Considerações Iniciais

Como principal resultado deste trabalho, este capítulo é destinado à construção de um meta-modelo para o processo de sistemas com RV [Olher et al 2004], meta por ser mais geral e abrangente, considerando, inicialmente, enfoque no usuário, iteratividade de projeto e, posteriormente, usabilidade através da inserção dos critérios de usabilidade para esses sistemas ao longo de seu processo.

Inicialmente, no item 5.2 é apresentado o meta-modelo para sistemas com RV com enfoque no usuário e iteratividade de projeto, retratando as contribuições das metodologias para sistemas com RV VRID, CLEVR, de Stuart e de Scaife e Rogers na construção para esse meta-modelo. O meta-modelo para sistemas com RV com enfoque no usuário, iteratividade de projeto e critérios de usabilidade é retratado no item 5.3, apresentando critérios de usabilidade específicos para esses sistemas. Por fim, no item 5.4 são realizadas algumas considerações finais sobre o meta-modelo para sistemas com RV.

5.2. Meta-modelo com Enfoque no Usuário e Iteratividade de Projeto

A construção do meta-modelo para sistemas com RV é iniciada a partir da análise comparativa das fases ou etapas ou passos, bem como atividades que compõem as metodologias para sistemas com RV citadas na literatura, incorporando as características de cada uma dessas metodologias que melhor atendem às exigências de enfoque do usuário e iteratividade de projeto ao longo do processo de sistemas com RV.

5.2.1. Contribuições da Metodologia VRID

Inicialmente, são analisadas as contribuições da metodologia VRID, conforme a tabela 5.1, para o meta-modelo para sistemas com RV proposto.

Tabela 5.1 - Contribuições da Metodologia VRID para Compor o Meta-modelo

Fases e Atividades da Metodologia VRID	Etapas sugeridas para o Meta-modelo	Atividades sugeridas para as Etapas do Meta-modelo
<p>Projeto de alto nível de abstração</p> <ul style="list-style-type: none"> - Identificar elementos de dados e objetos de interface na descrição do sistema, decidindo a legitimidade desses objetos; - Distinguir objetos de interface em objetos físicos e objetos virtuais; - Modelar cada objeto virtual, em alto nível de abstração, utilizando o modelo VRID; - Avaliar o projeto da interface. <p>Projeto de baixo nível de abstração</p> <ul style="list-style-type: none"> - Remodelar cada objeto virtual, em baixo nível de abstração, refinando e detalhando de modo formal as especificações; - Avaliar o projeto da interface; - Discutir o projeto da interface com os envolvidos; - Determinar a necessidade de nova iteração. 	Projeto	<ul style="list-style-type: none"> - Especificar ou modelar os componentes de interface e de interação; - Refinar o projeto da interface.
	Avaliação de Projeto	<ul style="list-style-type: none"> - Avaliar o projeto da interface; - Reportar os problemas de projeto, determinando se há iteração a partir do Projeto.

A tabela 5.1, sobre a metodologia VRID, mostra que as fases Projeto de alto nível de abstração e Projeto de baixo nível de abstração contribuem para sugerir as etapas Projeto e Avaliação de Projeto para o meta-modelo, apresentando atividades distintas de projeto e de avaliação de interface, que podem ser distribuídas através das duas etapas sugeridas.

5.2.2. Contribuições da Metodologia CLEVR

As contribuições da metodologia CLEVR para o meta-modelo para sistemas com RV proposto são analisadas como mostra a tabela 5.2.

Tabela 5.2 - Contribuições da Metodologia CLEVR para Compor o Meta-modelo

Etapas e Atividades da Metodologia CLEVR	Etapas sugeridas para o Meta-modelo	Atividades sugeridas para as Etapas do Meta-modelo
Análise de Requisitos - Identificar objetos e cenas do sistema; - Identificar classes dos objetos do sistema; - Modelar a visão geral do sistema.	Análise de Requisitos	- Definir e delimitar o espaço do problema; - Refinar os requisitos.
Projeto e Implementação - Modelar e refinar funções dos objetos, seus comportamentos e interações do usuário e suas formas e cenas do sistema; - Refinar os modelos de especificação, para melhorar sua característica de imersão; - Simular os modelos de especificação e ou traduzir os modelos de especificação em um modelo de implementação.	Projeto	- Especificar ou modelar as funcionalidades; - Especificar ou modelar os componentes de interface e de interação; - Refinar o projeto.
	Implementação	- Implementar o projeto; - Corrigir o sistema com base na Avaliação de Sistema.
Validação - Estimar o desempenho do sistema e selecionar os modelos de especificação; - Testar o sistema; - Avaliar a característica de imersão; - Determinar a necessidade de nova iteração.	Avaliação de Requisitos e Projeto	- Testar o sistema; - Avaliar o desempenho do sistema; - Reportar os problemas do sistema, determinando se há iteração a partir da Análise de Requisitos ou de Projeto.
	Avaliação de Sistema	- Testar o sistema; - Avaliar o desempenho do sistema; - Reportar os problemas do sistema, determinando se há iteração a partir da Análise de Requisitos ou de Projeto.

A tabela 5.2, sobre a metodologia CLEVR com as considerações realizadas por este estudo, mostra que a etapa de Requisitos contribui para sugerir a etapa de Análise de Requisitos para o meta-modelo. A etapa de Projeto e Implementação contribui para sugerir as etapas de Projeto e de Implementação, apresentando atividades de projeto e de implementação distintas, que podem ser distribuídas através das etapas sugeridas. A tabela 5.2 mostra ainda que a etapa de Validação de Requisitos, Projeto e Sistema contribui para sugerir as etapas de Avaliação de Requisitos e Projeto e de Avaliação de Sistema para o meta-modelo, apresentando atividades distintas de avaliação de requisitos e projeto e de avaliação de sistema, que também podem ser distribuídas através das etapas sugeridas.

5.2.3. Contribuições da Metodologia de Stuart

As contribuições da metodologia de Stuart para o meta-modelo para sistemas com RV proposto são analisadas conforme a tabela 5.3.

Tabela 5.3 - Contribuições da Metodologia de Stuart para Compor o Meta-modelo

Passos e Atividades da Metodologia de Stuart	Etapas sugeridas para o Meta-modelo	Atividades sugeridas para as Etapas do Meta-modelo
Definição de requisitos - Definir capacidades e desempenho humano; - Definir usuários, tarefas e ambiente; - Definir requisitos funcionais, práticos e específicos para o sistema.	Análise de Requisitos	- Definir os usuários, suas capacidades, as tarefas e o ambiente; - Definir os requisitos funcionais, práticos e específicos do sistema; - Refinar os requisitos.
Projeto do sistema - Projetar ou selecionar tecnologias de entrada e saída, arquitetura e tecnologias computacionais; - Projetar objetos, comportamentos e interações. Prototipação do sistema - Construir protótipo.	Projeto (baseado em prototipação)	- Projetar ou selecionar as tecnologias de entrada e saída, a arquitetura e as tecnologias computacionais; - Especificar ou modelar os componentes de interface e de interação; - Construir protótipos, retratando funcionalidades e componentes de interface e de interação projetados; - Refinar o projeto ou o protótipo.
Avaliação do sistema - Testar o sistema utilizando usuários; - Avaliar o desempenho e a usabilidade do sistema; - Avaliar o valor para tarefa e aplicação; - Determinar a necessidade de nova iteração.	Avaliação de Requisitos e Projeto	- Testar os protótipos utilizando usuários; - Avaliar o desempenho do protótipo e do usuário; - Avaliar a usabilidade do protótipo; - Reportar os problemas de requisitos, projeto e ou do protótipo, determinando se iteração a partir da Análise de Requisitos ou de Projeto (baseado em prototipação).
Implementação - Implementar o projeto; - Corrigir o sistema com base nos erros reportados.	Implementação	- Implementar o projeto; - Corrigir o sistema.
Testes - Testar o sistema; - Reportar erros do sistema; - Determinar a necessidade de nova iteração.	Avaliação de Sistema	- Testar o sistema; - Avaliar o desempenho do sistema e do usuário; - Avaliar a usabilidade do sistema; - Reportar os erros do sistema, determinando se há iteração a partir da Implementação.

A tabela 5.3, sobre a metodologia de Stuart com as considerações realizadas por este trabalho, apresenta que o passo de Definição de requisitos contribui para sugerir a etapa de Análise de Requisitos para o meta-modelo. Os passos de Projeto do sistema e de Prototipação do sistema contribuem para sugerir a etapa de Projeto (baseado em Prototipação), apresentando atividades de projeto e de prototipação que podem se fundir na etapa sugerida. O passo de Avaliação do sistema contribui para sugerir a etapa de Avaliação de Requisitos e Projeto. O passo de Implementação contribui para sugerir a etapa de Implementação. A tabela 5.3 mostra ainda que o passo de Testes contribui para sugerir a etapa de Avaliação de Sistema para o meta-modelo.

5.2.4. Contribuições da Metodologia de Scaife e Rogers

Finalmente, são analisadas as contribuições da metodologia de Scaife e Rogers para o meta-modelo para sistemas com RV proposto, como mostra a tabela 5.4.

Tabela 5.4 - Contribuições da Metodologia de Scaife e Rogers para Compor o Meta-modelo

Etapas e Atividades da Metodologia de Scaife e Rogers	Etapas sugeridas para o Meta-modelo	Atividades sugeridas para as Etapas do Meta-modelo
<p>Levantamento dos requisitos de alto nível</p> <ul style="list-style-type: none"> - Definir o espaço do problema; - Identificar benefícios da utilização da tecnologia de RV; - Analisar pesquisas, teorias e possibilidades tecnológicas; - Definir usuários e suas dificuldades no espaço do problema. <p>Informantes de projeto e Estudos exploratórios</p> <ul style="list-style-type: none"> - Solicitar aos informantes de projeto suas visões do espaço do problema; - Realizar estudos exploratórios, identificando práticas atuais de representação, interação e controle e possibilidades futuras. 	Análise de Requisitos	<ul style="list-style-type: none"> - Definir e delimitar o espaço do problema; - Identificar os benefícios da utilização da tecnologia de RV; - Identificar e analisar pesquisas, teorias e possibilidades tecnológicas existentes; - Definir os usuários, suas capacidades, as tarefas e o ambiente; - Solicitar aos informantes de projeto suas visões do espaço do problema; - Realizar estudos exploratórios, identificando práticas atuais de representação, interação e controle e possibilidades futuras; - Refinar os requisitos.
<p>Prototipação, Projeto do sistema com RV e Testes com usuários</p> <ul style="list-style-type: none"> - Construir protótipos das funcionalidades e dos componentes de interface e de interação; - Testar os protótipos utilizando usuários; - Avaliar o desempenho do usuário; - Analisar e descartar idéias de projeto, solucionando possíveis problemas de projeto; - Determinar a necessidade de nova iteração. <p>Especificação do modelo conceitual</p> <ul style="list-style-type: none"> - Especificar o modelo conceitual; - Discutir e refinar o modelo conceitual com parceiros. 	Projeto (baseado em Prototipação)	<ul style="list-style-type: none"> - Construir protótipos, com ou sem o apoio de tecnologia computacional, retratando funcionalidades e componentes de interface e de interação identificados; - Descartar idéias de projeto; - Especificar o modelo conceitual; - Discutir e refinar o modelo conceitual com parceiros. - Refinar o projeto.
	Avaliação de Requisitos e Projeto	<ul style="list-style-type: none"> - Testar os protótipos utilizando usuários; - Avaliar o desempenho do usuário; - Avaliar a usabilidade do protótipo; - Reportar os problemas de requisitos e de projeto, determinando se há necessidade de nova iteração a partir da Análise de Requisitos.
<p>Implementação e Avaliação</p> <ul style="list-style-type: none"> - Implementar o projeto; - Avaliar o sistema nas várias etapas como é iterado; - Determinar a necessidade de nova iteração. 	Implementação	<ul style="list-style-type: none"> - Implementar o projeto; - Corrigir o sistema.
	Avaliação de Sistema	<ul style="list-style-type: none"> - Testar o sistema; - Avaliar o desempenho do usuário; - Avaliar a usabilidade do sistema; - Reportar os erros do sistema e os problemas de requisitos, projeto e ou protótipo, determinando se há iteração a partir da Análise de Requisitos, do Projeto (baseado em prototipação) ou da Implementação.

A tabela 5.4, sobre a metodologia de Scaife e Rogers com as considerações realizadas por este estudo, mostra que as etapas de Levantamento dos requisitos de alto nível e de Informantes de projeto e Estudos exploratórios contribuem para sugerir a etapa de Análise de Requisitos para o meta-modelo, apresentando atividades de elicitação e análise de requisitos que podem se fundir na etapa sugerida. As etapas de Prototipação, Projeto do sistema com RV e Testes com usuários e de Especificação do modelo conceitual contribuem para sugerir a etapas de Projeto (baseado em Prototipação) e de Avaliação de Requisitos e Projeto, apresentando atividades distintas de projeto, de prototipação e de avaliação de requisitos e projeto, que podem ser distribuídas através das etapas sugeridas. A tabela 5.4 mostra ainda que a etapa de Implementação e Avaliação contribui para sugerir as etapas de Implementação e de Avaliação de Sistema para o meta-modelo, apresentando atividades de implementação e de avaliação de sistema distintas, que também podem ser distribuídas através das etapas sugeridas.

5.2.5. Interpretação e Representação do Meta-modelo com Enfoque no Usuário e Iteratividade de Projeto

A síntese de quais passos ou fases ou etapas das metodologias para sistema com RV compõem, quando analisadas em conjunto, cada etapa do meta-modelo, é apresentada na tabela 5.5.

Tabela 5.5 - Síntese das Contribuições das Metodologias para Sistemas com RV para Compor o Meta-modelo para Sistemas com RV

Etapas do Meta-modelo	Atividades das Etapas do Meta-modelo	Fases ou Passos ou Etapas, e Atividades das Metodologias que Contribuem para o Meta-modelo	Metodologias que Contribuem para o Meta-modelo
Análise de Requisitos	a. Definir e delimitar o espaço do problema;	Análise de Requisitos	CLEVR
	b. Identificar os benefícios da utilização da tecnologia de RV para o espaço do problema delimitado;	- Identificar objetos e cenas do sistema; - Identificar classes dos objetos do sistema; - Modelar a visão geral do sistema.	
	c. Identificar e analisar as pesquisas, teorias e possibilidades tecnológicas existentes para o espaço do problema;	Definição de requisitos	de Stuart
	d. Definir os usuários, suas capacidades, as tarefas e o ambiente;	- Definir capacidades e desempenho humano; - Definir usuários, tarefas e ambiente; - Definir requisitos funcionais, práticos e específicos para o sistema.	
Projeto (baseado em prototipação)	e. Definir os requisitos funcionais, práticos e específicos do sistema com RV;	Levantamento dos requisitos de alto nível	de Scaife e Rogers
	f. Solicitar aos informantes de projeto suas visões do espaço do problema delimitado;	- Definir o espaço do problema; - Identificar benefícios da utilização da tecnologia de RV; - Analisar pesquisas, teorias e possibilidades tecnológicas; - Definir usuários e suas dificuldades no espaço do problema.	
	g. Realizar estudos exploratórios, identificando as práticas atuais das formas de representação, dos mecanismos e dispositivos de interação e controle, da melhor combinação entre eles e as possibilidades futuras no espaço do problema;	Informantes de projeto e Estudos exploratórios	
	h. Refinar os requisitos do sistema com RV conforme o resultado da etapa de Avaliação de Requisitos e Projeto.	- Solicitar aos informantes de projeto suas visões do espaço do problema; - Realizar estudos exploratórios, identificando práticas atuais de representação, interação e controle e possibilidades futuras.	
Avaliação de Requisitos e Projeto	a. Construir protótipos do sistema com RV, com ou sem o apoio de tecnologia computacional como em papel ou em ferramenta computacional para prototipação em RV, retratando funcionalidades e componentes de interface e de interação identificados;	Projeto de alto nível de abstração	VRID
	b. Descartar idéias de projeto com base no resultado da etapa de Avaliação de Requisitos e Projeto;	- Identificar elementos de dados e objetos de interface na descrição do sistema, decidindo a legitimidade desses objetos; - Distinguir objetos de interface em objetos físicos e objetos virtuais; - Modelar cada objeto virtual, em alto nível de abstração, utilizando o modelo VRID.	
	c. Especificar o modelo conceitual do sistema com RV;	Projeto de baixo nível de abstração	CLEVR
	d. Discutir e refinar o modelo conceitual do sistema com RV com parceiros de projeto;	- Remodelar cada objeto virtual, em baixo nível de abstração, refinando e detalhando de modo formal as especificações.	
	e. Projetar ou selecionar as tecnologias de entrada e saída, a arquitetura e as tecnologias computacionais;	Projeto e Implementação	de Stuart
	f. Especificar ou modelar as funcionalidades do sistema com RV;	- Modelar e refinar funções dos objetos, seus comportamentos e interações do usuário e suas formas e cenas do sistema; - Refinar os modelos de especificação, para melhorar sua característica de imersão.	
Implementação	g. Especificar ou modelar os componentes de interface e de interação do sistema com RV;	Projeto do sistema	de Scaife e Rogers
	h. Construir protótipos do sistema com RV, com apoio de tecnologia computacional como ferramenta computacional para prototipação em RV, retratando funcionalidades e componentes de interface e de interação projetados;	- Projetar ou selecionar tecnologias de entrada e saída, arquitetura e tecnologias computacionais; - Projetar objetos, comportamentos e interações.	
	i. Refinar o projeto do sistema com RV conforme o resultado da etapa de Avaliação de Requisitos e Projeto.	Prototipação do sistema	
		- Construir protótipo.	
Avaliação de Sistema	a. Testar os protótipos do sistema com RV utilizando usuários ou profissionais de avaliação;	Prototipação, Projeto do sistema com RV e Testes com usuários	de Stuart
	b. Avaliar o desempenho do protótipo e do usuário;	- Construir protótipos das funcionalidades e dos componentes de interface e de interação; - Analisar e descartar idéias de projeto, solucionando possíveis problemas de projeto.	
	c. Avaliar a usabilidade do protótipo;	Especificação do modelo conceitual	de Scaife e Rogers
	d. Reportar os problemas de requisitos, projeto e ou do protótipo, determinando se há iteração a partir da etapa de Análise de Requisitos ou de Projeto (baseado em prototipação).	- Especificar o modelo conceitual; - Discutir e refinar o modelo conceitual com parceiros.	
Avaliação de Sistema	a. Implementar o projeto do sistema com RV	Projeto de alto nível de abstração	VRID
	b. Corrigir o sistema com RV conforme o resultado da etapa de Avaliação de Sistema.	- Avaliar o projeto da interface. Projeto de baixo nível de abstração	
		- Avaliar o projeto da interface; - Discutir o projeto da interface com os desenvolvedores; - Determinar a necessidade de nova iteração.	
		Validação	CLEVR
Avaliação de Sistema	a. Testar o sistema com RV utilizando usuários ou profissionais de avaliação;	- Estimar o desempenho do sistema e selecionar os modelos de especificação; - Testar o sistema; - Avaliar a característica de imersão; - Determinar a necessidade de nova iteração.	de Stuart
	b. Avaliar o desempenho do sistema;	Avaliação do sistema	
	c. Avaliar o desempenho do usuário;	- Testar o sistema utilizando usuários; - Avaliar o desempenho e a usabilidade do sistema; - Avaliar o valor para tarefa e aplicação; - Determinar a necessidade de nova iteração.	de Scaife e Rogers
	d. Avaliar a usabilidade do sistema;	Prototipação, Projeto do sistema com RV e Testes com usuários	
Avaliação de Sistema	e. Reportar os erros do sistema e os problemas ainda remanescentes de requisitos, projeto e ou do protótipo, determinando se há iteração a partir da etapa de Análise de Requisitos, de Projeto (baseado em prototipação) ou de Implementação.	- Testar os protótipos utilizando usuários; - Avaliar o desempenho do usuário; - Determinar a necessidade de nova iteração.	CLEVR
		Projeto e Implementação	de Stuart
		- Simular os modelos de especificação e ou traduzir os modelos de especificação em um modelo de implementação.	de Scaife e Rogers
		Implementação	
Avaliação de Sistema	a. Testar o sistema com RV utilizando usuários ou profissionais de avaliação;	- Implementar o projeto; - Corrigir o sistema com base nos erros reportados.	de Stuart
	b. Avaliar o desempenho do sistema;	Implementação e Avaliação	de Scaife e Rogers
	c. Avaliar o desempenho do usuário;	- Implementar o projeto.	
	d. Avaliar a usabilidade do sistema;	Validação	CLEVR
Avaliação de Sistema	e. Reportar os erros do sistema e os problemas ainda remanescentes de requisitos, projeto e ou do protótipo, determinando se há iteração a partir da etapa de Análise de Requisitos, de Projeto (baseado em prototipação) ou de Implementação.	- Testar o sistema; - Avaliar a característica de imersão; - Determinar a necessidade de nova iteração.	de Stuart
		Testes	
		- Testar o sistema; - Reportar erros do sistema; - Determinar a necessidade de nova iteração.	de Scaife e Rogers
		Implementação e Avaliação	
	- Avaliar o sistema nas várias etapas como é iterado; - Determinar a necessidade de nova iteração.		

Conforme a tabela 5.5, a etapa de Análise de Requisitos da metodologia CLEVR, o passo de Definição de requisitos da metodologia de Stuart e as etapas de Levantamento dos requisitos de alto nível e de Informantes de projeto e Estudos exploratórios da metodologia de Scaife e Rogers dão origem à etapa de Análise de Requisitos do meta-modelo, uma vez que apresentam como objetivo em comum a análise de requisitos do sistema com RV.

A etapa de Projeto da metodologia CLEVR, os passos de Projeto do sistema e de Prototipação do Sistema da metodologia de Stuart e as etapas de Prototipação, Projeto do Sistema com RV e Testes com Usuários e de Especificação do modelo conceitual da metodologia de Scaife e Rogers dão origem à etapa de Projeto (baseado em prototipação) do meta-modelo, pois têm, dentre seus objetivos específicos, o objetivo em comum de especificar o projeto do sistema com RV.

A etapa de Validação da metodologia CLEVR, o passo de Avaliação do sistema da metodologia de Stuart e a etapa de Prototipação, Projeto do Sistema com RV e Testes com Usuários da metodologia de Scaife e Rogers dão origem à etapa de Avaliação de Requisitos e Projeto do meta-modelo, uma vez que apresentam como objetivo em comum avaliar os requisitos e o projeto do sistema com RV.

A etapa de Projeto da metodologia CLEVR, o passo de Implementação da metodologia de Stuart e a etapa de Implementação e Avaliação da metodologia de Scaife e Rogers dão origem à etapa de Implementação do meta-modelo, pois têm, dentre seus objetivos específicos, o objetivo em comum de implementar o projeto do sistema com RV.

Ainda conforme a tabela 5.5, a etapa de Validação da metodologia CLEVR, o passo de Testes da metodologia de Stuart e a etapa de Implementação e Avaliação da metodologia de Scaife e Rogers dão origem à etapa de Avaliação de Sistema do meta-modelo, uma vez que apresentam como objetivo em comum avaliar o sistema com RV desenvolvido. Além dessas considerações, nota-se que, de acordo com a análise realizada neste trabalho:

- 1) A metodologia VRID sugere apenas as etapas de Projeto e de Avaliação de Projeto, já que contém atividades para projeto e avaliação de interface de sistema com RV,
- 2) A metodologia CLEVR sugere a etapa de Projeto sem prototipação, ao passo que as metodologias de Stuart e de Scaife e Rogers sugerem a etapa de Projeto com a atividade de prototipação, apresentando enfoque em usabilidade e no usuário e, desse modo, essa atividade é adotada neste trabalho;
- 3) As metodologias CLEVR, de Stuart e de Scaife e Rogers, conforme as novas interpretações e sugestões deste trabalho, sugerem as etapas esperadas para o meta-

modelo de sistema com RV, contendo atividades direcionadas para todo o processo de sistemas com RV e, dessa forma, essas etapas são adotadas neste estudo;

- 4) Algumas fases ou etapas ou passos das metodologias contêm atividades que podem ser distribuídas em duas ou mais etapas no meta-modelo, enquanto outras fases ou etapas ou passos das metodologias contêm atividades que podem ser fundidas em uma única etapa no meta-modelo.

O enfoque no usuário necessário para o meta-modelo é sugerido a partir de sua fundamentação sobre a abordagem do PCU ou do DP da IHC e, conseqüentemente, sobre o modelo de processo Prototipação da ES.

A iteratividade necessária para o meta-modelo é sugerida a partir da iteratividade encontrada através das fases, passos ou etapas que compõem as metodologias para sistema com RV, determinando os ciclos inseridos através das etapas do meta-modelo para sistemas com RV proposto, assim como as atividades necessárias para cada etapa do meta-modelo para sistemas com RV proposto são sugeridas a partir da análise comparativa das atividades que compõem cada fase, passo ou etapa das metodologias para sistema com RV, determinando as atividades inseridas através das etapas do meta-modelo para sistemas com RV proposto.

Desse modo, o meta-modelo para sistemas com RV proposto por este trabalho é composto por cinco etapas: 1) Análise de Requisitos, 2) Projeto (baseado em prototipação), 3) Avaliação de Requisitos e Projeto, 4) Implementação e 5) Avaliação de Sistema.

Inicialmente, no meta-modelo proposto, são realizadas a etapa de Análise de Requisitos, de Projeto (baseado em prototipação) e de Avaliação de Requisitos e Projeto, representadas na figura 5.1.

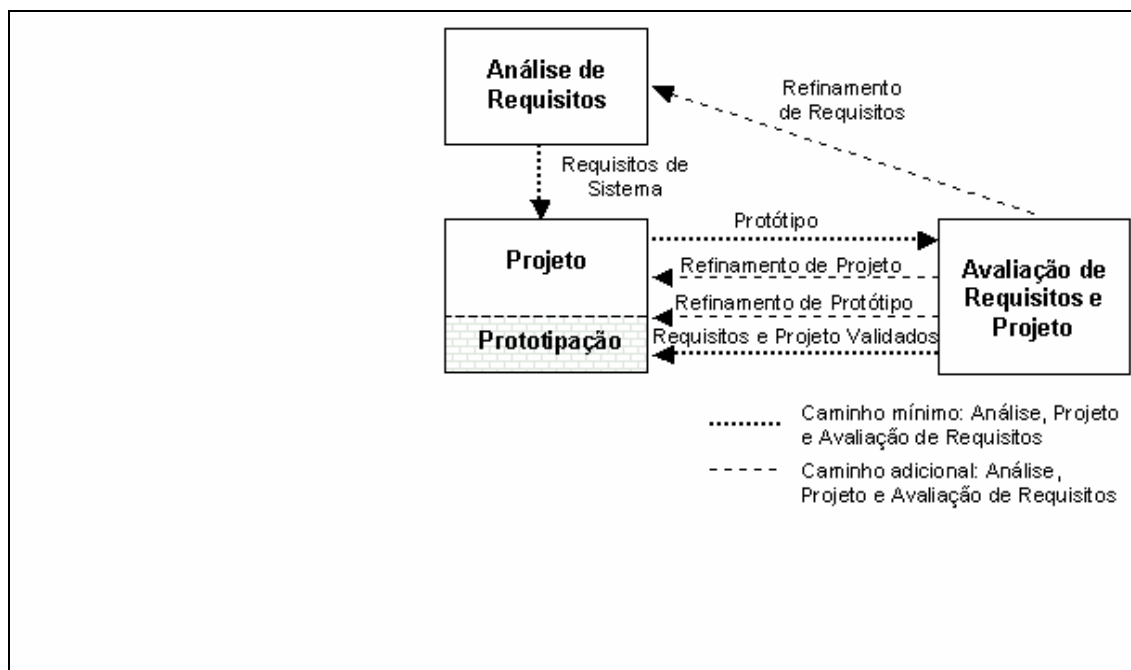


Figura 5.1 - Etapas Iniciais do Meta-modelo para Sistemas com RV com Enfoque no Usuário e Iteratividade de Projeto

Pela figura 5.1, essas três etapas iniciais formam o primeiro ciclo que permite iteratividade de projeto no meta-modelo para sistemas com RV (obtida pelos caminhos mínimo e adicional para Análise, Projeto (baseado em prototipação) e Avaliação de Requisitos e Projeto na figura 5.1). O primeiro ciclo se encerra quando os requisitos do sistema com RV e o projeto são validados de modo satisfatório por desenvolvedores através da construção do protótipo e da sua avaliação com usuários do sistema com RV e ou com profissionais da IHC especializados em avaliação, garantindo o enfoque no usuário através do PCU ou do DP (seta nomeada Requisitos e Projeto Validados na figura 5.1). Nota-se que caso a avaliação do protótipo indique problemas na definição dos requisitos, as etapas de Análise de Requisitos, Projeto (baseado em prototipação) e de Avaliação de Requisitos e Projeto devem ser realizadas novamente. Caso a avaliação do protótipo indique problemas na especificação de projeto, as etapas de Projeto (baseado em prototipação) e de Avaliação de Requisitos e Projeto devem ser realizadas novamente. E, caso a avaliação do protótipo indique problemas na implementação do protótipo, caso não esteja de acordo com os requisitos definidos ou com o projeto especificado ou caso apresente erros de execução, as etapas de Projeto (baseado em prototipação) e de Avaliação de Requisitos e Projeto devem ser realizadas novamente.

Na etapa de Análise de Requisitos devem ser realizadas as atividades descritas a seguir:

- a. Definir e delimitar o espaço do problema;
- b. Identificar os benefícios da utilização da tecnologia de RV para o espaço do problema delimitado;
- c. Identificar e analisar as pesquisas, teorias e possibilidades tecnológicas existentes para o espaço do problema;
- d. Definir os usuários, suas capacidades, as tarefas e o ambiente;
- e. Definir os requisitos funcionais, práticos e específicos do sistema com RV;
- f. Solicitar aos informantes de projeto suas visões do espaço do problema delimitado;
- g. Realizar estudos exploratórios, identificando as práticas atuais das formas de representação, dos mecanismos e dispositivos de interação e controle, da melhor combinação entre eles e as possibilidades futuras no espaço do problema;
- h. Refinar os requisitos do sistema com RV conforme o resultado da etapa de Avaliação de Requisitos e Projeto.

A etapa de Projeto (baseado em prototipação) envolve as atividades, conforme a análise resultante da etapa de Análise de Requisitos, descritas a seguir:

- a. Construir protótipos do sistema com RV, com ou sem o apoio de tecnologia computacional como em papel ou em ferramenta computacional para prototipação em RV, retratando funcionalidades e componentes de interface e de interação identificados;
- b. Descartar idéias de projeto com base no resultado da etapa de Avaliação de Requisitos e Projeto;
- c. Especificar o modelo conceitual do sistema com RV;
- d. Discutir e refinar o modelo conceitual do sistema com RV com parceiros de projeto;
- e. Projetar ou selecionar as tecnologias de entrada e saída, a arquitetura e as tecnologias computacionais;
- f. Especificar ou modelar as funcionalidades do sistema com RV;
- g. Especificar ou modelar os componentes de interface e de interação do sistema com RV;
- h. Construir protótipos do sistema com RV, com apoio de tecnologia computacional como ferramenta computacional para prototipação em RV, retratando funcionalidades e componentes de interface e de interação projetados;
- i. Refinar o projeto do sistema com RV conforme o resultado da etapa de Avaliação de Requisitos e Projeto.

Na etapa de Avaliação de Requisitos e Projeto devem ser realizadas as atividades descritas a seguir:

- Testar os protótipos do sistema com RV utilizando usuários ou profissionais de avaliação;
- Avaliar o desempenho do protótipo e do usuário;
- Avaliar a usabilidade do protótipo;
- Reportar os problemas de requisitos, projeto e ou do protótipo, determinando se há iteração a partir da etapa de Análise de Requisitos ou de Projeto (baseado em prototipação).

Finalmente, no meta-modelo proposto, são realizadas a etapa de Implementação e de Avaliação de Sistema, representadas na figura 5.2.

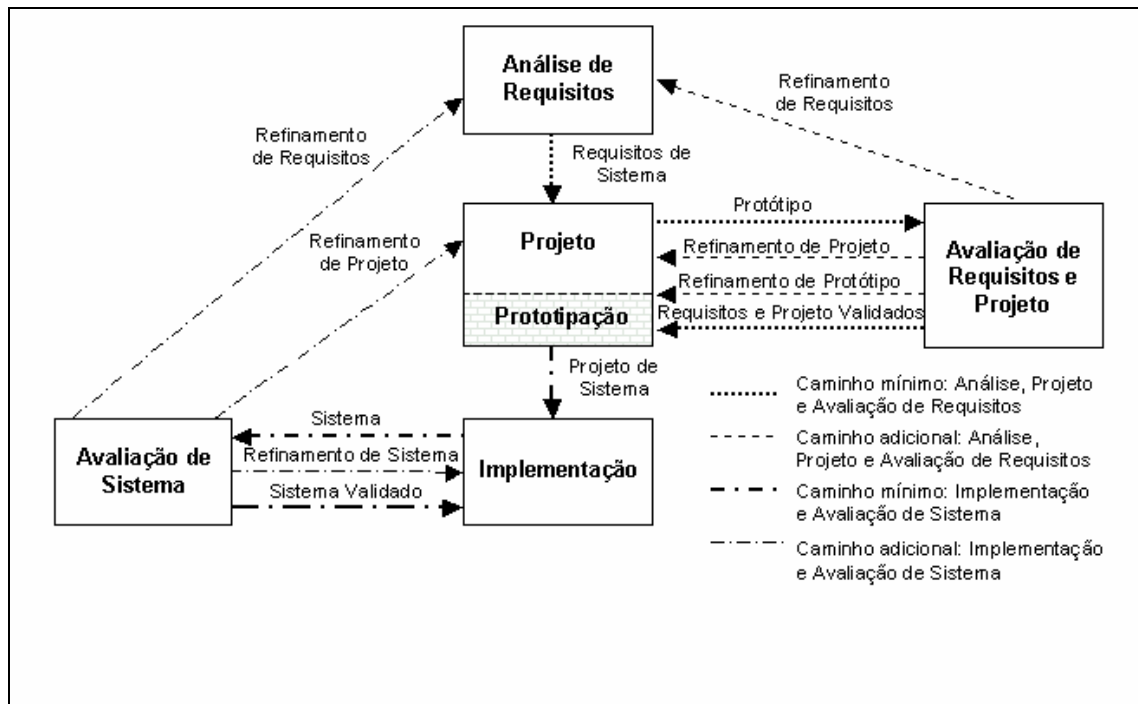


Figura 5.2 - Etapas Finais do Meta-modelo para Sistemas com RV com Enfoque no Usuário e Iteratividade de Projeto

Pela figura 5.2, essas duas etapas finais formam o segundo ciclo que também permite iteratividade de projeto no meta-modelo para sistemas com RV (obtida pelos caminhos mínimo e adicional para Implementação e Avaliação de Sistema na figura 5.2). O segundo ciclo se encerra quando o sistema é validado de modo satisfatório por desenvolvedores através da implementação do sistema e da sua avaliação com usuários do sistema com RV e ou com profissionais da IHC especializados em avaliação, garantindo o enfoque no usuário através do

PCU ou do DP (seta nomeada Sistema Validado na figura 5.2). Nota-se que caso a avaliação do sistema ainda indique problemas na definição dos requisitos ou na especificação do projeto, o primeiro ciclo, a partir, respectivamente, da etapa Análise de Requisitos ou Projeto (baseado em prototipação), deve ser realizado novamente e, posteriormente, as etapas de Implementação e Avaliação do Sistema também devem ser realizadas novamente. E, caso a avaliação do sistema indique problemas na implementação do sistema, o qual não está de acordo com os requisitos definidos ou com o projeto especificado ou apresenta erros de execução, as etapas Implementação e Avaliação do Sistema devem ser realizadas novamente.

A etapa Implementação envolve as atividades, conforme o projeto resultante da etapa de Projeto (baseado em prototipação), descritas a seguir :

- a. Implementar o projeto do sistema com RV;
- b. Corrigir o sistema com RV conforme o resultado da etapa de Avaliação de Sistema.

Na etapa de Avaliação de Sistema devem ser realizadas as atividades descritas a seguir:

- a. Testar o sistema com RV utilizando usuários ou profissionais de avaliação;
- b. Avaliar o desempenho do sistema;
- c. Avaliar o desempenho do usuário;
- d. Avaliar a usabilidade do sistema;
- e. Reportar os erros do sistema e os problemas ainda remanescentes de requisitos, projeto e ou do protótipo, determinando se há iteração a partir da etapa de Análise de Requisitos, de Projeto (baseado em prototipação) ou de Implementação.

Vale ressaltar que o meta-modelo para sistemas com RV proposto por este trabalho, mostrado na figura 5.3, se torna mais específico para o desenvolvimento desses sistemas uma vez que atividades propostas através de suas etapas (como identificação dos benefícios da utilização da tecnologia de RV considerando o espaço do problema, realização de estudos exploratórios identificando as práticas atuais de representação, mecanismos e dispositivos de interação e controle, construção de protótipos do sistema com RV retratando funcionalidades e componentes de interface e de interação e teste dos protótipos do sistema com RV utilizando usuários ou profissionais de avaliação) buscam contemplar características também mais específicas de sistemas com RV e de seu processo.

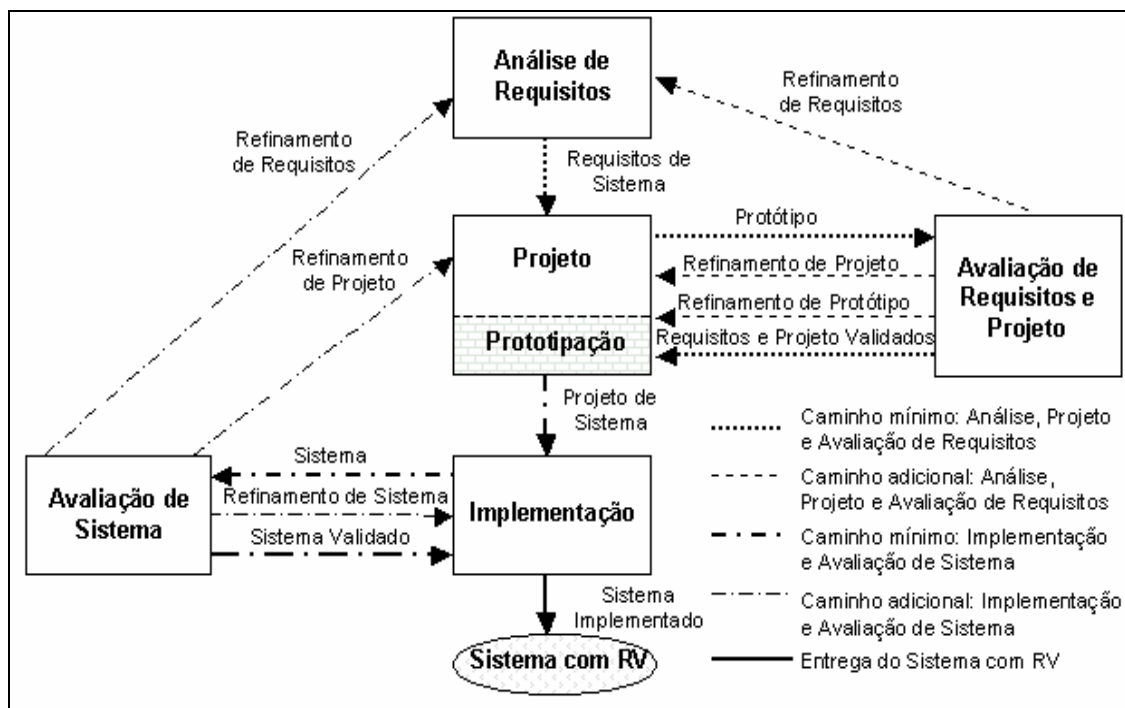


Figura 5.3 - Meta-modelo para Sistemas com RV com Enfoque no Usuário e Iteratividade de Projeto

Conforme a figura 5.3, refinamentos de requisitos e de projeto podem ser realizados no primeiro ou no segundo ciclo que permite iteratividade no meta-modelo para sistemas com RV. No entanto, esses refinamentos devem ser realizados, preferencialmente, logo no primeiro ciclo, para evitar que alterações de requisitos e ou de projeto sejam percebidas e realizadas apenas após a implementação do sistema com RV, pois se tornam mais custosas em etapas mais adiantadas de seu processo.

5.3. Meta-modelo com Enfoque no Usuário, Iteratividade de Projeto e Critérios de Usabilidade

Definidas as etapas, atividades e a iteratividade para o meta-modelo para sistemas com RV proposto, considerando o enfoque em usabilidade e no usuário e iteratividade de projeto, a hierarquia de critérios de usabilidade para sistema com RV adotada neste trabalho é incorporada a esse meta-modelo.

5.3.1. Critérios de Usabilidade Incorporados à Etapa de Análise de Requisitos

De acordo com Stanney et al (2003), o processo de interação deve ser natural, eficiente e apropriado para usuários, domínios e objetivos de tarefa do sistema com RV. Considerando-se que requisitos mais específicos, referentes ao processo de interação, podem ser elicitados na etapa de Análise de Requisitos para fornecer maior embasamento à modelagem do processo de interação do sistema com RV elaborada na etapa de Projeto (baseado em prototipação), acredita-se que se deve pensar em critérios de usabilidade de Interação na etapa de Análise de Requisitos do meta-modelo para sistemas com RV proposto.

Uma importante característica de sistemas com RV é apresentar aos usuários múltiplas entradas e saídas através de dispositivos visuais, auditivos e táteis, considerados dispositivos multimodais [Mills et al 1999] [Stanney et al 2003]. A forma de informação apresentada aos usuários deve ser prontamente entendida, não ambígua e necessária para atingir os objetivos das tarefas oferecidas pelo sistema com RV [Stanney et al 2003]. A escolha de exibição de parâmetros do sistema com RV através de seus dispositivos pode afetar significativamente sua usabilidade e o desempenho de seus usuários [Mills et al 1999]. Assumindo-se que requisitos mais específicos, referentes à exibição de parâmetros do sistema com RV e à escolha de seus dispositivos, podem ser levantados na etapa de Análise de Requisitos para melhor pautar as decisões sobre parâmetros e dispositivos na etapa de Projeto (com base com prototipação), sugere-se que se deve considerar critérios de usabilidade de Dispositivos de Entrada e Saída Multimodais na etapa de Análise de Requisitos do meta-modelo proposto por este trabalho.

Nota-se que utilizar critérios de usabilidade de Interação e de Dispositivos de Entrada e Saída Multimodais na etapa de Análise de Requisitos, a fim de garantir a usabilidade de sistemas com RV a partir de seu processo, permite que o usuário esteja envolvido não apenas desde o início desse processo, mas também que seus requisitos estejam presentes no projeto de interação e na escolha de dispositivos multimodais na etapa de Projeto (baseado em prototipação).

5.3.2. Critérios de Usabilidade Incorporados à Etapa de Projeto (Baseado em Prototipação)

A partir da etapa de Análise de Requisitos do meta-modelo para sistemas com RV proposto por este trabalho, acredita-se que se deve inserir critérios de usabilidade de Interação

e de Dispositivos de Entrada e Saída Multimodais também na etapa de Projeto (baseado em prototipação).

No que se refere aos critérios de usabilidade de Interação, o processo de interação do sistema com RV é modelado na etapa de Projeto [Seo et al 2002]. Os componentes de sistemas com RV utilizados no processo de interação (como ícones, textos, gráficos, áudios e vídeos) [Hix et al 2002] [Gabbard et al 1999] são modelados nessa etapa [Seo et al 2002]. Dessa forma, sugere-se que critérios de usabilidade de Interação devem ser considerados na etapa de Projeto (baseado em prototipação) do meta-modelo para sistemas com RV proposto.

No que diz respeito aos critérios de usabilidade de Dispositivos de Entrada e Saída Multimodais, a escolha de dispositivos para o sistema com RV é realizada na etapa de Projeto [Scaife et al 2001] [Stuart 1996]. A escolha de dispositivos e de parâmetros exibidos por esses dispositivos de sistemas com RV é de fundamental importância para sua usabilidade, uma vez que permite aos usuários interagir com esse sistema sem que haja frustração e irritação [Stanney et al 2003]. Dessa forma, acredita-se que critérios de usabilidade de Dispositivos de Entrada e Saída Multimodais devem ser pensados na etapa de Projeto (baseado em prototipação) do meta-modelo proposto por este trabalho.

Outra importante característica de sistemas com RV é o envolvimento do usuário, que é considerado de acordo com a sensação de imersão e de presença provida por esses sistemas [Sadowsky et al 2002] [Stanney et al 1998]. Para proporcionar a sensação de imersão, critérios de usabilidade podem ser considerados no projeto de software e de hardware do sistema com RV, uma vez que essa sensação é diretamente afetada por esses projetos [Stanney et al 2003]. Para facilitar a sensação de presença, critérios de usabilidade podem ser considerados ponderando-se como melhor implementar os componentes de hardware e de software do sistema com RV [Sadowsky et al 2002]. Considerando-se que o envolvimento do usuário é influenciado pelo projeto de hardware e de software de sistemas com RV, e por como melhor implementar seus componentes, sugere-se que se deve pensar em critérios de usabilidade de Envolvimento na etapa de Projeto (baseado em prototipação) do meta-modelo para sistemas com RV proposto.

Segundo Stanney et al (1998), efeitos colaterais como desconforto físico ou distúrbios fisiológicos e psicológicos associados com o uso de sistemas com RV devem ser minimizados. O ajuste e a aceitação de dispositivos usados pelos usuários no processo de interação com o sistema com RV é geralmente associado ao desconforto físico provocado por esse sistema. Considerando-se que o conforto e a segurança do usuário no uso desses dispositivos em sistemas com RV são fatores críticos para sua usabilidade [McCauley-Bell

2002], e que a escolha de dispositivos utilizados pelos usuários no processo de interação com esses sistemas é realizada na etapa de Projeto [Scaife et al 2001] [Stuart 1996], acredita-se que se deve considerar critérios de usabilidade de Efeitos Colaterais na etapa de Projeto (baseado em prototipação) do meta-modelo proposto por este trabalho.

5.3.3. Critérios de Usabilidade Incorporados às Etapas de Avaliação de Requisitos e Projeto, de Implementação e de Avaliação de Sistema

A partir das etapas de Avaliação de Requisitos e Projeto, de Implementação e de Avaliação de Sistema do meta-modelo para sistemas com RV proposto, todos os critérios de usabilidade propostos por Stanney et al (2003) estão incorporados a esse meta-modelo, uma vez que já foram inseridos ao longo das etapas de Análise de Requisitos e de Projeto (baseado em prototipação). Dessa forma, acredita-se que se deve avaliar e considerar todos os critérios de usabilidade, ou seja, de Interação, de Dispositivos de Entrada e Saída Multimodais, de Envolvimento e de Efeitos Colaterais, nas etapas de Avaliação de Requisitos e Projeto, de Implementação e de Avaliação de Sistema.

5.3.4. Interpretação e Representação do Meta-modelo com Enfoque no Usuário, Iteratividade de Projeto e Critérios de Usabilidade

Critérios de usabilidade devem ser considerados ao longo de todo o processo de sistemas com RV, ou seja, devem ser pensados nas etapas de Análise de Requisitos, de Projeto (baseado em prototipação) e de Implementação e pensados e avaliados nas etapas de Avaliação de Requisitos e Avaliação de Sistema, representados na figura 5.4.

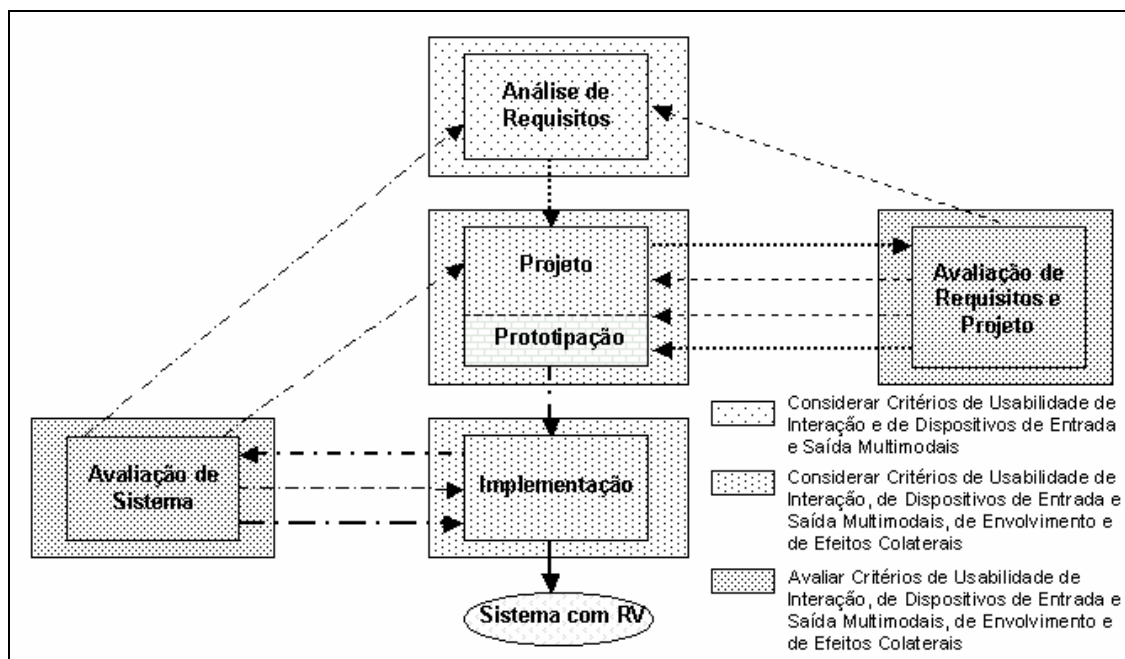


Figura 5.4 - Meta-modelo para Sistemas com RV com Enfoque no Usuário, Iteratividade de Projeto e Critérios de Usabilidade

Conforme a figura 5.4, critérios de usabilidade de Interação e de Dispositivos de Entrada e Saída Multimodais devem ser ponderados desde a etapa inicial de Análise de Requisitos até a etapa final de Avaliação de Sistema e, critérios de usabilidade de Envolvimento e Efeitos Colaterais devem ser considerados desde a etapa intermediária de Projeto até a etapa final de Avaliação de Sistema do meta-modelo para sistemas com RV proposto.

5.4. Considerações Finais

Este capítulo apresentou as contribuições das metodologias para sistemas com RV encontradas na literatura para a construção do meta-modelo para sistemas com RV e o meta-modelo para sistemas com RV.

Conforme as contribuições realizadas, nota-se que cinco etapas compõem o meta-modelo para sistemas com RV, as quais estão fundamentadas nas exigências no processo de sistemas com RV, considerando o enfoque no usuário, iteratividade de projeto e usabilidade, através da atividade de prototipação, dos ciclos e dos critérios de usabilidade inseridos ao longo da concepção, projeto e implementação de sistemas com RV. Ainda conforme as contribuições realizadas, percebe-se que todas as quatro metodologias contribuem para a

construção do meta-modelo para sistemas com RV de modo abrangente, tornando esse meta-modelo mais amplamente aceito no que se refere à ES e à IHC.

No capítulo a seguir, as conclusões sobre o meta-modelo para sistemas com RV e os demais resultados obtidos são discutidos, sendo propostos por este trabalho novos trabalhos para evoluir o meta-modelo para sistemas com RV e difundir a tecnologia de RV em sistemas interativos.

6. Conclusões

6.1. Considerações Iniciais

Conforme a pesquisa realizada neste trabalho, percebe-se que sistemas com RV possuem maior complexidade em seu processo, considerando, principalmente, os recursos de interface que devem apresentar para prover a interação mais inerente e natural aos seres humanos proposta pelo emprego da tecnologia de RV em sistemas interativos. No entanto, nota-se também, que as propostas de metodologias para sistemas com RV são poucas e, em sua maioria, não abrangentes o bastante para contemplar os princípios da ES e da IHC. Ainda conforme a pesquisa realizada neste trabalho, percebe-se a necessidade de melhor acomodar o processo de sistemas com RV nas práticas da ES e da IHC.

Desse modo, este capítulo se destina à apresentação das principais atividades de pesquisa realizadas e dos principais resultados obtidos para atingir o objetivo de melhor formalizar o processo de sistemas com RV considerando não apenas princípios da ES e da IHC, mas suas exigências nesse processo.

No item 6.2 são apresentadas as atividades de pesquisa e os resultados obtidos por este trabalho. O item 6.3 discute os desafios que devem ser suplantados para que o suporte necessário para o emprego da tecnologia de RV em sistemas interativos seja oferecido. No item 4.4 são sugeridos alguns trabalhos futuros a partir deste trabalho.

6.2. Resultados Obtidos

Para a obtenção de resultados mais abrangentes no que concerne às áreas de ES, IHC e RV, princípios da ES e da IHC foram considerados juntamente com as especificidades da tecnologia de RV e suas implicações no processo de sistemas com RV durante todas as etapas de pesquisa realizadas para o desenvolvimento do meta-modelo para sistemas com RV proposto por este trabalho, ilustrado na figura 6.1.

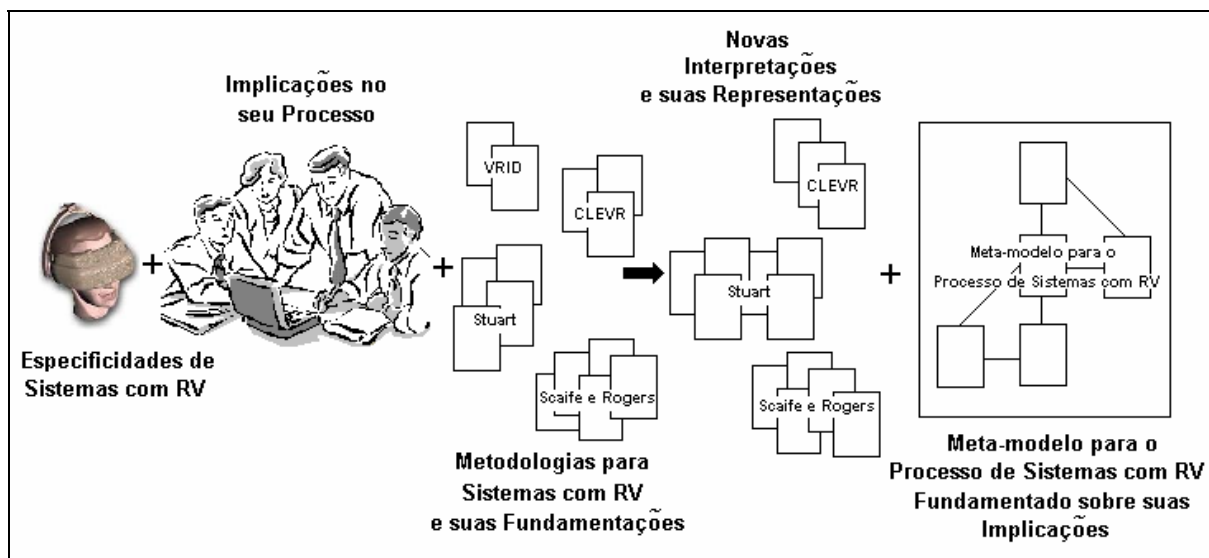


Figura 6.1 - Etapas de Pesquisa com seus Artefatos

Conforme a figura 6.1, a princípio, este trabalho se dedicou ao levantamento, à apresentação e à sintetização das especificidades encontradas em sistemas com RV, bem como à identificação e à apresentação das principais implicações existentes no processo de sistemas com RV, ou seja, enfoque no usuário, iteratividade de projeto e usabilidade, direcionando para o desenvolvimento de sistemas com RV que atendam mais amplamente aos anseios de seus usuários e mostrando as exigências fundamentais para esse processo.

Posteriormente, este estudo realizou o levantamento, a apresentação, a análise crítica e a discussão das metodologias para sistemas com RV citadas na literatura com seus proponentes.

Ainda conforme a figura 6.1, finalmente, este trabalho se dedicou à obtenção dos artefatos, propondo novas interpretações e representações baseadas nas especificidades de sistemas com RV e implicações no seu processo, as quais, confirmadas por seus proponentes, tornaram essas metodologias mais completas e abrangentes, bem como ao desenvolvimento de um meta-modelo para o processo de sistemas com RV, que abrange mais amplamente aos princípios da ES e da IHC, uma vez que está pautado por enfoque no usuário, iteratividade de projeto e critérios de usabilidade.

Para a confirmação dos resultados, alguns artigos relatando este estudo, durante seu desenvolvimento, foram submetidos a congressos e eventos de acordo com escopo deste trabalho. Dentre esses artigos, quatro foram aceitos e publicados:

- O Processo de Desenvolvimento de Sistemas com RV sob a Ótica da IHC, Congresso Latino-Americano de Interação Humano-Computador (CLIHC 2003), Artigo Completo, (Rio de Janeiro, Brasil, Agosto de 2003);

- Virtual Reality and Reengineering: Challenges for the Software Process based on a Single Vision of SE and HCI, Congresso Latino-Americano de Interação Humano-Computador (CLIHC 2003) - Integrating Human-Computer Interaction and Software Engineering Models and Processes Workshop, Workshop, (Rio de Janeiro, Brasil, Agosto de 2003);
- O Processo de Desenvolvimento de Sistemas com RV, II Congresso de Pós-Graduação da UFSCar, Painel, (São Carlos, Brasil, Setembro de 2003);
- Meta-Modelo para Processo de Sistema com RV Pautado por Enfoque no Usuário, Iteratividade de Projeto e Critérios de Usabilidade, I Workshop de Engenharia de Software (WES), Workshop, (Bauru, Brasil, Maio de 2004).

Vale ressaltar que a sintetização das especificidades, a identificação das implicações no processo, a análise crítica baseada na preocupação em contemplar a ES e a IHC, através das fundamentações das metodologias para sistemas com RV e, principalmente, a discussão dessas metodologias RV com seus autores, permitiu, além da sugestão de novas interpretações e representações para as metodologias, a construção de um meta-modelo para o processo de sistemas com RV realmente consistente e fiel ao processo exigido por esses sistemas, de modo que as necessidades de sistemas com RV possam ser atendidas e as dificuldades em seu processo possam ser superadas mais facilmente por seus analistas, projetistas e desenvolvedores.

6.3. Desafios a Suplantar

Durante todas as etapas de pesquisa realizadas para o desenvolvimento do meta-modelo para sistemas com RV proposto, o foco de contribuição deste trabalho esteve direcionado para suplantar o desafio de analisar e melhor formalizar o processo de sistemas com RV, uma vez que pesquisas nessa área ainda são pouco encontradas. No entanto, para que esse desafio possa ser completamente suplantado, deve ser realizada a experimentação prática desse meta-modelo nos mais diversos ramos do conhecimento, permitindo não apenas ratificar o meta-modelo para sistemas com RV proposto, mas também desenvolver metodologias e ou modelos de processo para sistemas com RV ainda mais específicos, bem como a concepção, projeto e implementação de sistemas com RV com qualidade, atendendo não somente às exigências da ES encontradas no processo desses sistemas, mas também às exigências da IHC.

Vale notar que além desse importante desafio, para que seja possível a exploração dos potenciais benefícios da tecnologia de RV em interfaces de sistemas interativos, inúmeros outros desafios de hardware, software e específicos para algumas atividades no processo de sistemas com RV também devem ser suplantados:

- **Hardware de RV:** os dispositivos de interface disponíveis já apresentam importantes avanços para o uso mais confortável pelos usuários de sistemas com RV. No que se refere aos dispositivos visuais como capacetes, referenciados neste trabalho como HMDs (em inglês, *Head-Mounted Displays*), problemas como tamanho, peso e fadiga decorrente de seu uso, vêm sendo solucionados através de pesquisas que tornam esses dispositivos visuais menores e mais leves. Em se tratando de dispositivos auditivos como fones de ouvido, pesquisas relacionadas a transmissões por ondas vêm buscando fazer com que esses dispositivos auditivos, sem a utilização de fios, não comprometam a movimentação do usuário. No entanto, além desses dispositivos, novos dispositivos que utilizem os sentidos do tato, olfato e paladar são importantes componentes em interfaces com RV, uma vez que permitir ao usuário tocar e sentir o mundo virtual pode ser uma experiência extremamente poderosa no que diz respeito à interação [Bowman et al 2001] [Netto et al 2002]. Desse modo, acredita-se que construir novos dispositivos, de tal forma que a usabilidade dos sistemas com RV seja garantida, se apresenta como desafio para novas pesquisas nessa área.
- **Software para RV:** a maioria das ferramentas de construção de interfaces disponíveis ainda oferecem um número bastante limitado de técnicas de interação. Segundo Willians et al (2001), alguns problemas de usabilidade associados com sistemas com RV podem ser provenientes desse número limitado de técnicas de interação oferecido, uma vez que essas técnicas podem não atender aos requisitos particulares dos mais diversos sistemas com RV. Vale lembrar que a maioria das ferramentas de construção de interfaces está voltada para projetos de interfaces WIMP, nas quais não se incluem interfaces mais avançadas como interfaces com RV, consideradas interfaces Pós-WIMP. Nesse sentido, uma ferramenta de apoio à atividade de prototipação, chamada GaCIV, é apresentada em [Silva 1999], bem como algumas de ferramentas de apoio às atividades de projeto, avaliação e refinamento de técnicas de interação em sistemas com RV, chamadas *Marigold Toolset*, são apresentados em [Willians et al 2001]. Dessa forma, acredita-se que desenvolver novas ferramentas, de tal modo que a diversidade de técnicas de

interação suportados por sistemas com RV seja garantida, se apresenta como desafio para novas pesquisas nessa área.

- **Processo de sistemas com RV:** o emprego da tecnologia de RV em sistemas interativos propicia o desenvolvimento de interfaces com aspectos de interação mais avançados, porém, a modelagem desses aspectos ainda é uma tarefa complexa e não trivial [Smith et al 1999b]. Deve-se notar que a maioria dos modelos e linguagens de descrição de interfaces com o usuário também está voltada para projetos de interfaces WIMP, nas quais não se incluem interfaces com RV, consideradas interfaces Pós-WIMP. Dessa forma, a utilização desses recursos em projetos de interface Pós-WIMP se torna inadequada, dificultando não apenas o projeto dessas interfaces, mas também a reutilização e compartilhamento de componentes de interfaces Pós-WIMP [Van Dam 1997] [Jacob et al 1999]. Com base nessas premissas, um novo modelo de sistema, uma nova linguagem para descrição e programação dos aspectos de interação em uma interface Pós-WIMP e um ambiente para a programação visual dessa linguagem, VRED (em inglês, *Virtual Reality Editor*), são apresentados em [Jacob et al 1999]. Em [Smith et al 1999b], uma discussão é realizada sobre como algumas técnicas desenvolvidas para modelagem de sistemas híbridos, compostos por componentes de interação contínua e discreta, podem ser utilizadas para representar as diferentes formas de interação em interfaces com RV. No entanto, percebe-se que inúmeras pesquisas nessa área devem ser desenvolvidas para que os modelos e linguagens de programação para o projeto de interfaces Pós-WIMP, mais especificamente de interfaces com RV, se tornem adequados o bastante às características dessas interfaces.

Esses desafios, e ainda outros não citados, evidenciam que pesquisas relacionadas à interação em sistemas com RV tendem a crescer continuamente à medida que esses sistemas se tornam mais complexos e difundidos [Bowman et al 2001].

6.4. Trabalhos Futuros

A partir do meta-modelo para o processo de sistemas com RV proposto por este trabalho, diversos trabalhos futuros poderão ser desenvolvidos para ratificar esse meta-modelo, principalmente no que se refere à sua aplicação na prática de desenvolvimento:

1. Sistemas com RV pertencentes aos mais diversos domínios de aplicação devem ser desenvolvidos com seu processo baseado no meta-modelo para sistemas com RV proposto por este trabalho, verificando se ainda são necessários ajustes nesse meta-modelo;
2. Caso sejam necessários, novos ajustes devem ser feitos para tornar o meta-modelo para sistemas com RV ainda mais completo e abrangente, tanto no que diz respeito à ES, quanto ao que se refere à IHC. Acredita-se que esses ajustes, caso sejam necessários, sejam poucos, pois observa-se que a pesquisa realizada neste estudo, identificou e ponderou incisivamente as principais implicações existentes no processo de sistemas com RV, as quais estão fortemente retratadas no meta-modelo para sistemas com RV proposto por este trabalho;
3. Ratificado o meta-modelo para sistemas com RV, no que concerne não apenas à sua validade conceitual, mas também à sua aplicação na prática de desenvolvimento, novas pesquisas devem ser feitas ponderando as especificidades encontradas em sistemas com RV de acordo com seu domínio de aplicação, verificando se são possíveis e necessárias metodologias e ou modelos de processo ainda mais específicos;
4. Caso sejam possíveis e necessárias, novas metodologias e ou novos modelos de processo específicos para determinados domínios de aplicação de sistemas com RV devem ser desenvolvidos, contemplando as especificidades encontradas nesses sistemas de acordo com seu domínio de aplicação, iniciando outros trabalhos futuros.

Vale ressaltar que pesquisadores, analistas, projetistas e desenvolvedores devem estar compromissados não apenas com a evolução do processo de sistemas com RV, mas também com a suplantação dos demais desafios que envolvem a tecnologia de RV, para que, desse modo, sistemas com RV possam ser efetivamente difundidos.

Referências Bibliográficas

[Appel et al 1999] APPEL, A. P. et al. **GACIV - A Realidade Virtual Apoiando o Desenvolvimento de Interfaces com a Participação Efetiva do Usuário**. In: XIII SBES - Simpósio Brasileiro de Engenharia de Software. out. 1999, Florianópolis. p. 1-4.

[Assis 2001] ASSIS, A. S. F. R. **Um Ambiente Computacional para Desenvolvimento de Interfaces Utilizando Realidade Virtual**. Dissertação (Mestrado) - Departamento de Computação, UFSCar - Universidade Federal de São Carlos, São Carlos, 2001.

[Assis et al 2000] ASSIS, A. S. F. R.; SILVA, J. C. A. **Development of Interactive Systems with Virtual Reality based on User Centered Design**. In: SCI' 2000 - World Multiconference on Systemics, Cybernetics and Informatics. jul. 2000, Orlando. p. 40-45.

[Bevan 1999] BEVAN, N. Quality in Use: Meeting User Needs for Quality. **Journal of Systems and Software**, v. 49, n. 1, p. 89-96, dez. 1999.

[Bowman et al 2000] BOWMAN, D. et al. **The Art and Science of 3D Interaction**. In: IEEE International Virtual Reality 2000 Conference. mar. 2000, New Brunswick. p. 18-22.

[Bowman et al 2001] BOWMAN, D. A. et al. An Introduction to 3-D User Interface Design. **Presence: Teleoperators and Virtual Environments**, v. 10, n. 1, p. 96-108, fev. 2001.

[Brown 1997] BROWN, J. HCI and Requirements Engineering - Exploring Human-Computer Interaction and Software Engineering Methodologies for the Creation of Interactive Software. **ACM SIGCHI Bulletin**, v. 29, n. 1, p. 32-35, jan. 1997.

[Burdea et al 1994] BURDEA, G.; PHILIPPE, C. **Virtual Reality Technology**. New York: John Wiley & Sons, 1994.

[Earnshaw et al 1995] EARNSHAW, R. A.; VINCE, J. A.; JONES, H. **Virtual Reality Applications**. 2. ed. Cambridge: Academic Press Limited, 1995.

[Gabbard et al 1997] GABBARD, J. L.; HIX, D. **A Taxonomy of Usability Characteristics in Virtual Environments**. Office of Naval Research (n. N00014-96-1-0385). Department of Computer Science - Virginia Polytechnic Institute and State University Blacksburg, Blacksburg, 1997.

[Gabbard et al 1999] GABBARD, J.L.; HIX, D.; SWAN, E. User-centered design and evaluation of virtual environments. **IEEE Computer Graphics and Applications**, v.19, n. 6, p. 51–59, nov. 1999.

[Gould et al 1985] GOULD, J. D.; LEWIS, C. Designing for Usability: Key Principles and What Designers Think. **Communications of the ACM**, v. 28, n. 3, p. 300-311, mar. 1985.

[Goransson et al 1999] GORANSSON B.; SANDBACK T. **Usability designers improve the user-centred design process**. In: INTERACT' 99. set. 1999, Edinburgh. p. 1-4.

[Gulliksen et al 1999] GULLIKSEN, J.; LANTS, A.; BOIVIE, I. User Centered Design - Problems and Possibilities: A Summary of the 1998 PDC & CSCW workshop. **ACM SIGCHI Bulletin**, vol. 31, n. 2, p. 25-34, apr. 1999.

[Hall 2001] HALL, R.R. Prototyping for usability of new technology. **International Journal of Human-Computer Studies**, vol. 55, n. 2, p. 485-501, ago. 2001.

[Hewett et al 1996] HEWETT, T. et al. Curricula for Human-Computer Interaction. **ACM SIGCHI Report**. Disponível em: <http://www.acm.org/sigchi/cdg>, 1996. Acesso em: 17 abr. 2004.

[Hix et al 2002] HIX, D.; GABBARD, J. L. **Usability Engineering of Virtual Environments**. In: STANNEY K. M. **Handbook of Virtual Environments: Design, Implementation, and Applications (Human Factors and Ergonomics Series)**. Mahwah: Lawrence Erlbaum Associates, Inc., 2002. p. 681-699.

[Jacob et al 1999] JACOB, R. J. K.; DELIGIANNIDIS, L.; MORRISON, S. A Software Model and Specification Language for Non-WIMP User Interfaces. **ACM Transactions on Computer-Human Interaction**, vol. 6, n. 1, p. 1-46, mar. 1999.

[Kalawsky 1999] KALAWSKY, R. S. VRUSE - A computerized diagnostic tool: for usability evaluation of virtual/synthetic environments systems. **Applied Ergonomics**, vol. 30, n. 1, p. 11-25, fev. 1999.

[Kennedy et al 1993] KENNEDY, R. S. et al. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. **International Journal of Aviation Psychology**, vol. 3, n. 3, p. 203-220, mai. 1993.

[Kim et al 1998] KIM, G. J. et al. **Software Engineering of Virtual Worlds**. In: ACM symposium on virtual reality software and technology. nov. 1998, Taipei. p. 131-138.

[Kuhn et al 1993] KUHN, S.; MULLER, M. J. Participatory Design. **Communications of the ACM**, vol. 36, n. 4, p. 25-28, jun. 1993.

[Lee et al 2002] LEE, G. A.; KIM, G. J.; PARK, C. **Modeling Virtual Object Behavior within Virtual Environment**. In: ACM symposium on virtual reality software and technology. nov. 2002, Hong Kong. p. 41-48.

[Lucena et al 1997] LUCENA, F. N.; LIESENBERG, H. K. E. Interfaces Homem-Computador: Uma Primeira Introdução. **Projeto Xchart**. Disponível em: <http://www.dcc.unicamp.br/proj-xchart/start/welcome.html>, 1997. Acesso em: 17 abr. 2004.

[McCauley-Bell 2002] McCAULEY-BELL, P. **Ergonomics in Virtual Environments**. In: STANNEY K. M. **Handbook of Virtual Environments: Design, Implementation, and Applications (Human Factors and Ergonomics Series)**. Mahwah: Lawrence Erlbaum Associates, Inc., 2002. p. 807-826.

[Mills et al 1999] MILLS, S.; NOYES, J. Virtual Reality: an overview of user-related design issues. **Interacting with Computers**, vol. 11, n. 4, p.375-386, abr. 1999.

[Murray et al 1997] MURRAY, J.; SCHELL, D.; WILLIS, C. **User Centered Design in Action: Developing an Intelligent Agent Application**. In: SIGDOC' 97 - 15th Annual International Conference on Computer Documentation. out. 1997, Salt Lake City. p. 181-188.

[Myers et al 1996] MYERS, B.; HOLLAN, J.; CRUZ, I. Strategic Directions in Human-Computer Interaction. **ACM Computing Surveys**, vol. 28, n. 4, p. 794-809, dez. 1996.

[Netto et al 2002] NETTO, A. V.; MACHADO, L. S.; OLIVEIRA, M. C. F. **Realidade Virtual: Fundamentos e Aplicações**. Florianópolis: Visual Books Editora, 2002.

[Neale et al 2001] NEALE, H.; NICHOLS, S. Theme-based content analysis: a flexible method for virtual environment evaluation. **International Journal of Human-Computer Studies**, vol.55, n. 2, p. 167-189, ago. 2001.

[Nielsen 1993] NIELSEN, J. **Usability Engineering**. Cambridge: Academic Press Limited, 1993.

[Olher et al 2003] OLHER, M. M.; SILVA J. C. A. **O Processo de Desenvolvimento de Sistemas com RV sob a Ótica da IHC**. In: CLIHC 2003 - Congresso Latino-Americano de Interação Humano-Computador. ago. 2003, Rio de Janeiro. p. 79-89.

[Olher et al 2004] OLHER, M. M.; SILVA J. C. A. **Meta-Modelo para Processo de Sistema com RV Pautado por Enfoque no Usuário, Iteratividade de Projeto e Critérios de Usabilidade**. In: WES 2004 - Workshop de Engenharia de Software. mai. 2004, Bauru.

[Pereira et al 2000] PEREIRA, A.; TISSIANI, G.; REBELO, I. B. **Design de Interfaces para Ambientes Virtuais: como obter usabilidade em 3D**. In: SIGRaDi'2000 - IV Sociedad Iberoamericana de Gráfica Digital. set. 2000, Rio de Janeiro. p. 25-28.

[Pimentel et al 1995] PIMENTEL, K.; TEIXEIRA, K. **Virtual Reality: Through the New Looking Glass**. 2. ed. New York: McGraw-Hill, 1995.

[Pinho et al 1997] PINHO, M. S.; KIRNER, C. **Uma Introdução à Realidade Virtual**. In SIBGRAPI'97 - X Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens. out. 1997, Campos do Jordão.

[Preece 1993] PREECE, J. **A Guide to Usability: Human-Computer Interaction**. Cambridge: Addison-Wesley, 1993.

[Pressman 2001] PRESSMAN, R. **Software Engineering: A Practitioner's Approach**. 5. ed. New York: McGraw-Hill, 2001.

[Rocha et al 2003] ROCHA, H. V.; BARANAUSKAS, M. C. C. **Design e Avaliação de Interfaces Humano-Computador**. Campinas : Emopi Editora e Gráfica, 2003.

[Sadowsky et al 2002] SADOWSKY, W.; STANNEY, K.M. **Measuring and Managing Presence in Virtual Environments**. In: STANNEY K. M. **Handbook of Virtual Environments: Design, Implementation, and Applications (Human Factors and Ergonomics Series)**. Mahwah: Lawrence Erlbaum Associates, Inc., 2002. p. 791-806.

[Scaife et al 2001] SCAIFE, M.; ROGERS, Y. Informing the design of a virtual environment to support learning in children. **International Journal of Human-Computer Studies**, vol. 55, n. 2, p. 115-143, ago. 2001.

[Seo et al 2002] SEO, J.; KIM, G. J. Design for Presence: A Structured Approach to Virtual Reality System Design. **Presence: Teleoperators and Virtual Environments**, vol. 11, n. 4, p. 378-403, ago. 2002.

[Silva 1999] SILVA, J. C. A. **Development of Virtual Interfaces Using Configurable Templates**. In: ICCIMA' 99 - 3rd International Conference On Computational Intelligence And Multimedia Applications, set. 1999, New Delhi. p. 354-358.

[Smith et al 1999a] SMITH, S.; DUKE, D.; MASSINK, M. The hybrid world of virtual environments. **Computer Graphics Forum**, vol. 18, n. 3, p. C297-C307, set. 1999.

[Smith et al 1999b] SMITH, S.; DUKE, D. **Virtual Environments as Hybrid Systems**. In: EG-UK'99 - Eurographics UK 17th Annual Conference, set. 1999, Cambridge. p. 113-128.

[Smith et al 2001] SMITH, S. P.; HARRISON, M. D. Editorial: User centred design and implementation of virtual environments. **International Journal of Human-Computer Studies**, vol. 55, n. 2, p. 109-114, ago. 2001.

[Shneiderman 1998] SHNEIDERMAN, B. **Designing the User Interface: Strategies for Effective Human-Computer Interaction**. 3. ed. Cambridge: Addison-Wesley, 1998.

[Stanney et al 1998] STANNEY, K. M. et al. Aftereffects and sense of presence in virtual environments: formulation of a research and development agenda. **International Journal of Human-Computer Interaction**, vol. 10, n. 2, p. 135-187, mar. 1998.

[Stanney et al 2003] STANNEY, K. M. et al. Usability engineering of virtual environments (VEs): identifying multiple criteria that drive effective VE system design. **International Journal of Human-Computer Studies**, vol. 58, n. 4, p. 447-481, abr. 2003.

[Stuart 1996] STUART, R. **Design of Virtual Environments**. New York: McGraw-Hill, 1996.

[Tanriverdi et al 2001] TANRIVERDI, V.; JACOB, R. J. K. **A Design Model and Methodology for Developing Virtual Reality Interfaces**. In: ACM symposium on Virtual reality software and technology, nov. 2001, Banff, p. 175-182

[Van Dam 1997] Van DAM, A. Post-WIMP User Interfaces. **Communications of the ACM**, vol. 40, n. 2, p. 63-67, fev. 1997.

[Vince 1995] VINCE, J. **Virtual Reality Systems**. Cambridge: Addison-Wesley, 1995.

[Willians et al 2001] WILLIAMS, J. S.; HARRISON, M. D. A toolset supported approach for designing and testing virtual environment interaction techniques. **International Journal of Human-Computer Studies**, vol. 55, n. 2, p. 146-165, ago. 2001.

[Witmer et al 1993] WITMER, B.; SINGER, M. Measuring presence in virtual environment: a presence questionnaire. **Presence: Teleoperators and Virtual Environments**, vol. 7, n. 3, p. 225-240, apr. 1993.

Apêndice A

Contexto em Realidade Virtual (RV)

1. Dentre termos como (a) sistema com RV, (b) software de RV, (c) hardware de RV, (c) aplicação de RV, (d) ambiente virtual e (e) interface com o usuário com RV ou somente interface com RV, como o principal termo utilizado em sua abordagem é entendido no contexto dessa abordagem e quais as suas definições para os termos mencionados.
2. Como sua abordagem é classificada: (a) Modelo de ciclo de vida, (b) Metodologia, (c) Método e ou (d) outro e quais as suas definições para os termos mencionados.
3. Entre fases genéricas como (a) Definição (análise de sistema, planejamento do projeto de software, análise de requisitos), (b) Desenvolvimento (projeto de software, codificação, realização de testes de software) e (c) Manutenção (correção, adaptação, melhoramento funcional) para o processo de desenvolvimento de softwares comuns, quais fases são previstas por sua abordagem e quais possíveis correspondências entre essas fases e as fases, etapas e ou atividades que compõem essa abordagem.

Engenharia de Software (ES)


4. Como sua abordagem está fundamentada sobre modelos de ciclo de vida da ES como (a) Modelo de ciclo de vida Clássico, (b) Prototipação e (c) Espiral e sobre princípios da ES, considerando a necessidade de uma estrutura iterativa para o processo de desenvolvimento de sistemas de software interativos.

Interação Humano-Computador (IHC)

5. Como sua abordagem está fundamentada sobre abordagens da IHC como (a) Projeto Centrado no Usuário (PCU), (b) Design Participativo (DP) e (c) Modelo Cognitivo e sobre princípios da IHC, considerando o foco sobre o usuário e a atividade de prototipação no processo de desenvolvimento de sistemas de software interativos.

Anexo A

E-mail com a resposta dos autores da metodologia VRID para o questionário específico para essa metodologia, elaborado e enviado por esta pesquisadora.

Data:	Sexta, Maio 30 2003 06:35 pm
De:	Vildan Tanriverdi <vildan@us.ibm.com> 
Para:	milena@dc.ufscar.br
Responder Para:	Vildan Tanriverdi <vildan@us.ibm.com>
Assunto:	Re: About VRID (Questionnaire)
Cabeçalho Completo:	Mostrar Cabeçalho

Please see my responses below.

Regards,
Vildan Tanriverdi

VRID (Virtual Reality Interface Design)

Context in Virtual Reality (VR)

1. Among terms like (a) VR system, (b) VR software, (c) VR hardware, (c) VR application, (d) virtual environment and (e) VR user interface or just VR interface, VR interface is the principal term used in your VRID approach presented in your article "VRID: A Design Model and Methodology for Developing Virtual Reality Interfaces", (Tanriverdi, V.; Jacob, R. J. K..In Proceedings of the ACM symposium on Virtual reality software and technology, November 2001, 175-182). How is the term VR interface understood in the context of this approach? Please, comment about your definitions for these mentioned terms.

In Section 3 : "Components of a Virtual Reality System" of the paper, we defined the components of the VR system and explained the "VR interface" in that context. It is the front-end component of the system that lets users and other external entities exchange information with and manipulate the system. The reason for frequent use of "VR interface" term in the paper is that our research focuses on the VR interface component of a VR system.

2. How is VRID approach classified: (a) Model of life cycle, (b) Methodology, (c) Method and or (d) other? If other, which? Please, comment about your definitions for these mentioned terms.

We proposed the VRID as a model and methodology for designing VR interfaces. In the model, we described the architecture of the objects that are part of the VR interface and their relationship with each other and data elements. In the methodology, we describe the phases and steps that need to followed in each phase to systematically apply the model.

3. Among generic phases as (a) Definition (analysis of the system, planning of the software project, analysis of requirements), (b) Development (software project, code, accomplishment of tests of the software) and (c) Maintenance (correction, adaptation, functional improvement) foreseen for

the development process of common softwares, which are foreseen by VRID approach for the development process of VR interfaces? Is it possible to accomplish some kind of correspondence among the mentioned generic phases and the phases and or steps that compose VRID approach? If yes, which mentioned generic phases and phases and or steps that compose this approach are corresponding?

Among the major software life cycle components (Analysis, Design, Development and Maintenance), the VRID model and methodology focuses on the design of a VR software, specifically, the interface part of the software. The low level design specifications produced at the low level design phase of the methodology aim to help software developers in the development.

4. How are the need of new iterations and refinements determined between the phases and levels of details foreseen by the VRID approach?

Evaluation of the design e.g. heuristic evaluation and communication with software developers may determine the need for iterations and refinements.

5. According to some citations contained in the article "VRID: A Design Model and Methodology for Developing Virtual Reality Interfaces", (a) the Designer of VR interfaces and (b) the software developer work in the development process of VR systems. However, the development process of VR softwares need to be considered by a single vision of Software Engineering and Human- Computer Interaction. How are these professionals considered in the context of the VRID approach? Please, comment about this regard.

The designer of a VR interface works on the design aspects of the interface and produces the specifications ready to be implemented by the software developer. The software developer uses these specifications in implementing the VR software. It is possible that one person can take both the designer and the software developer role. If these roles are taken by different people, the design specifications provide a good communication mechanism between the designer and the software developer.

Software Engineering (SE)

6. In agreement with the study accomplished through the article "VRID: A Design Model and Methodology for Developing Virtual Reality Interfaces", The VRID approach is not founded on the models of life cycle as the Classic Life Cycle, Prototyping and Spiral Model of SE. There are not citations in this article about this consideration. Please, comment about how VRID approach is based on principles of SE.

We haven't explicitly specified the VR software life cycle in the paper. However, we emphasized that VRID model and methodology focus on the design of interface component of a VR system and requires an iterative process. We assume the following major phases for a VR software: Analysis, Design, Implementation and Maintenance.

Human-Computer Interaction (HCI)

7. Still in agreement with the study accomplished through the article "VRID: A Design Model and Methodology for Developing Virtual Reality Interfaces", the VRID approach is not founded on approaches of HCI as User Centered Design (UCD), Participatory Design (PD) and Cognitive Modeling. There are not citations in this article about this consideration. However,

the focus on the user has been showing if as important requirement for the development of approaches that support the development process of VR softwares. How does the VRID approach consider the focus on the user during the development process of VR interfaces? Please, comment how this approach is based on principles of HCI.


In this study, our first goal was to develop and obtain proof-of-concept for the VRID model and methodology and we assumed that designers received preprepared descriptions of design task. When end users are involved in the design process, it is important for designers to use the VRID in conjunction with user centered design and evaluation methodologies. Also, the low level design specifications produced at the low level design phase of the methodology aim to enhance communication and collaboration among users, designers and software developers and fits into the principles of participatory design.

8. In agreement with the study accomplished through the article "VRID: A Design Model and Methodology for Developing Virtual Reality Interfaces", The VRID approach doesn't present users' evaluations through the prototyping. However, due to the necessary integration between the VR softwares and its users, the prototyping has been showing if important requirement for the development of approaches that support the development process of VR softwares. How does the VRID approach consider the prototyping during the development process of VR interfaces?

As I explained in the previous question, users were not involved in this study. However, the languages, frameworks and models that we propose to use in producing the design specifications may involve/encourage prototyping.

Anexo B

E-mail com a resposta dos autores da metodologia CLEVR para o primeiro questionário específico para essa metodologia, elaborado e enviado por esta pesquisadora.

Data: Sábado, Junho 7 2003 02:47 am
De: Jinseok Seo <jsseo@postech.ac.kr> 
Para: milena@dc.ufscar.br
Responder Para: Jinseok Seo <jsseo@postech.ac.kr>
Assunto: Re: About CLEVR (Questionnaire) (2)
Cabeçalho Completo: Mostrar Cabeçalho

I hope my late reply helps.

> CLEVR (Concurrent and Level by Level Development of VR System)
>
>
> Context in Virtual Reality (VR)
>
> 1. Among terms like (a) VR system, (b) VR software, (c) VR hardware, (c)
> VR application, (d) virtual environment and (e) VR user interface or
> just VR interface, VR system is the principal term used in your CLEVR
> approach presented in your article "Design for Presence: A Structured
> Approach to Virtual Reality System Design" (Seo, J.; Kim, G. J. In
> Presence: Teleoperators and Virtual Environments, vol. 11, no. 4, August
> 2002, 378- 403). How is the term VR system understood in the context of
> this approach? Please, comment about your definitions for these
> mentioned terms.
>
I don't know I can dare to define the terms you mentioned.
Even when I was writing my paper, I was confusing the exact meanings of the
phrases that I had written.
Anyway, I write my own thoughts on the terms.
(a) VR system: VR HW + VR SW + (content?)
(b) VR software: == VR application
(c) VR hardware: trackers, HMD, ...
(d) VR application: == VR software
(e) VR user interface: it has two meanings. 1) physical interfaces like
trackers, HMD, ... 2) logical interfaces or metaphorical symbols for user
interaction, like 3D menu, 3D widgets, ...
>
>
> 2. How is CLEVR approach classified: (a) Model of life cycle, (b)
> Methodology, (c) Method and or (d) other? If other, which? Please,
> comment about your definitions for these mentioned terms.
(b) Methodology. (I hope ^^)
>
>
> 3. Among generic phases as (a) Definition (analysis of the system,
> planning of the software project, analysis of requirements), (b)
> Development > (software project, code, accomplishment of tests of the
> software) and (c) Maintenance (correction, adaptation, functional
> improvement) foreseen for the development process of common softwares,
> which are foreseen by CLEVR approach for the development process of VR
> systems?

all the phases.

> Is it possible to accomplish some kind of
> correspondence among the mentioned generic phases and the stages and or
> iterations that compose CLEVR approach? If yes, which mentioned generic
> phases and stages and or iterations that compose this approach are
> corresponding?

In this case, some terms have ambiguities.

(a) Definition -> Requirements

(b) Development -> Design

(c) Validation -> Maintenance

>

>

> Software Engineering (SE)

>

> 4. According to some citations contained in the article "Design for
> Presence: A Structured Approach to Virtual Reality System Design", the
> CLEVR approach is founded on the model of life cycle Spiral of SE.
> According to other citations contained in this same article, the CLEVR
> approach is also funded on incremental and hierarchical development.
> However, some characteristic stages and activities of Spiral paradigm of
> SE as analysis of risks are not clearly accomplished during the
> development process of VR systems foreseen by CLEVR approach. How the
> CLEVR approach is founded on the model of life cycle Spiral of SE and
> incremental and hierarchical development through its stages and
> activities? Please, comment how CLEVR approach is based on principles of
> SE.

It is true that CLEVR approach is based on some principles of SE.

But, it is too difficult to match CLEVR to the specific SE principles.

>

>

> 5. According to the article "Design for Presence: A Structured Approach
> to Virtual Reality System Design", "requirements-design-validation"
> iterations are accomplished continually in each stage foreseen by the
> CLEVR approach. However, citations in the article not explain clearly if
> (1) "requirements-design-validation" iterations happen in the first
> stage, iterations of "requirements-design-validation" happen in the
> second stage and "requirements-design-validation" iterations happen in
> the third stage or if (2) "requirements-design-validation" iterations
> happen along the three stages as a completely. How are these iterations
> and stages accomplished in the development process of VR systems foreseen
> by the CLEVR approach?

(1)

The numbers and boundaries of stages and phases are not so important.
the division of states and phases were only for explanation.

>

>

> Human-Computer Interaction (HCI)

>

> 6. In agreement with the study accomplished through the article "Design
> for Presence: A Structured Approach to Virtual Reality System Design",
> the CLEVR approach is not founded on the approaches as User Centered
> Design (UCD), Participatory Design (PD) and Cognitive Modeling of HCI.
> There are not citations in this article about this consideration.
> However, the focus on the user has been showing if as important
> requirement for the development of approaches that support the
> development process of VR softwares. How does the CLEVR approach consider
> the focus on the user during the development process of VR systems?
> Please, comment how this approach is based on principles of HCI.


I have not considered the elements about HCI yet.

I think It might be the future work of my graduation paper.

>
>
> 7. Still in agreement with the study accomplished through the article
> "Design for Presence: A Structured Approach to Virtual Reality System
> Design", the CLEVR approach doesn't present users' evaluations through
> the prototyping. However, due to the necessary integration between the VR
> softwares and its users, the prototyping has been showing if important
> requirement for the development of approaches that support the
> development process of VR softwares. How does the CLEVR approach consider
> the prototyping during the development process of VR systems?
In CLEVR, there is no difference between prototyping and development.
>
>
> Support Tools
>
> 8. The article "Design for Presence: A Structured Approach to Virtual
> Reality System Design" presents a group of tools "P-VoT" (P? - Virtual
> reality system development Tool) that support the development process of
> VR systems. Are these tools available for VR systems developers and
> searchers? If yes, how? Please, confirm the name given to this group of
> tools.
The old version was highly dependent on 3rd party APIs and tools from other
lab in my universiy and other 2 universities in Korea. Among them, WTMO
and ASADAL played an important roles in the simulation of the Statecharts.
And, it had not assumed the form of the integrated package.
But, I hope that the new version (has completely different form and let me
graduate) would be available at the end of 2003. The current version is
only dependent on OpenGL
Performer and CAVELib.

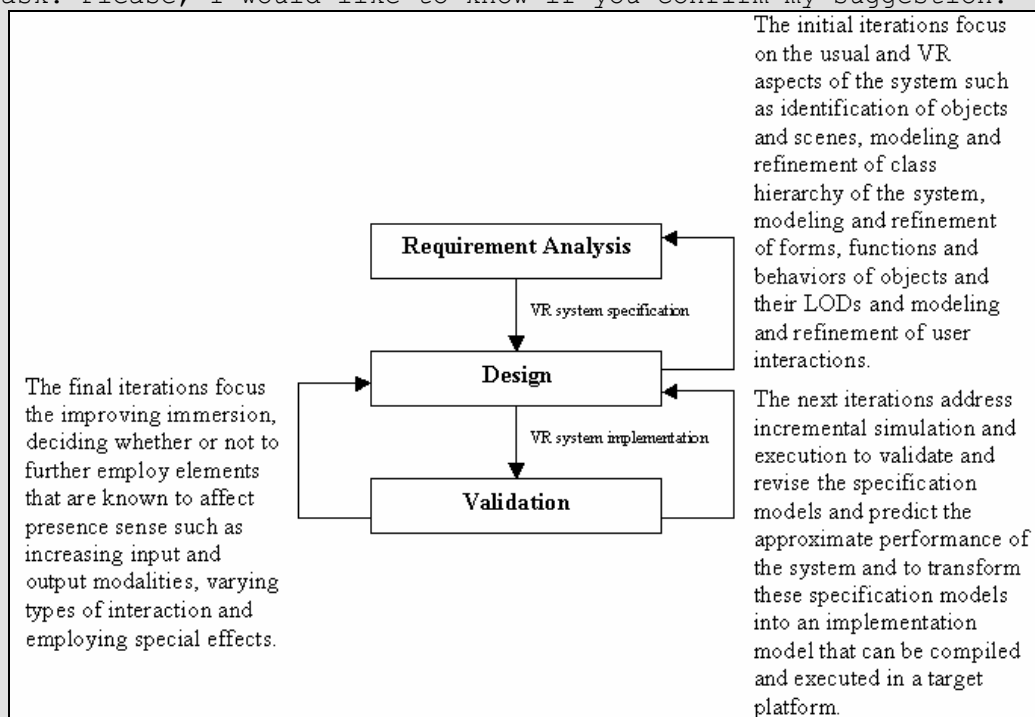
Jinseok Seo
VR lab, Dept of CSE
POSTECH

E-mail com a resposta dos autores da metodologia CLEVR para o segundo questionário específico para essa metodologia, elaborado e enviado por esta pesquisadora com base na resposta do primeiro questionário específico para a metodologia CLEVR.

Data: Quinta, Julho 10 2003 10:13 pm
De: Jinseok Seo <jsseo@postech.ac.kr> 
Para: MilenaMarquezinOlher <milena@dc.ufscar.br>
Responder Para: Jinseok Seo <jsseo@postech.ac.kr>
Assunto: Re: About CLEVR (Confirmation)
Cabeçalho Completo: Mostrar Cabeçalho

> As I sent you and you answered me:
>
> A)
> 3. Among generic phases as (a) Definition (analysis of the system,
> planning of the software project, analysis of requirements), (b)
> Development (software project, code, accomplishment of tests of the
> software) and (c) Maintenance (correction, adaptation, functional
> improvement) foreseen for the development process of common softwares,
> which are foreseen by CLEVR approach for the development process of VR
> systems?
> all the phases.
> Is it possible to accomplish some kind of correspondence among the
> mentioned generic phases and the stages and or iterations that compose
> CLEVR approach? If yes, which mentioned generic phases and stages and or
> iterations that compose this approach are corresponding?
> In this case, some terms have ambiguities.
> (a) Definition -> Requirements
> (b) Development -> Design
> (c) Validation -> Maintenance
>
> My comments: I agree with you. In this case, some terms have ambiguities.
> But I believe that the Maintenance phase presented by Pressman occurs in
> terms of an existing system and your Validation iteration occurs in terms
> of an system in development.
> Studying your article "A Structured Approach to Virtual Reality System
> Design", I believe that your approach involve Requirements (my "a" -
> Definition) and Design and Validation (my "b" - Development).
>
> My ask: Please, I would just like to know if you agree with me.
I agree with you.
>
>
> B)
> 5. According to the article "Design for Presence: A Structured Approach
> to Virtual Reality System Design", "requirements-design-validation"
> iterations are accomplished continually in each stage foreseen by the
> CLEVR approach. However, citations in the article not explain clearly if
> (1) "requirements-design-validation" iterations happen in the first
> stage, iterations of "requirements-design-validation" happen in the
> second stage and "requirements-design-validation" iterations happen in
> the third stage or if (2) "requirements-design-validation" iterations
> happen along the three stages as a completely. How are these iterations
> and stages accomplished in the development process of VR systems foreseen
> by the CLEVR approach?
> (1)

> The numbers and boundaries of stages and phases are not so important.
 > the division of states and phases were only for explanation.
 >
 > My comments: Again, studying your article "A Structured Approach to
 > Virtual Reality System Design", I came to the conclusion that your
 > approach could be represented in a iterative structure of development
 > process, as showed in the annexed picture. In this iterative structure
 > some cycles allow to eliminate the necessity to possess a complete set of
 > requirement specifications for a system before the start of its
 > implementation, allowing many changes to be applied afterwards in a more
 > simple fashion (which you call "incremental development") and allow to
 > focus on the more critical features in form, function and behavior in a
 > top-down manner (which you call "hierarchical development") and bottom-up
 > manner.
 >
 > My ask: Please, I would like to know if you confirm my suggestion.




I think your suggestion would be very close to what I've stated.
 >
 >
 > C)
 > 8. The article "Design for Presence: A Structured Approach to Virtual
 > Reality System Design" presents a group of tools "P-VoT" (P? - Virtual
 > reality system development Tool) that support the development process of
 > VR systems. Are these tools available for VR systems developers and
 > searchers? If yes, how? Please, confirm the name given to this group of
 > tools.
 > The old version was highly dependent on 3rd party APIs and tools from
 > other lab in my universiy and other 2 universities in Korea. Among them,
 > WTMO and ASADAL played an important roles in the simulation of the
 > Statecharts. And, it had not assumed the form of the integrated package.
 > But, I hope that the new version (has completely different form and let
 > me graduate) would be available at the end of 2003. The current version
 > is only dependent on OpenGL Performer and CAVELib.
 >
 > My comments: Studying your article, I came to the conclusion that the
 > version called "P-VoT" (P? - Virtual reality system development Tool)
 > presents tools that allow 1) to construct separately the form, function
 > and behavior specification models (supporting the design specification),

```
> 2) to simulate these specification models (supporting the design
> validation), and 3) to transform these specification models into an
> implementation model that can be compiled and executed in a target
> platform (supporting the design implementation).
>
> My ask: Please, I would just like to know if you confirm my conclusion.
You're right.
>
>
```

```
-----
Jinseok Seo
VR lab, Dept of CSE
POSTECH
```

Anexo C

E-mail com a resposta do autor da metodologia de Stuart para o primeiro questionário específico para essa metodologia, elaborado e enviado por esta pesquisadora.

Data: Domingo, Junho 29 2003 02:06 am
De: Rory Stuart <rory@rorystuart.com> 
Para: milena@dc.ufscar.br
Responder Para: Rory Stuart <rory@rorystuart.com>
Assunto: Re: About VR Systems Approach (Questionnaire)
Cabeçalho Completo: Mostrar Cabeçalho

Hello Milena,
> -----
> Approach of Stuart
>
>
> Context in Virtual Reality (VR)
>
> 1. Among terms like (a) VR system, (b) VR software, (c) VR hardware, (c)
> VR application, (d) virtual environment and (e) VR user interface or
> just VR interface, virtual environment is the principal term used in
> your approach presented in your book "The Design of Virtual
> Environments" (Stuart, R. McGraw-Hill, 1996.). How is the term virtual
> environment understood in the context of this approach? Please, comment
> about your definitions for these mentioned terms.

I hope you're okay with my replying rather informally, to convey the understanding to you -- I'm not worried here about writing things with a wording that would be appropriate to formal definitions -- if you need that, please let me know, and I can give you feedback on your formal definitions...

VR system suggests to me the entire system -- the hardware, software, etc... -- with perhaps an emphasis or focus on "system-level" features, such as system architecture (I compare a couple of examples in my book with diagrams contrasting coupled and decoupled simulation architecture -- that's what I mean by "system-level" focus).

VR software and VR hardware are simply the software and hardware components of the system, respectively. In fact, people would tend to use these terms especially about those parts of the hardware and software that are unique to VR ... so, although a general-purpose unix workstation might be a hardware part of the system, people often think of the unique input/output devices used in VR systems (HMDs, instrumented gloves, etc...) when they speak of "VR hardware."

The virtual environment is the interactive simulation presented to the user (more completely defined in my book) -- it is in essence the experience presented to the user. This is almost the same as the VR user interface; perhaps "VR user interface" puts extra focus on how the user interacts with the system (e.g. how the user is able to navigate, moving down the aisles through a simulated aircraft, for example), while virtual environment puts extra focus on the environment itself (in this example, the simulated

aircraft and what sensory experiences it offers the user). This is just my personal take on why someone might switch between these two terms, but, as I said, they are pretty much interchangeable.

> 2. How is your approach classified: (a) Model of life cycle, (b)
> Methodology, (c) Method and or (d) other? If other, which? Please,
> comment about your definitions for these mentioned terms.

It is iterative design, a relatively common hci (human computer interaction) approach, applied to virtual environments. It could be called a design methodology. "Model of life cycle" I think is an expression used more by computer scientists, e.g. in discussing object-oriented design, I have not heard it used as much by people with hci focus who are looking at the user-oriented perspective, and the bigger picture of users/tasks/environments (discussed in one section of the book).

> 3. Among generic phases as (a) Definition (analysis of the system,
> planning of the software project, analysis of requirements), (b)
> Development (software project, code, accomplishment of tests of the
> software) and (c) Maintenance (correction, adaptation, functional
> improvement) foreseen for the development process of common softwares,
> which are foreseen by your approach for the development process of
> virtual environments? Is it possible to accomplish some kind of
> correspondence among the mentioned generic phases and the steps and or
> activities that compose your approach? If yes, which mentioned generic
> phases and steps and or activities that compose your approach are
> corresponding?

The approach I'm describing involves defining requirements (your "a"), followed by prototyping, evaluation, and then iteration: Have the users try to accomplish specified tasks using the prototype, observe strengths and weaknesses of the design (everything from problems with performance of the prototype to problems the users have in figuring out how to accomplish the tasks or interact with the system), then revise the design & prototype, and do further rounds of human factors - style evaluation of users attempting to accomplish tasks until a satisfactorily successful design has been validated. At this point, implement this design in a version to be deployed (which may involve quite a bit more than did the prototype).

How does this differ in a practical way from the more traditional phases you describe in your question? The big difference is that, with that traditional approach, the team tends to decide on a final design and begin coding the version of the application that is to be deployed (no rapid prototyping). If they do *any* testing with users, it tends to happen so late in the process that it can't significantly alter the design (there is so much already invested in the design choices that were made at the beginning); often this kind of testing is a sort of superficial exercise to try to prove that they've done something useful and "care about the user" but it doesn't take advantage of all of the important insights gained from real users (trying to accomplish real tasks with a prototype) that can cause radical alterations in design and result in a much more useful and usable final system.

I suggest, if this is unfamiliar, you look at some of the conference Proceedings of organizations such as ACM-CHI and Human Factors Society, for papers on rapid prototyping and evaluation -- there have been many diverse

sorts of user-computer interfaces designed using this technique.

> Software Engineering (SE)

>

> 4. In agreement with the study accomplished through the book "The Design
> of Virtual Environments", your approach is founded on the model of life
> cycle Prototyping of SE. However, there are not citations in this book
> about this consideration. Please, comment about how your approach is
> based on principles of SE.

Try typing in "iterative design" "rapid prototyping" to a search engine
such as Google, and you will find many examples of the use of this
approach, as described above.

> 5. Still in agreement with the study accomplished through the book "The
> Design of Virtual Environments", your approach presents including steps
> in regard to VR technology, that they describe in a detailed way, among
> others, technological characteristics of available input and output
> devices. However, for the development process of virtual environments,
> more specifically for the development process of VR software, in high
> level of abstraction, some of these descriptions could be suppressed, so
> that only its implications for virtual environments were portrayed in
> this development process. Please, comment about your opinion about this
> suggestion.

If I understand what you're asking, it is along the lines of "could we
plan some general things about the VE system we want to design before
we worry about the specific input and output devices a user would use
to interact with the system?". Yes, I think you certainly could.
You might start to plan a system for training pilots to use a new
type of helicopter, for example. You might hypothesize that it would be
useful for the pilot to be able to practice doing certain maneuvers as
though they were in the cockpit of the helicopter, and view and hear
things from that perspective; but that it would also be useful for
them to be able to "replay" the maneuver they just did, and observe it
from a fixed point in space, as a sort of 3rd party observer. You
could think about a lot of these sorts of issues before you ever worried
about whether the system you design will use particular kinds of motion
platforms, visual displays, real or virtual cockpit controls, etc...

Of course, in order to even begin to do a first prototype, you will
need to make certain decisions about input and output devices,
interaction styles, etc... -- some of these decisions may prove to
be "wrong" i.e. when you have users try the prototypes you may discover
problems that lead you to try alternative input/output devices or designs.

So, while you can defer addressing these details for a while, they
do ultimately impact a lot about the system you are going to design,
so I would be careful not to go *too* far where it becomes just a thought
exercise without addressing how it will actually be done.

> Human-Computer Interaction (HCI)

>

> 6. According to the book "The Design of Virtual Environments", the design
> philosophy for your approach is based on the design philosophy known
> as "iterative design" and this philosophy has been used in designing many
> types of human-computer interfaces. However, this methodology is nothing
> new in the field of computer-human interaction, although there are some

> special challenges to applying this approach to virtual environments.
> What are these special challenges? How are these special challenges
> treated in the development process of virtual environments foreseen by
> your approach?

I think I discussed these some in the book. But certainly, a few of them are: 1) no common/standard platform available for VE's; 2) shortage of good rapid prototyping tools for VE's; 3) few labs have the range of "flavors" of VE systems to work with; costly systems -> people end up making certain hardware decisions that they get locked into because they can't afford to try alternatives that might prove to work better for the specific application in question. Also, the book addresses some of the others, either explicitly or implicitly -- for example, with so many possible input technologies that can interact in different ways, and different conceptual approaches to navigation I describe, there tend to be many more possibilities (for example in how the user *might* navigate in the environment) than there are in most of the more commonly designed human-computer interfaces. And because it is a relatively young field, there are fewer conclusive studies that help the user narrow down which of these many choices can be eliminated at the outset -- so the space of possible designs is so large, yet it is costly in resources and effort to prototype so many choices. This is an example common to many young areas of design -- as more work is done in the field, there will be a greater body of knowledge to guide the designer in at least narrowing down the choices.


> 7. In agreement with the study accomplished through the book "The Design
> of Virtual Environments", your approach is founded on the approach User
> Centered Design (UCD) of HCI. However, there are not citations in this
> book about this consideration. Please, comment about how your approach
> is based on principles of HCI.

From your question, I suspect that you may be more familiar with things from the software side than from the HCI side. As I mentioned above, there is an entire field of HCI, with conferences (ACM-CHI, Human Factors Society, Interact, etc...). If you read some of the books, journals, and conference proceedings in this field, the answer to this will become clear to you. You might even start by obtaining just a single year's proceedings from SIGCHI (a chapter of ACM -- see www.acm.org/sigchi), which would contain a hodgepodge of papers, but give you a sense of the field. If one did *not* use this perspective, one could, for example, design a VE system by putting focus only on the technology (e.g. throughput, frame rate, number of pixels per frame, and this sort of thing), with no consideration about the actual people who would use the system, why they would use the system, or what *they* need in the system to accomplish the tasks they need to do.

I hope this helps, good luck!

Best,
Rory Stuart

E-mail com a resposta dos autores da metodologia de Stuart para o segundo questionário específico para essa metodologia, elaborado e enviado por esta pesquisadora com base na resposta do primeiro questionário específico para a metodologia de Stuart.

Data: Sábado, Julho 5 2003 07:13 pm
De: Rory Stuart <rory@rorystuart.com> 
Para: milena@dc.ufscar.br
Responder Para: Rory Stuart <rory@rorystuart.com>
Assunto: Re: About VR Systems Approach (Confirmation)
Cabeçalho Completo: Mostrar Cabeçalho

Hello Milena,

I've incorporated my responses, following your questions. I hope this helps!:

> Hello Mr. Stuart,
>
>
> Thank you for your reply and thank you for your advice. You are right
> saying that ACM library has lots of information that I need (I am a
> subscriber), and be sure that I have been looking for other information
> sources to anchor my research. Then, your answers and tips will help me
> to develop some parts of my research.
> But there are some points about your methodology that I would like you to
> ratify my understanding. That is why I am writing you again. If you can
> just confirm my interpretations about some of your answers, it would be
> very helpful.
> I have been contacting other researchers, like Ivonne Rogers and Vildan
> Taniverdi, among others, in order to have a global view about virtual
> environment development approaches considering SE and HCI points of view.
> And I have been considering other researchers, like Roger Pressman and
> Judy Brown, among others, in order to elaborate my questions.
> If it is interesting for you, don't hesitate to ask me the results of my
> studies.
>
>
> Best regards,
> Milena.
>
> -----
> As I sent you and you answered me:
>
> A)
> 3. Among generic phases as (a) Definition (analysis of the system,
> planning of the software project, analysis of requirements), (b)
> Development (software project, code, accomplishment of tests of the
> software) and (c) Maintenance (correction, adaptation, functional
> improvement) foreseen for the development process of common software,
> which are foreseen by your approach for the development process of
> virtual environments? Is it possible to accomplish some kind of
> correspondence among the mentioned generic phases and the steps and or
> activities that compose your approach? If yes, which mentioned generic
> phases and steps and or activities that compose your approach are
> corresponding?
>

> The approach I'm describing involves defining requirements (your "a"),
> followed by prototyping, evaluation, and then iteration: Have the users
> try to accomplish specified tasks using the prototype, observe strengths
> and weaknesses of the design (everything from problems with performance
> of the prototype to problems the users have in figuring out how to
> accomplish the tasks or interact with the system), then revise the
> design & prototype, and do further rounds of human factors - style
> evaluation of users attempting to accomplish tasks until a
> satisfactorily successful design has been validated. At this point,
> implement this design in a version to be deployed (which may involve
> quite a bit more than did the prototype).
> How does this differ in a practical way from the more traditional phases
> you describe in your question? The big difference is that, with that
> traditional approach, the team tends to decide on a final design and
> begin coding the version of the application that is to be deployed (no
> rapid prototyping). If they do *any* testing with users, it tends to
> happen so late in the process that it can't significantly alter the
> design (there is so much already invested in the design choices that
> were made at the beginning); often this kind of testing is a sort of
> superficial exercise to try to prove that they've done something useful
> and "care about the user" but it doesn't take advantage of all of the
> important insights gained from real users (trying to accomplish real
> tasks with a prototype) that can cause radical iterations in design and
> result in a much more useful and usable final system.
>
> My comments: I agree with you. I believe that due the necessary
> integration between VR software and users, prototyping has been showing
> an important requirement for approaches that support VR software
> development process. Studying your book "The Design of Virtual
> Environments", I came to the conclusion that your approach is including
> for virtual environments development process, involving defining
> requirements (my "a" - Definition) and design, prototyping, evaluation
> and implementation (my "b" - Development), using iterative design.
>
> My ask: Please, I would just like to know if you confirm my conclusion.

Yes, I understand your conclusion. In a sense, you could include all of that in "Development," but I would mention a few things to consider about this:

1) the code that is created in generating rapid prototypes may be entirely different than the code used in the final deployed application (or some of it *may* be used in the deployed application, but one should not count on this). The goal of the rapid prototype implementation is to, as quickly as possible, create interfaces that adequately convey the elements you want to examine in usability testing. The final code for the deployed application must be an implementation of the design (which was generated and revised as a result of the iterative design cycle), but it must be one that is bullet-proofed, bug-free, and fully featured. Imagine that the system in question allows multiple maintenance personnel, pilots, etc... to interact in a simulation of an aircraft; the prototype might allow only a couple of users to interact in certain ways with a small section of the aircraft, which is a mockup rather than a fully realized simulation of the aircraft.

2) Your original description of development "(software project, code, accomplishment of tests of the software)" suggests in its wording (at least to me) a conventional model of software engineering, in which it is assumed that there is a design generated w/in the "Definition" which is handed off to programmers for implementation -- this is why I suggest you might think about a separate design phase. In fact, in the model I present in the book, my thinking was that it would be used to generate the *design*. Once

this is generated, programmers can implement the design, and then a team to test reliability and report bugs can do testing, with programmers making corrections based on reported bugs. So, in this big scheme, I might think of this as Definition -> Iterative Design -> Development -> System Testing. If you would prefer to use the framework you've described, I suggest you make it clear all of what you mean by "Development," much of which is not included in that term when it is used in most conventional computer science contexts.

> B)

> 4. In agreement with the study accomplished through the book "The Design of Virtual Environments", your approach is founded on the model of life cycle Prototyping of SE. However, there are not citations in this book about this consideration. Please, comment about how your approach is based on principles of SE.

>

> Try typing in "iterative design" "rapid prototyping" to a search engine such as Google, and you will find many examples of the use of this approach, as described above.

>

> My comments: Again, studying your book "The Design of Virtual Environments", I came to the conclusion that your approach is founded on "iterative design" and it can be also founded on the Prototyping life cycle model from SE.

>

> My ask: However, I would like to know if you thought about the Prototyping life cycle model presented by Pressman in a formal way or if you just thought about the prototyping activity in a more informal way (intuitively) when you defined your methodology.

I just looked up Pressman's work on the web -- I do not recall anything of Pressman's being a source on iterative design, as the lab I was at did iterative design since I joined in 1987, I learned it from other members of the team, and we used it for many user interface design projects (I see a 1991 reference for Pressman) -- it is so many years ago, perhaps I am simply forgetting Pressman -- in any case, one summary I found on the web says:

Pressman identifies an alternative approach suitable for software engineering:

- evaluate the software request and determine whether suitable for prototyping;

- analyst develops an abbreviated representation of requirements;

- an abbreviated set of design specifications are created for the prototype;

prototype software is created, tested and refined;

- the tested prototype is presented to the customer, who uses it to suggest modifications;

- steps 4 and 5 are repeated iteratively.

The next to last of these steps ("the tested prototype is presented to the customer, who uses it to suggest modifications") is very different from the process I'm describing -- this is more of a focus group or market study test approach. What I'm describing assumes that we know the tasks the user will be doing, and are testing usability, with some specific quantifiable criteria (task completion rates, time to complete tasks, error rates, learning, etc...). Asking the user to "suggest modifications" is alright, but it is a *very small* part of usability testing. By contrast, a marketing group might ask people to imagine a system or try a little demo and then ask them "what would you like this to do? how would

you like us to modify it?"

If that synopsis of Pressman is correct

(<<http://www.docm.mmu.ac.uk/online/SAD/T12/proto.htm>>) --

and, of course, it may not be(!) -- then it looks like the procedure I described is actually more "formal" than Pressman's, as it has specific objectives, specific tasks for the user to complete during usability tests, and a number of objective (as well as some subjective) measures used to assess usability. (I don't have any particular drive to be "formal," but I have found a lot is missed if you only ask users for suggestions!).

> C)

> 5. Still in agreement with the study accomplished through the book "The Design of Virtual Environments", your approach presents including steps in regard to VR technology, that they describe in a detailed way, among others, technological characteristics of available input and output devices. However, for the development process of virtual environments, more specifically for the development process of VR software, in high level of abstraction, some of these descriptions could be suppressed, so that only its implications for virtual environments were portrayed in this development process. Please, comment about your opinion about this suggestion.

>

> If I understand what you're asking, it is along the lines of "could we plan

> some general things about the VE system we want to design before we worry about the specific input and output devices a user would use to interact with the system?". Yes, I think you certainly could. You might start to plan a system for training pilots to use a new type of helicopter, for example. You might hypothesize that it would be useful for the pilot to be able to practice doing certain maneuvers as though they were in the cockpit of the helicopter, and view and hear things from that perspective; but that it would also be useful for them to be able to "replay" the maneuver they just did, and observe it from a fixed point in space, as a sort of 3rd party observer. You could think about a lot of these sorts of issues before you ever worried about whether the system you design will use particular kinds of motion platforms, visual displays, real or virtual cockpit controls, etc... Of course, in order to even begin to do a first prototype, you will need to make certain decisions about input and output devices, interaction styles, etc... -- some of these decisions may prove to be "wrong" i.e. when you have users try the prototypes you may discover problems that lead you to try alternative input/output devices or designs. So, while you can defer addressing these details for a while, they do ultimately impact a lot about the system you are going to design, so I would be careful not to go *too* far where it becomes just a thought exercise without addressing how it will actually be done.

>

> My comments: I believe that your answer to my question no. 3 describes very well how the virtual environment development process should be in a high level of abstraction. Nevertheless, your book brings some informations (like: in chapter 1 pag 19 - basics of visual perception, pag 30 - basics of auditory perception, pag 35 - basics of haptic perception, pag 41 - basics of olfactory perception, among others in the next chapters) that I think they are important, but could be suppressed to describe how the virtual environment development process should be (independent of technological informations). You are giving us some informations that are not vital to the virtual environment specification. They are complementary.

>

> My ask: Please, I would just like to know if you agree with me.

I agree that, at a high level of abstraction, these details of human perceptual systems need not be your focus -- and some of them *never* will be your concern (for example, if you are creating a system with no olfactory component, there is no need to consider olfactory perception). At the high level of abstraction, you might, for example, simply specify that an element in the visual display be shown at sufficient resolution that the user can detect a certain feature.***

Past this high level abstraction, though, when you are actually specifying requirements for the design, you must visit these details. For example what *is* the resolution required for the visual display if the user is to be able to detect this feature (and this requires visiting those details of visual perception).

*** To take another example, I recall Steve Ellis of NASA Ames (in a talk he gave at a VR conference I chaired) explaining that they had found that, for a particular application they were developing, it was necessary that the position trackers and displays handle yaw and pitch, but that head roll was not needed. In this case, there would be no need to consider perceptual system as applied to head roll...

> D)

> 7. In agreement with the study accomplished through the book "The Design of Virtual Environments", your approach is founded on the approach User Centered Design (UCD) of HCI. However, there are not citations in this book about this consideration. Please, comment about how your approach is based on principles of HCI.

>

> From your question, I suspect that you may be more familiar with things from the software side than from the HCI side. As I mentioned above, there is an entire field of HCI, with conferences (ACM-CHI, Human Factors Society, Interact, etc...). If you read some of the books, journals, and conference proceedings in this field, the answer to this will become clear to you. You might even start by obtaining just a single year's proceedings from SIGCHI (a chapter of ACM -- see www.acm.org/sigchi), which would contain a Hodgepodge of papers, but give you a sense of the field. If one did *not* use this perspective, one could, for example, design a VE system by putting focus only on the technology (e.g. throughput, frame rate, number of pixels per frame, and this sort of thing), with no consideration about the actual people who would use the system, why they would use the system, or what *they* need in the system to accomplish the tasks they need to do.

>

> My comments: I agree with you. I believe that it is very important to keep the focus on the user. This focus is mandatory for approaches that support the development process of VR software.

> Studying your book, I came to the conclusion that your approach presents this focus on user and your approach could be founded on User Centered Design (UCD) approach from HCI.

Milena, let me give an example in a speech recognition interface I worked on, of this distinction between focus on the user vs. purely technology-based focus. We did usability studies of a speech recognition system, and the development group helped create prototypes. One thing they asked us to test was recognition rates (misrecognitions, in which the user says one word that the system has in its database, and the system thinks it is another; and failure to reject, in which the user says a word that is not known by the system, but the system thinks it is one of the

words that it knows). Meanwhile, the development group was doing their own tests on recognition rates of their software. Both groups presented their findings during a meeting we had.

I don't remember the exact numbers, but I will make up numbers to give you a sense of what happened: The usability group showed recognition rates of 84%, and the development group showed rates of 99.734%.

How could the rates have been so different? It turns out the development Group recorded some utterances using very high quality microphones with a person sitting in a special anechoic (completely quiet) room. Then they precisely end-pointed these recordings (by hand), so that they had nearly perfect examples of each word, and ran the words through the recognizer repeatedly.

In contrast, the usability group had the users speak to the recognizer using the prototype of the system, which was phone-based, in a regular (non-anechoic) room. Sometimes, the user would speak too soon (before the beep on the phone); sometimes, they would cough before they spoke; sometimes there would be some background noise (their chair squeaking, etc...). These were all things that would happen in the real world if people were using this recognizer, but the development group was appalled that we would consider any of these cases, where their recognizer didn't get a perfect utterance to work on.

The technology-centered approach they had would, for example, have suggested there was little need for a simple efficient way for the user to correct misrecognitions (since they thought these would be so rare); while our user-centered approach indicated this was essential (misrecognitions would actually be quite common with real users using the system in the real world, users had to have a way to correct them)!

> My ask: However, I would like to know if you thought about adopting UCD
> approach in a formal way or if you thought about keep the focus on the
> user in a more informal way (intuitively) when you defined your
> methodology.


To be candid, I attended some UCD symposia, and read some in the early '90's but, at least back then, I never found anything in the UCD world that I would have called a truly "formal approach" -- it seemed informal in every way! Perhaps that has changed in the last several years?

Good luck!

Best,
Rory

Anexo D

E-mail com a resposta dos autores da metodologia de Scaife e Rogers para o questionário específico para essa metodologia, elaborado e enviado por esta pesquisadora.

Data:	Mon, 01 Dec 2003 11:54:55 -0500
De:	Yvonne Rogers <yrogers@indiana.edu> 
Para:	milena@dc.ufscar.br
Responder Para:	Yvonne Rogers <yrogers@indiana.edu>
Assunto:	Re: About VR Systems Approach (Questionnaire)
Cabeçalho Completo:	Mostrar Cabeçalho

Milena
You certainly are very persistent. Here are some brief answers as requested
Yvonne Rogers

> -----
> Approach of Scaife and Rogers
>
>
> Context in Virtual Reality (VR)
>
> 1. Among terms like (a) VR system, (b) VR software, (c) VR hardware,
> (c) VR application, (d) virtual environment and (e) VR user interface
> or just VR interface, virtual environment is the principal term used
> in your approach presented in your article "Informing the design of the
> virtual environment to support learning in children" (Scaife, M.;
> Rogers, Y. In Journal of Human- Computer Studies, vol.55, no. 2, August
> 2001, 115-143.). How is the term virtual environment understood in the
> context of this approach? Please, comment about your definitions for
> these mentioned terms.
Virtual environment focuses on the space where the interactions take place
rather than the 'reality' attempted to be built. Thus it is an environment
for learning.
>
> 2. How is your approach classified: (a) Model of life cycle, (b)
> Methodology, (c) Method and or (d) other? If other, which? Please,
> comment about your definitions for these mentioned terms.
Methodology, as it is a number of approaches and methods.
>
> 3. Among generic phases as (a) Definition (analysis of the system,
> planning of the software project, analysis of requirements), (b)
> Development (software project, code, accomplishment of tests of the
> software) and (c) Maintenance (correction, adaptation, functional
> improvement) foreseen for the development process of common softwares,
> which are foreseen by your approach for the development process of
> virtual environments? Is it possible to accomplish some kind of
> correspondence among the mentioned generic phases and the stages
> and or activities that compose your approach? If yes, which mentioned
> generic phases and stages and or activities that compose your approach
> are corresponding?
Not sure how to answer this.
>
> 4. According to some citations contained in the article "Informing the
> design of the virtual environment to support learning in children",
> your approach is addressed to the development process of virtual

> environments that support several learning activities. According to
 > other citations contained in this same article, your approach is
 > addressed to the development process of virtual environment in a
 > generic way. Please, comment about this regard.

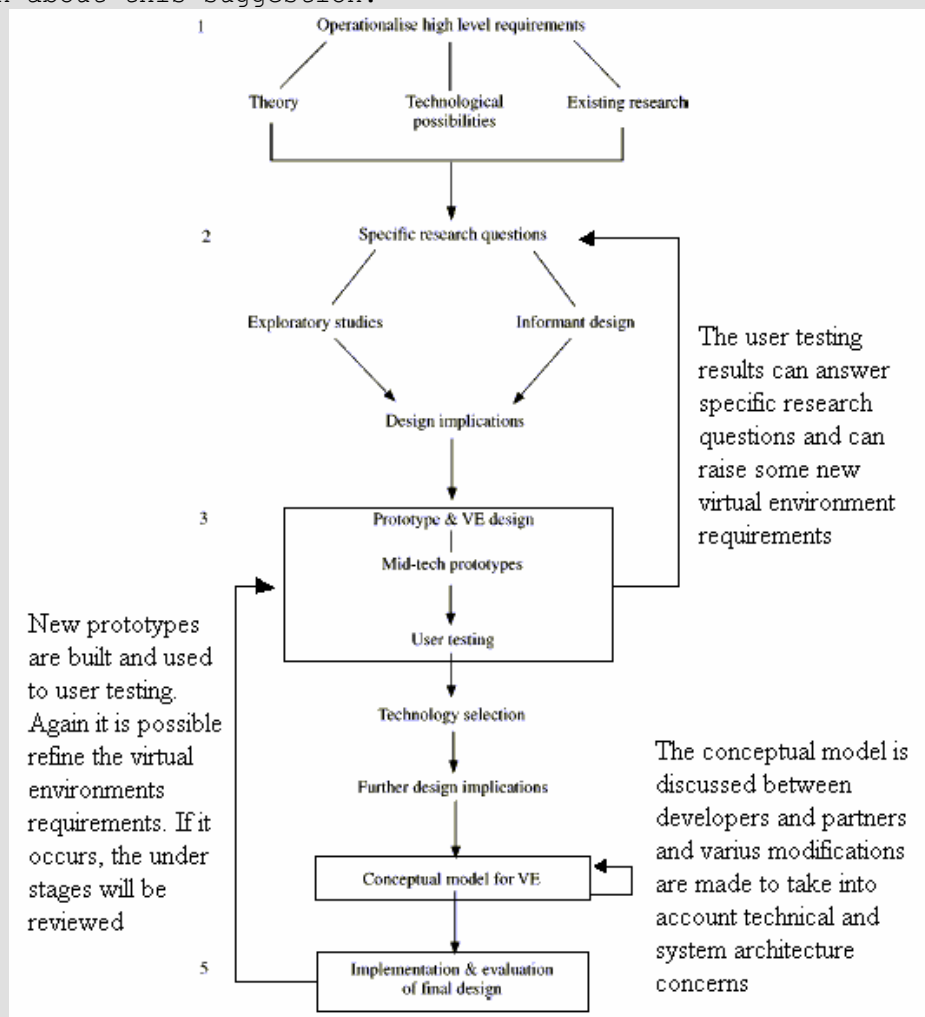
It can be used for other applications besides learning and hence is generic.

>
 >
 > Software Engineering (SE)

> 5. In agreement with the study accomplished through the article
 > "Informing the design of the virtual environment to support learning in
 > children", your approach is founded on the model of life cycle
 > Prototyping of SE. However, there are not citations in this article
 > about this consideration. Please, comment about how your approach is
 > based on principles of SE.

It is not really about SE but about VEs and which methods to use.

>
 > 6. Still in agreement with the study accomplished through the
 > article "Informing the design of the virtual environment to support
 > learning in children", your approach is proposed through a linear
 > structure of development process for virtual environments. However, to
 > assist the need of a iterative structure of development process
 > demanded by VR softwares, some cycles could be added in this linear
 > structure, as showed in the annexed picture. Please, comment about your
 > opinion about this suggestion.



We agree it can be iterative too.

```
>
>
> Human-Computer Interaction (HCI)
>
> 7. According to the article "Informing the design of the virtual
> environment to support learning in children", your approach is founded
> on the approach User Centered Design (UCD) of HCI. The focus on the
> user has been showing if as important requirement for the development
> of approaches that they support the development process of VR
> softwares. How does your approach consider the focus on the user during
> the development process of virtual environments? Please, comment about
> how your approach is based on principles of HCI.
HCI principles are very important in any design process and we follow them
too.
> -----
```
