

Usabilidade de Interfaces com Realidade Virtual para Sistemas que passaram por Reengenharia

Cristiano L. Soares, Júnia C. Anacleto Silva, Rosângela Ap. Deloso Penteadó

Universidade Federal de São Carlos (UFSCar), Departamento de Computação,

São Carlos - SP, Brasil, 13565-905

anechini@yahoo.com {junia,rosangel}@dc.ufscar.br

Abstract

This paper reports an usability evaluation of the implementation paradigm influence to the user interface reengineering process of a system using a computational tool to support the new interfaces generation. In this work the GaCIV environment - Configurable Templates for Development of Virtual Reality Interfaces, in portuguese - is used as support to generate interfaces using virtual reality for three versions of an example system: legacy system (originally implemented using Clipper), segmented reengineered system (implemented using Clipper with object-oriented paradigm characteristics), and the object-oriented reengineered code (implemented in Java).

Keywords: VR interfaces, interface reengineering, VR interfaces usability, GaCIV.

Resumo

Este artigo apresenta uma avaliação de usabilidade sobre a influência do paradigma de implementação no processo de reengenharia de interfaces de um sistema utilizando uma ferramenta computacional como apoio para a geração das novas interfaces. Neste artigo o ambiente GaCIV – Gabaritos Configuráveis para elaboração de Interfaces com RV – é usado como apoio para a criação de interfaces utilizando RV para três versões de um sistema exemplo: versão legada (implementado originalmente em Clipper), versão segmentada (resultante de um processo de reengenharia, em Clipper com características de orientação a objetos), e a versão orientada a objetos resultante de outro processo de reengenharia (implementado em Java).

Palavras-chaves: interfaces com RV, reengenharia de interfaces, usabilidade de interfaces com RV, GaCIV.

1 Introdução

A incorporação de novas tecnologias de interface em sistemas existentes pode trazer como resultado aumento substancial de usabilidade através da melhoria da *performance* e satisfação dos usuários, reduzindo a necessidade de treinamento e o número de erros cometidos pelo usuário [1]. Entretanto, de acordo com Nielsen [2], utilizar um tipo diferente de tecnologia de interface não traz automaticamente melhorias na usabilidade do sistema interativo. Nenhuma interface pós-WIMP (*Windows, Icons, Menus and Pointing devices*), incluindo interface com realidade virtual (RV), está livre de problemas relacionados à usabilidade.

RV é uma tecnologia que tem como principal característica exploração de diversos sentidos do usuário para causar sensação de imersão ao usuário em um ambiente virtual quando ele manipula objetos e realiza tarefas de maneira semelhante à realizada no mundo real [3]. Dessa forma a RV torna a interação do usuário mais intuitiva e natural atingindo muitos aspectos de usabilidade desejados para interfaces de sistemas interativos.

Apesar de toda a extensa pesquisa realizada no campo de RV e em usabilidade, ainda há pouca proximidade entre elas. Existem diversos métodos para avaliação de usabilidade de aplicações interativas, porém esses métodos são limitados quando se trata de avaliação de ambientes virtuais. Grande parte das publicações que relacionam usabilidade e RV trata de problemas específicos tais como diretivas para mecanismos de entrada e dicas para projeto de interfaces 3D, portanto, falhando em abranger as complexas interdependências entre usuários, tarefas, mecanismos de entrada, mecanismos de saída, etc, existentes em um ambiente de RV [4].

O teste de usabilidade de interfaces com RV imersiva junto ao usuário, por exemplo, pode apresentar diversas dificuldades tais como a dificuldade de compartilhar uma mesma experiência de interação entre o usuário e o avaliador e obter informações úteis a partir dela. Possíveis problemas dessa natureza, entretanto, devem servir como estímulo ao teste de interfaces com RV junto ao usuário, pois indicam a necessidade de treinamento especial para avaliadores e, até mesmo, criação de novas ferramentas e processos para o teste de usabilidade nesse ambiente.

A complexidade de testes com participação do usuário sugere o uso de um método de avaliação heurística baseado em um conjunto de heurísticas de usabilidade. Atualmente poucas abordagens de heurísticas de usabilidade específicas para interfaces com RV podem ser encontradas na literatura. Dois exemplos significativos são encontrados em [4] e em [5]. Essas limitações dificultam a tarefa do engenheiro de software que necessita garantir a usabilidade da interface com RV gerada, tanto na engenharia avante quanto na reengenharia de interfaces de sistemas legados.

Neste trabalho, financiado pela Fapesp (processo 00/04577-0), três versões de um mesmo sistema são consideradas na realização de um processo de reengenharia de suas interfaces para interfaces com RV. As novas interfaces foram geradas utilizando o ambiente GaCIV [6].

A reengenharia de sistemas, segundo a taxonomia de Chikofsky e Cross [7], é “o exame e alteração de um sistema para reconstituição em uma nova forma, e subsequente implementação dessa nova forma”. A reengenharia de sistemas implica na fase de engenharia reversa para a recuperação de uma visão abstrata do sistema e uma fase de transformação ou re-interpretação dessa visão seguida de nova implementação.

Durante a fase de reengenharia das interfaces das três versões do sistema exemplo, a engenharia de usabilidade para as novas interfaces com RV está sendo abordada através das diretivas reunidas por Gabbard em uma taxonomia de características de usabilidade desejadas para um ambiente virtual [4]. As diretivas estão sendo consideradas especificamente na geração de interfaces com RV apoiada pelo GaCIV.

Neste artigo o sistema exemplo é descrito na seção 2, os critérios de avaliação heurística aplicados nas três versões do sistema são apresentados na seção 3, uma avaliação das interfaces com RV criadas com o apoio do GaCIV é apresentada na seção 4 e na seção 5 são analisados, de forma comparativa, os resultados obtidos com reengenharia das interfaces das três versões do sistema. As conclusões obtidas a partir dos resultados são apresentadas na seção 6.

2 Sistema Exemplo - Três Versões

O sistema exemplo utilizado neste estudo é o mesmo apresentado em [8]. Ele foi originalmente desenvolvido em Clipper 5.0, com aproximadamente 20K linhas de código, com 25 tabelas do tipo “dbf”, relacionadas por chaves estrangeiras. Trata-se de um sistema de uma oficina mecânica real que controla os serviços executados em veículos, bem como o controle de estoque das peças utilizadas e de sua reposição. O usuário típico do sistema exemplo é o funcionário da oficina. Esse sistema, originalmente com uma estrutura de código monolítica e referenciado neste trabalho como a “versão legada do sistema exemplo”. A segunda versão considerada neste trabalho é a que passou pela fase de

engenharia reversa utilizando o método Fusion/RE e seu código foi reorganizado mantendo-se a linguagem legada porém seguindo quando possível as características de orientação a objetos. Essa versão é chamada de "segmentada" [8].

Com base no modelo de análise obtida durante a realização da engenharia reversa, neste projeto, realizou-se a engenharia avante implementando o sistema em linguagem OO, Java, que é referenciada neste artigo como "versão OO do sistema".

O estilo da interface das versões legada e segmentada, implementadas em Clipper, está ilustrada na figura 1.

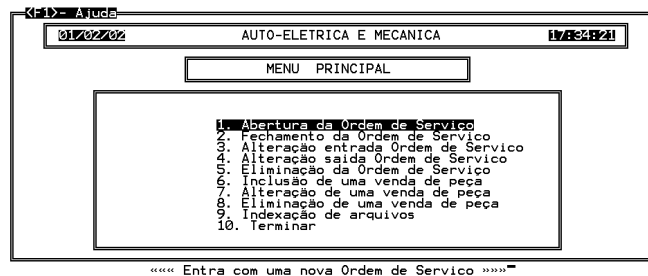
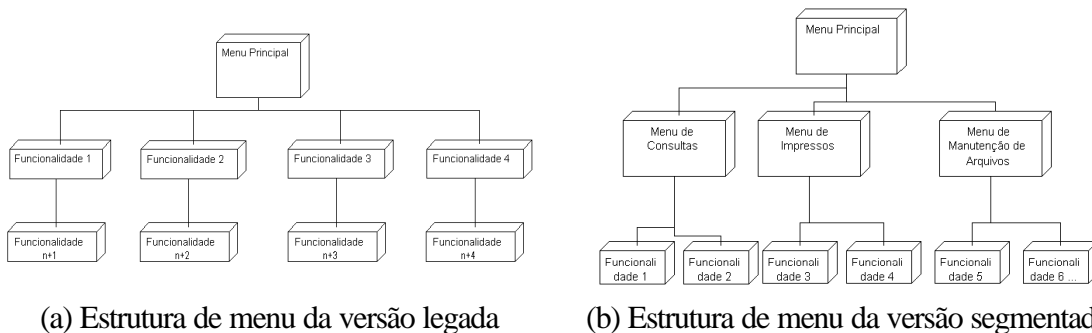


Figura 1 – Estilo de interface do sistema exemplo nas versões legada e segmentada.

As estruturas das árvores de menus das interfaces das versões legada e segmentada são representadas na figura 2.



(a) Estrutura de menu da versão legada

(b) Estrutura de menu da versão segmentada

Figura 2 – Estruturas das árvores de menus das versões legada e segmentada do sistema exemplo.

Nota-se no diagrama da figura 2(a) que a árvore de menus é pouco profunda, com praticamente apenas um nível de menu para se ter o acesso às funcionalidades. Apesar de aparentemente simplificar a navegação do usuário, esse tipo de estrutura compromete a organização das funcionalidades oferecidas pelo sistema. Na versão legada do sistema exemplo, algumas funções independentes estão associadas sob um mesmo rótulo na forma de sub-tarefas, como acontece com a inclusão de dados de um cliente no banco de dados que está associado “Lançamento da Ordem de Serviço”.

Para que o usuário possa cadastrar um cliente ele deve selecionar a função “Abertura da Ordem de Serviço” e digitar “99999” no espaço dedicado ao código do cliente. Essa organização não natural das funcionalidades gera carga cognitiva desnecessária para o usuário ou mesmo desorientação. Além disso, a orientação do usuário fica prejudicada quando se passa de uma sub-tarefa para outra dentro de uma mesma tarefa. Para executar a tarefa “Abrir Ordem de Serviço” o usuário deve cumprir uma série de sub-tarefas como “Selecionar Cliente”, “Verificar Dados do Cliente”, “Selecionar Veículo do Cliente”, “Verificar Veículo do Cliente” e finalmente descrever os serviços a serem executados.

A estrutura da árvore de menus da interface da versão segmentada, representada na figura 2(b), mostra-se um pouco mais profunda que a apresentada pela versão legada. Há duas ou mais camadas da árvore de menus para que o usuário acesse a funcionalidade desejada. Essa organização permite que funcionalidades independentes encontrem-se separadas e classificadas segundo algum critério que satisfaça as necessidades do usuário. Por exemplo, as tarefas relacionadas a consultas à base de dados, foram rotuladas como “Consultas” e dessa forma acessadas a partir do menu principal.

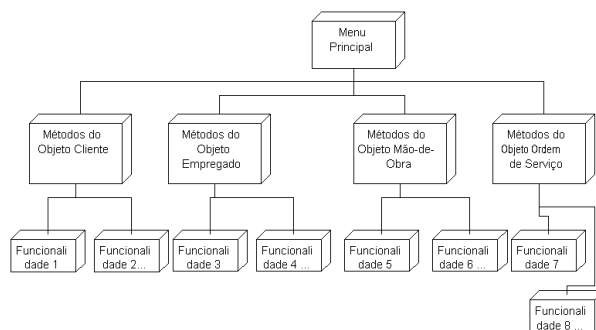
Atividade de manutenção de dados da base de dados receberam o rótulo de “Manutenção de Arquivos”, etc. A reorganização das funcionalidades, facilitando o reconhecimento pelo usuário e conseqüentemente melhorando a *performance* na realização de tarefas, foi possível devido ao processo de segmentação ao qual a versão legada foi submetido em um trabalho anterior [8].

A interface da versão orientada a objetos baseia-se em *widjets* WIMP e segue o paradigma de interação orientada a objetos (OOUI), ao invés do paradigma funcional usado anteriormente. Os objetos do domínio do sistema são representados por ícones gráficos que podem ser reorganizados de acordo com a preferência do usuário. Os ícones substituem os menus principais das versões anteriores do sistema. A organização dos objetos na interface pode ser modificada através da ação “arrastar e soltar”. Nessa interface, como em qualquer OOUI, o usuário escolhe o objeto da interface, e em seguida utiliza os métodos oferecidos por ele para interagir com o objeto. O usuário deve clicar com o botão direito do mouse e obter um menu *popup* com os métodos apropriados para a interação. Cada um dos “ícones – objetos” apresenta sua identificação quando o cursor do mouse é posicionado sobre eles como mostra a figura 3(a).

A estrutura da versão OO do sistema é mostrada na figura 3(b).



a-) Interface da versão OO



b-) Estrutura de menu da versão OO

Figura 3 – Interface da versão OO do sistema exemplo

3 Critérios de Avaliação de Usabilidade Utilizados

Durante o estudo de caso, as interfaces geradas pelos processos de reengenharia e as interfaces legadas foram analisadas utilizando heurísticas de usabilidade. Para as interfaces textuais e WIMP utilizou-se as seguintes diretivas apresentadas em [9]:

1. Visibilidade do estado do sistema. Os sistemas devem sempre manter o usuário informado sobre o que ocorre, através de *feedback* em tempo razoável.
2. Concordância entre o sistema e o mundo real. O sistema deve usar linguagem familiar ao usuário com conceitos familiares ao usuário ao invés de termos ligados ao sistema. Seguir convenções do mundo real, fazendo a informação aparecer de forma natural e em ordem lógica.
3. Controle e liberdade do usuário. Usuários geralmente escolhem tarefas por engano das quais ele deve poder sair imediatamente sem que seja forçado a passar por um diálogo extenso desnecessário. Deve haver também apoio ao “desfazer” e “refazer”.
4. Consistência e padrões. Usuários não devem se preocupar se palavras, situações e ações diferentes possuem o mesmo significado.
5. Prevenção de erros. É melhor evitar erros do que corrigi-los.
6. Reconhecer, não relembrar. Tornar objetos, ações e opções visíveis. O usuário não deve ter que se lembrar de uma parte da interface para a outra.

7. Eficiência de uso e reconhecimento. Oferecer atalhos para melhorar a *performance* de usuários experientes.
8. Estética e design. Diálogos não devem conter informações que não sejam relevantes ou raramente usadas.
9. Ajuda para reconhecimento, diagnóstico e recuperação de erros. Mensagens de erros devem ser expressadas em linguagem simples, sem usar códigos, indicando precisamente o problema e sugerindo soluções.
10. Ajuda e documentação. Mesmo que o sistema seja melhor utilizado sem documentação é importante que haja documentação e ajuda para o usuário.

Para analisar a usabilidade das interfaces com RV geradas com apoio do GaCIV, nas três versões do sistema exemplo durante o processo de reengenharia de interface, foram observadas as seguintes diretrizes extraídas da taxonomia de Gabbard [3] de acordo com a aplicabilidade no sistema em questão. As tabelas de 1 a 6 apresentam o contexto das diretrizes, os rótulos pelos quais serão referenciadas no texto e a descrição da sugestão.

Tabela 1 – Diretrizes para “Usuários em Ambientes Virtuais”

Usuários em Ambientes Virtuais	
Rótulo	Diretiva de Usabilidade
Usuários1	Contar com a experiência do usuário
Usuários2	Contar com as aptidões técnicas do usuário (orientação, visualização espacial etc)
Usuários3	Apoiar tanto usuários destros quanto canhotos
Usuários4	Acomodar interação natural para usuários de várias idades

Tabela 2 – Diretrizes para “Navegação e Locomoção”

Navegação e Locomoção	
Rótulo	Sugestão de Usabilidade
Nav1	Considerar princípios organizacionais na construção do ambiente
Nav2	Evitar navegação baseada em modo

Tabela 3 – Diretrizes para “Seleção de Objeto”

Seleção de Objeto	
Rótulo	Sugestão de Usabilidade
Seleção1	Usar manipulação direta para seleções baseadas em atributos espaciais. (forma, local, orientação)
Seleção2	Quando houver seleção de objetos distantes, exagerar o tamanho, aparência e inter – distância de objetos
Seleção3	Seleção de objetos deve ser o mais óbvio e acessível possível

Tabela 4 – Diretiva para “Apresentação do Usuário e Representação”

Apresentação do Usuário e Representação	
Rótulo	Sugestão de Usabilidade
URep1	Prover ponto de vista egocêntrico quando usuários devem experimentar um forte senso de presença

Tabela 5 – Diretiva para “Envolvimento Virtual e Configuração”

Envolvimento Virtual e Configuração	
Rótulo	Sugestão de Usabilidade
Config1	Explorar experiências do mundo real, mapeando funcionalidades desejadas para itens do dia a dia

Tabela 6 – Diretivas para “Mecanismos de Entrada em Geral”

Mecanismos de Entrada em Geral	
Rótulo	Sugestão de Usabilidade
Entrada1	Eliminar graus-de-liberdade estranhos implementando apenas as dimensões que os usuários percebem como relacionadas as suas tarefas
Entrada2	Segundo o ponto de vista do usuário, o dispositivo de saída deve ser consistente e ligado cognitivamente a suas ações
Entrada3	Diminuir a carga cognitiva evitando dispositivos tais como joysticks e bastões
Entrada4	Evitar integração entre dispositivos de entrada tradicionais como teclados e mouses e dispositivos de entrada 3D

4 Interfaces com RV geradas pelo GaCIV

O ambiente GaCIV [6], uma ferramenta atualmente em desenvolvimento no Departamento de Ciência da Computação da Universidade Federal de São Carlos, tem o objetivo de prover apoio rápido e viável para a criação de interfaces com RV, através de uma biblioteca de gabaritos configuráveis, cada um gerado de acordo com o domínio da aplicação considerada, contendo um conjunto de objetos 3D em um ambiente virtual. Esse gabarito pode ser customizado pela equipe de desenvolvimento (projetistas de interfaces, engenheiros de software e usuários), seguindo os requisitos levantados. Esses gabaritos configurados geram as interfaces que são ligadas à aplicação, gerando assim um sistema interativo com interfaces com RV.

Como mostrado em [6] (figuras 4 e 5), o GaCIV, através do recurso de *link*, integra os componentes da interface gerada por ele com as aplicações desenvolvidas independentemente. Como descrito em [6], um *link* é uma interconexão entre o objeto da interface e um arquivo executável, fazendo com que este objeto na interface represente a execução da tarefa.

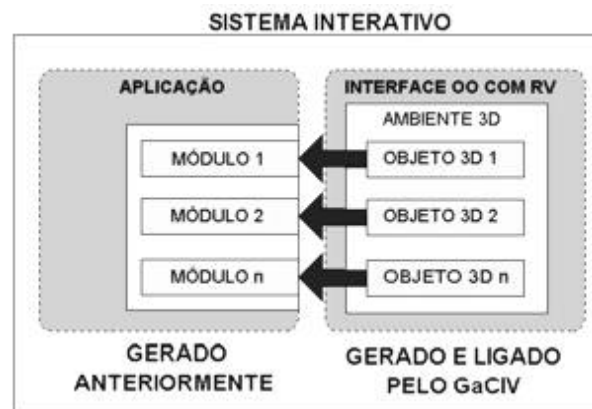


Figura 4– Criação de Interfaces no ambiente GaCIV

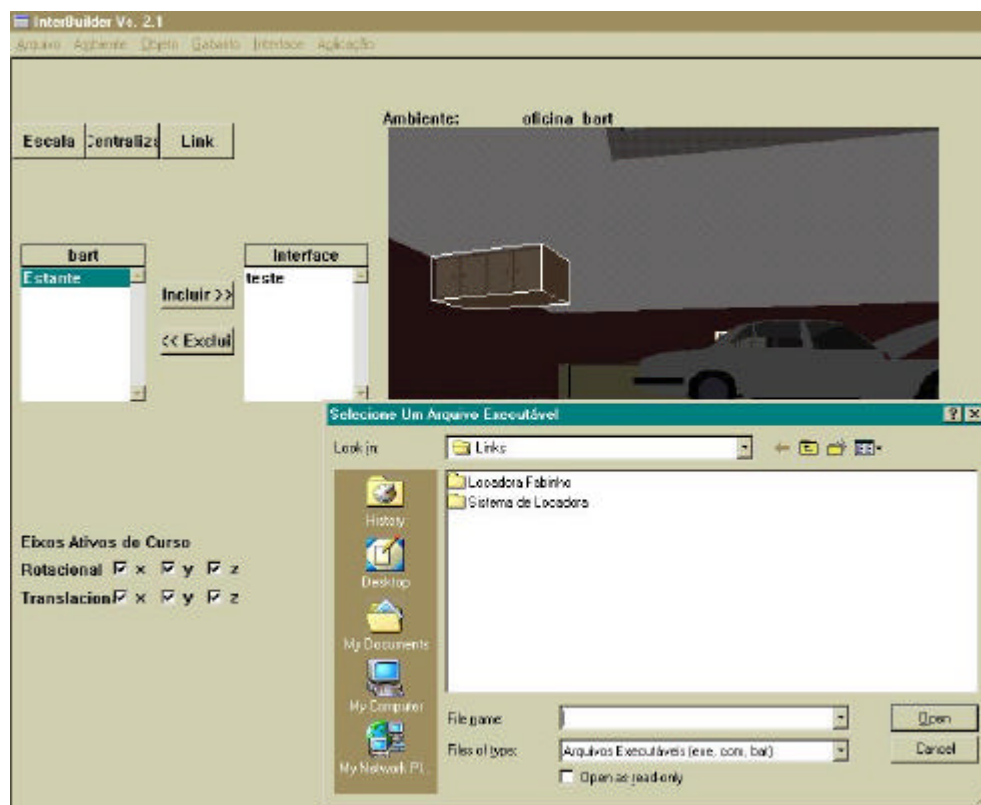


Figura 5– Atribuição de *link* ao objeto no ambiente GaCIV

As principais características das interfaces com RV geradas utilizando o ambiente GaCIV são:

- 1 Natureza não-imersiva, isto é, apresentam objetos e ambientes 3D que podem ser visualizados em dispositivos 2D tais como monitores comuns, sem a necessidade de equipamentos avançados e caros como por exemplo *head-mounted displays*;
- 2 Preservam o estilo de interação 2D utilizado em interfaces WIMP, sendo o mouse comum o único dispositivo de entrada necessário para a seleção e manipulação de objetos. Objetos podem ser selecionados desde que estejam no campo de visão do usuário independente da distância que os separam. Objetos apresentados no ambiente estão ligados a aplicações que são executadas quando os objetos são selecionados, funcionando como “ícones 3D” que representam as funcionalidades do sistema. Os objetos não interagem entre si;
- 3 Navegação não exige posicionamento preciso do usuário para que tarefas sejam executadas e é realizada utilizando o mouse como dispositivo de entrada. O cursor indica a direção a ser seguida pelo usuário e o movimento no plano XZ é feito através do pressionar contínuo do botão esquerdo enquanto o movimento no plano XY é feito através do pressionar contínuo do botão direito;
- 4 Substituem ou adicionam camadas da árvore de interface de aplicações desenvolvidas previamente, resultando, em muitos casos, em interfaces mistas combinando ambientes e objetos 3D e interfaces 2D WIMP ou mesmo textuais;
- 5 Objetos e ambiente 3D são gerados e posicionados de acordo com o domínio da aplicação e as necessidades do usuário.
- 6 O ponto de vista do usuário no ambiente virtual gerado é egocêntrico.

Analisando as características das interfaces geradas com o GaCIV durante os processos de reengenharia de interfaces, em relação às características de usabilidade extraídas de [3] e citadas anteriormente, foram obtidos os seguintes resultados:

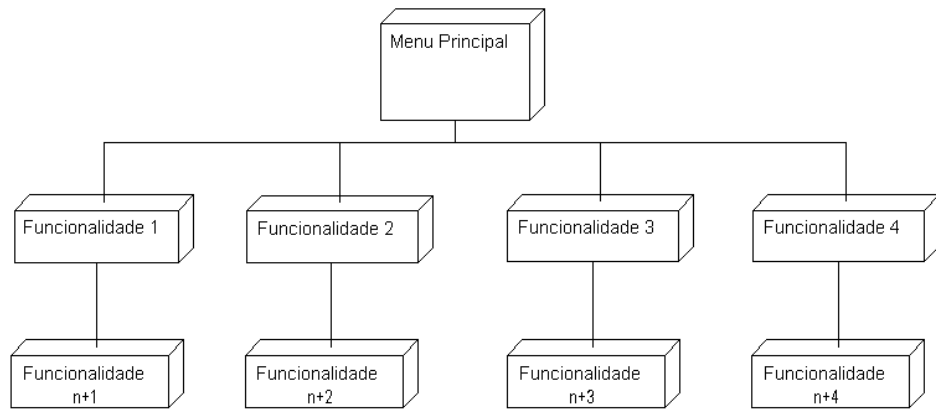
- A utilização da RV não-imersiva mostrou-se positiva em relação à diretiva “Entrada2” pois o uso de monitores comuns para a apresentação dos gráficos 3D evita a necessidade de treino para utilização de dispositivos especiais. Além disso, o usuário também entrará em contato com menus 2D da aplicação, menus estes que são comprovadamente eficientes em monitores simples.
- Tendo como base a diretiva “Entrada3” e “Entrada4”, associando-a ao contexto da mudança de interfaces puramente textuais para interfaces RV, conclui-se que o uso de mecanismos avançados de entrada necessários para a navegação em um ambiente de RV imersiva poderia aumentar o esforço cognitivo do usuário. As interfaces geradas neste trabalho permitem que os usuários do sistema original continuem utilizando os mesmos mecanismos de entradas aos quais ele já estava habituado, isto é, mouse e teclado.
- Ainda em relação aos dispositivos de entrada, a opção por dispositivos 2D vai de acordo com as diretivas “Usuários1”, “Usuários2”, “Usuários3” e “Usuários4” pois trata-se de dispositivos consolidados e populares aos quais a grande maioria dos usuários está familiarizada.
- A seleção de objetos permitida pelo GaCIV é simples e intuitiva pois se aproxima da idéia de seleção no mundo real [3], como indicado em “Seleção1” e “Seleção3”. Como os objetos podem ser selecionados independentemente da distância entre ele e o usuário, evita-se a necessidade de excessiva locomoção do usuário dentro do ambiente RV para acessar suas tarefas. Entretanto para que isso ocorra, os objetos apresentados devem ter formas, cores, tamanho e posicionamento que os distingam bem um dos outros, como observado em “Seleção2”, cuidado este que foi tomado durante a geração das interfaces.
- Durante a reengenharia de interfaces textuais para interfaces com RV, notou-se que realizar o mapeamento da navegação do usuário nas interfaces legadas, através de menus, para objetos 3D da forma mais natural possível é fundamental para se obter uma interface com usabilidade, como indicam as diretivas “Config1” e “Setting4” de [3]. Segundo essas diretivas, o ambiente virtual no qual o usuário está imerso afeta a usabilidade ao transmitir um contexto para as tarefas do usuário.
- O ponto de vista egocêntrico, ou ponto de vista do usuário, nas interfaces com RV geradas pelo GaCIV reforça a idéia de imersão que traz grande parte do ganho em usabilidade ao optar por esse tipo de interface como indica a diretiva “URep1”.
- Apesar de ser possível a movimentação usando os seis graus de liberdade, propostos para ambientes com RV, as interfaces do GaCIV não necessariamente exigem essa movimentação. Os graus de liberdade utilizados durante a navegação do usuário dependem do posicionamento dos objetos, que por sua vez depende do contexto do sistema. As diretivas “Entrada1” e “Nav1” deve ser considerada durante a construção das interfaces, visando o melhor posicionamento dos objetos segundo o domínio do sistema.
- Apesar da interface resultante ser uma interface mista (parte 3D e parte 2D), como o estilo de interação básico não muda (apontar e clicar) não há aumento significativo de esforço cognitivo em relação a esse aspecto.

5 Análise Comparativa do Processo de Reengenharia das Versões do Sistema Exemplo

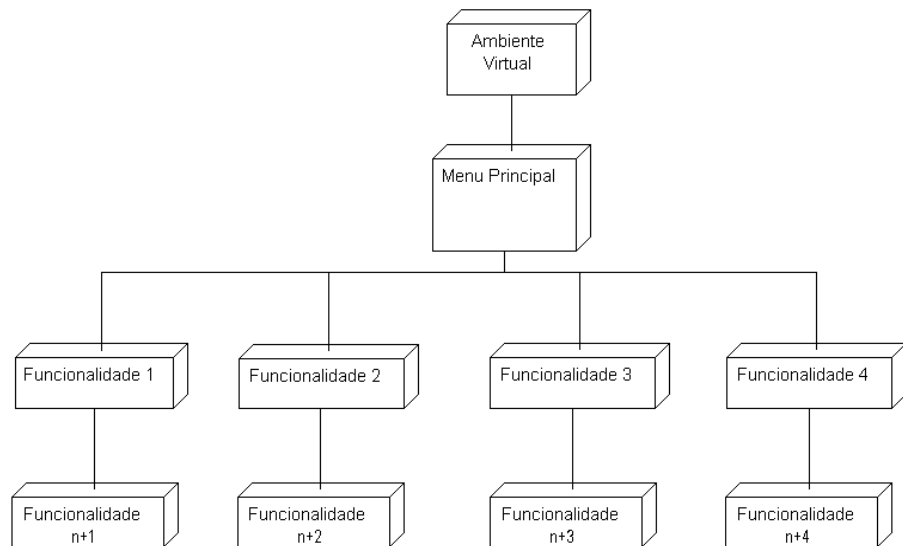
Os processos de reengenharia de interfaces realizados visaram aumentar a usabilidade do sistema ao incrementar o senso de envolvimento do usuário através da modificação significativa de elementos de navegação e apresentação nos níveis mais altos de abstração da árvore de menus do sistema exemplo. Dessa forma, a análise comparativa foi realizada visando principalmente o impacto da reengenharia de interface nas características relacionadas à navegação e apresentação das interfaces.

A reengenharia da interface das três versões do sistema, utilizando o GaCIV como apoio automatizado, modificou significativamente a navegação e acesso às tarefas do usuário devido às características inerentes à tecnologia RV analisadas na seção anterior.

O processo de reengenharia da interface da versão legada apresentou poucos benefícios, pois apenas acrescentou uma camada de interface, aumentando o esforço necessário para o acesso das tarefas fornecidas pelo sistema, como mostra a figura 6.



(a) Interface da Versão Legada



(b) Interface Avançada com RV para a Versão Legada

Figura 6– Tabela comparativa das estruturas de interface para a versão legada do sistema exemplo

A figura 7 apresenta a interface com RV gerada para a versão legada com apoio do ambiente GaCIV.

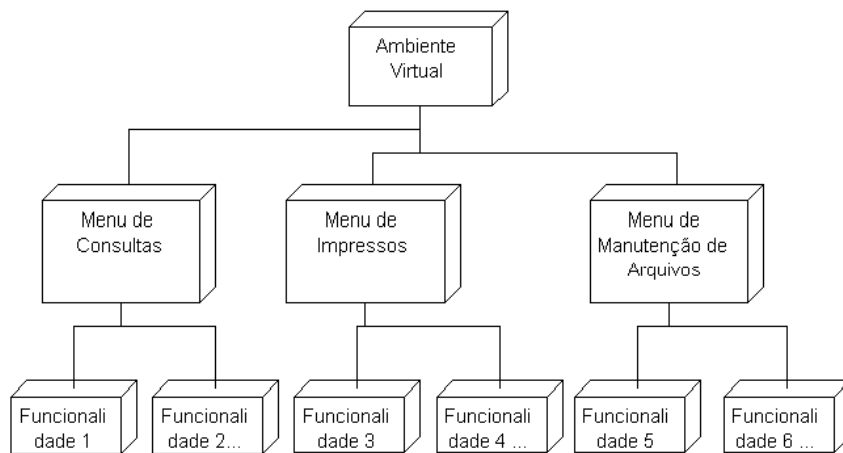


Figura 7– Interface avançada para a versão legada

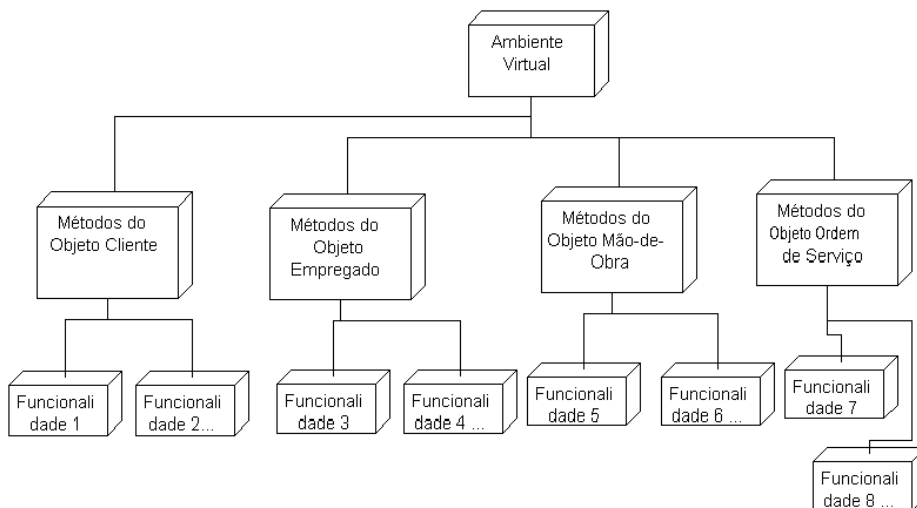
Nas versões segmentadas e OO do sistema, obteve-se maiores benefícios que na versão legada, pois foi possível representar de forma mais natural as funcionalidades dos sistemas em um nível alto de abstração, diminuindo o esforço cognitivo do usuário naquele nível.

Observou-se que na versão OO do sistema que o esforço para a reengenharia das interfaces é muito próximo do despendido para o segmentado. Isso acontece, segundo ponto de vista destes pesquisadores, pela segmentação do sistema ter sido pautada pelo paradigma OO, resultando em estruturas de menus semelhantes (figura 8). Também pode ser observado que nos dois casos o nível mais alto de abstração foi substituído pela nova representação gráfica, isto é, menus textuais ou de gráficos 2D substituídos por gráficos 3D.

A principal diferença entre as interfaces avançadas geradas para a versão segmentada e a versão OO é a natureza dos objetos que compõem a interface. Para a interface da versão segmentada, utilizam-se objetos que abstraem tarefas a ser realizadas pelo usuário, como por exemplo “imprimir”. A interface avançada com RV para a versão segmentada contém objetos gráficos que devem servir de metáforas para ações que na interface original eram expressas em menus, o que pode se tornar complexo em alguns casos como, por exemplo, a ação “lançar ordem de serviço”.



(a) Mapa Parcial da Interface da Versão Segmentada com RV



(b) Versão OO com RV

Figura 8 – Estruturas de interfaces com RV para as versões segmentada e OO

Já na versão OO do sistema, é possível utilizar objetos que refletem os objetos do domínio do sistema, como por exemplo “cliente”, “empregado” ou “peças”. Como o domínio do sistema está relacionado diretamente com objetos do

mundo real, há maior facilidade em representá-los na interface 3D. Dessa forma, a interface avançada resultante do processo de reengenharia de interfaces aplicado no sistema OO apresenta-se mais intuitiva e natural para o usuário de acordo com as diretrizes “Usuários1” e “Config1” citadas anteriormente. A diferença entre a natureza dos objetos está ilustrada na figura 9.

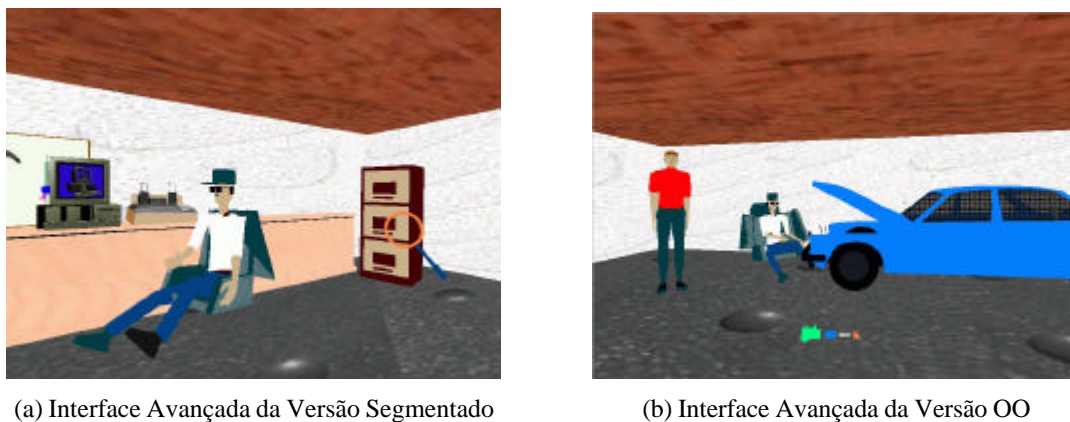


Figura 9 – Interfaces avançadas para as versões segmentada e OO

A vantagem apresentada pelo processo de reengenharia de interface aplicado à versão segmentada do sistema é que os objetos 3D obtidos para representar ações do usuário na interface com RV, por não estarem ligados diretamente ao domínio do sistema, possuem a possibilidade de serem reutilizados em interfaces de outros sistemas que sejam baseados em ações de acesso a bancos de dados e impressão de documentos. Ressalta-se que grande parte dos sistemas legados comerciais é baseada nessas ações.

O estilo de interação também se reflete nas interfaces com RV geradas no processo de reengenharia. O usuário em um estilo funcional de interação (segmentado) pensa na tarefa a ser realizada antes de escolher o dado que vai utilizar, ou seja, o objeto que sofrerá a ação. No outro estilo, OO, a ordem de raciocínio é inversa, pois o usuário escolhe primeiro o dado (objeto) que vai ser utilizado, depois a ação que esse objeto sofrerá. Isso faz com que, no estilo de interação OO, o usuário cometa menos erros ou enganos, pois ele só pode realizar as tarefas permitidas para aquele tipo de dado/objeto escolhido.

6 Conclusão

As principais contribuições resultantes deste projeto são, como esperado, a confirmação do paradigma de orientação a objetos de desenvolvimento como paradigma mais adequado para apoiar a reengenharia de interfaces para interfaces com RV. A tabela 7 apresenta, de forma resumida, os principais pontos de influência dos paradigmas de implementação no processo de reengenharia para interfaces com RV.

Tabela 7 – Influência dos paradigmas de implementação na reengenharia para interfaces com RV

Características \ Versão	Legado	Segmentado	OO
Modularização	Não	Sim	Sim
Facilidade de Representação	-	Mediana (3 Semanas)	Simples (1 Semana)
Domínio x Ambiente Virtual	Não - Natural	Mediano	Intuitivo
Reuso dos Objetos 3D	Não	Sim	Sim

Em relação à modularização, observou-se que a existência dessa característica na estruturação do código do sistema resulta em maior facilidade de representação das funcionalidades do sistema em sua interface com RV. Como dito anteriormente, a versão legada, por possuir código monolítico, impossibilita a representação de módulos isolados de funcionalidades na interface com RV. Já a versão segmentada, mesmo modularizada, apresenta alguma dificuldade para a representação na interface com RV devido as suas características funcionais de interface, tendo sido necessárias três

semanas para a elaboração de metáforas, duas a mais em relação ao tempo necessário para compor a interface com RV para a versão OO. A versão OO, devido às características inerentes ao paradigma de implementação, facilitou a representação de suas funcionalidade utilizando objetos relacionados ao mundo real, sendo necessária apenas uma semana para criação de sua interface.

A partir da avaliação heurística realizada, observou-se que o domínio do sistema não se refletiu na interface com RV gerada para o sistema legado. A interface com RV gerada para a versão segmentada apresentou muitos objetos que refletem o domínio do sistema porém apresentou também objetos representando metáforas computacionais não relacionados ao domínio. Entretanto, a interface com RV do sistema OO apresenta apenas objetos que refletem o domínio do sistema, tornando-o mais intuitivo.

Foi percebido durante os processos de reengenharia para interfaces com RV das versões segmentada e OO a possibilidade de reuso dos objetos 3D, comprovados com o reuso de diversos objetos relacionados a compras e vendas de peças da versão segmentada na interface da versão legada.

Pode-se inferir, a partir dos resultados obtidos, que a modularização do sistema, facilitada pelo uso do paradigma OO de desenvolvimento e implementação, apresenta vantagens em relação aos outros paradigmas estudados quando o objetivo é realizar a reengenharia de interfaces para interfaces com RV utilizando metáforas do mundo real.

Referências

- [1] Plaisant, C. et al., (1997) “A User interface reengineering: Low effort, high payoff strategies” IEEE Software, vol.14, 4, p. 66-72.
- [2] Nielsen, J. (1993) “Noncommand user interfaces” Communications of the ACM 36, 4, p. 83-99.
- [3] Weiss, P., Jessel, A.S. (2001) “Virtual Reality Applications to Work”, URL: <http://www.infocommons.utoronto.ca/atrc/rd/library/papers/weiss.html>.
- [4] Gabbard J. L. (1997) “A Taxonomy of Usability Characteristics for Virtual Environments”, Masters Thesis. Department of Computer Science, Virginia Tech.
- [5] Kaur K. (1998). “Designing virtual environments for usability”. PhD thesis. Centre for HCI Design, City University, London.
- [6] Assis, A. S. F. R.; Silva, J. A. C. (2000) “Development of Interactive Systems with Virtual Reality Based on User Centered Design” In SCI 2000 - World Multiconference on Systemics, Cybernetics and Informatics, 4, 2000, USA.
- [7] Chikofsky, E.J., Cross II, J.H. (1990) “Reverse Engineering and Design Recovery: A Taxonomy”. IEEE Software, v.7, n.1, pp. 13-17.
- [8] Penteado, R., Braga, R., Masiero, P.C. (1998) “Improving the Quality of Legacy Code by Reverse Engineering” In: International Conference on Information Systems Analysis and Synthesis, ISAS’98, 4, Orlando-Florida. p.364-370.
- [9] Molich, R.; Nielsen, J. (1990). “Improving a human-computer dialogue”, Communications of the ACM 33, 3, p. 338-348.