Delmar Galisi Domingues

O Uso de Metáforas na Computação

**Dissertação apresentada como exigência parcial para
obtenção do título de Mestre em Ciências da
Comunicação à Escola de Comunicações e Artes da
Universidade de São Paulo**

Orientador: Prof. Dr. Martin Grossmann

São Paulo, 2001

**Universidade de São Paulo
Escola de Comunicações e Artes
Curso de Ciências
Área de Ciências da Informação e Documentação**

O Uso de Metáforas na Computação

Delmar Galisi Domingues

**Dissertação apresentada como exigência parcial para
obtenção do título de Mestre em Ciências da
Comunicação à Escola de Comunicações e Artes da
Universidade de São Paulo**

Orientador: Prof. Dr. Martin Grossmann

São Paulo, 2001

Comissão Julgadora

Sumário

AGRADECIMENTOS	6
RESUMO.....	7
ABSTRACT.....	8
INTRODUÇÃO.....	9
A INSERÇÃO DA METÁFORA NAS INTERFACES DE COMPUTADOR	13
1.1 AS FUNÇÕES DA METÁFORA	14
1.2 METÁFORAS E COMPUTADOR.....	16
1.2.1 <i>Metáforas que se tornaram nomes</i>	16
1.2.2 <i>Metáfora-palavra</i>	20
1.2.3 <i>Instruções de computador</i>	20
1.2.4 <i>Computador: níveis de linguagem</i>	21
1.3 UMA EVOLUÇÃO DAS LINGUAGENS COMPUTACIONAIS E SUAS INTERFACES	25
1.3.1 <i>Denominação de instruções a partir de metáforas em interfaces orientadas a linhas de comando</i>	26
1.3.2 <i>Interfaces orientadas a menus</i>	30
1.3.3 <i>A GUI (Graphic User Interface) – A Interface Gráfica</i>	31
1.4 METÁFORAS EM INTERFACES GRÁFICAS	35
1.4.1 <i>Metáfora-enunciado</i>	36
1.4.2 <i>A estrutura da imagem digital</i>	39
1.4.3 <i>Orientação a objetos</i>	42
1.4.4 <i>A escolha de uma metáfora no Espaço de Soluções: a interface e seus objetos</i>	47
1.4.5 <i>A metáfora na interface gráfica</i>	57
1.4.6 <i>A popularização da adoção de metáforas nos computadores</i>	62
1.5 ESTUDO DE CASO: OS SOFTWARES DE AUTORIA	65
1.6 CONCLUSÃO DO CAPÍTULO.....	69
UMA AMOSTRAGEM DO USO DE METÁFORAS NO COMPUTADOR.....	72
2.1 INTRODUÇÃO	73
2.2 A HISTÓRIA DO COMPUTADOR: METÁFORAS, ANALOGIAS E SIMULAÇÕES	74
2.3 FORMAÇÃO DE PALAVRAS NA COMPUTAÇÃO: ALGUMAS METÁFORAS QUE VIRARAM NOMES	78
2.4 <i>HARDWARE E SOFTWARE</i>	81
2.5 OUTRAS PARTES DO <i>HARDWARE</i> E DO <i>SOFTWARE</i>	83
2.6 DIRETÓRIOS, ÁRVORES, GALHOS, RAIZ E AÇÕES ASSOCIADAS	85
2.7 SALVAR, ARMAZENAR, CARREGAR E IMPORTAR DADOS.....	88
2.8 REDES	91
2.9 <i>USER-FRIENDLY</i>	92
2.10 A METÁFORA DO <i>DESKTOP</i>	96
2.11 PASTAS E DOCUMENTOS.....	99
2.12 JANELAS	100
2.13 MENUS.....	102
2.14 BOTÕES.....	103
2.15 ÍCONES.....	104
2.16 RECORTAR-COPIAR-COLAR	105
2.17 PROCESSADOR DE TEXTOS.....	107
2.18 PLANILHA	108
2.19 OUTROS PROGRAMAS.....	109
TENDÊNCIAS RECENTES NO USO DE METÁFORAS: DAS INTERFACES GRÁFICAS À REALIDADE VIRTUAL	111

3.1 INTRODUÇÃO	112
3.2 O AVANÇO TECNOLÓGICO QUE LEVOU À INTERNET	113
3.3 HIPERMÍDIA E NAVEGAÇÃO.....	115
3.3.1 Mapas e metáforas de sistemas hipermídia	118
3.3.2 As metáforas de orientação.....	121
3.3.3 Desorientação na Web.....	123
3.3.4 A metáfora do browser.....	124
3.4 O MUNDO INTERLIGADO: DA METÁFORA DO TAMPO DE ESCRIVANINHA PARA A DO NAVEGADOR	128
3.5 METÁFORAS DE OUTROS MEIOS	130
3.5.1 Interfaces Transparentes	135
3.6 A REALIDADE VIRTUAL.....	138
3.6.1 O Ciberespaço: o mundo como metáfora	144
3.7 A INVERSÃO DO POSTULADO: DA REPRESENTAÇÃO AO OBJETO.....	147
CONCLUSÃO.....	149
REFERÊNCIAS BIBLIOGRÁFICAS	152
BIBLIOGRAFIA	154

Agradecimentos

Ao Prof. Dr. Martin Grossmann: meu especial agradecimento pela dedicação, pela paciência, pela sabedoria e pelo convívio extremamente rico em todos os momentos.

Aos membros da banca de qualificação, Prof. Dr. Julio Plaza e Prof. Dr. Romero Tori, pelas sugestões fundamentais, antes, durante e após a apresentação. Agradeço também à Profa. Dra. Anna Maria Marques Cintra pelo pronto atendimento na solução de algumas dúvidas.

Aos amigos e colaboradores de sempre, Eduardo J. Stefanelli, Maurício Pirillo e Prof. Dr. Alexander J. Romiszowski, pela ajuda que nunca faltou. E à Jane Marques pela força no encaminhamento das traduções.

Finalmente meu agradecimento pessoal e carinhoso à minha parceira de hoje e sempre, Cintia Valente, por ter segurado a barra (da obra, da gravidez e de todo o resto) no decorrer do período.

Resumo

A década de 90 poderá ficar marcada na história das comunicações pelo estabelecimento em definitivo dos meios interativos. Indubitavelmente, o computador foi o maior responsável por este fato. No entanto, esta máquina, que havia se tornado uma importante ferramenta de trabalho nos anos anteriores, ainda tinha uma grande resistência de uso pela maioria das pessoas. O principal motivo alegado era a sua dificuldade de operação e comunicação, em função de sua linguagem nativa. Por conta disso, a interface passou a ser considerada um elemento chave nas pesquisas nesta área; e, conseqüentemente, comunicadores entraram neste processo. Afinal de contas é através da interface que o usuário estabelece sua interação com o sistema operado pelo computador. Ela é um espaço tradutor, uma face intermediária entre a linguagem nativa da máquina e as linguagens naturais do homem. É neste ponto que a metáfora tomou seu lugar. Tropo lingüístico por excelência, a metáfora foi agregada à interface gráfica do computador e tornou-se um dos recursos tradutores de maior expressão neste processo. De meio técnico-científico utilizado no ambiente acadêmico, o computador foi se tornando cada vez mais também um aparelho doméstico. Este trabalho pretende demonstrar o papel que a metáfora teve neste período e o modo como uma figura tradicionalmente lingüística pode se manifestar em várias instâncias de sistemas computacionais e principalmente em interfaces de representação gráfica digital.

Abstract

The '90s may come down into the history of communications as the decade in which interactive media were finally established. Computers were undoubtedly the main driving force behind it. Despite having become an important work tool in previous years, computers still met with great resistance on the part of most people. The primary reason against using computers was that it was not easy to operate and interact with, owing to the source language. As a result, the user interface became a key focus of research in this area; it was then that communication experts were drawn into the process. After all, it is through the interface that users interact with computer-controlled systems. The interface is a translation environment, a transition between the machine's source language and the languages typical of men. It is at this moment that metaphors found their place in this process. A figure of speech by nature, metaphors were added to computer graphic interfaces and became one of the most relevant translation features in this process. From a technical and scientific device used in academic environments, computers were gradually adopted as a household appliance as well. This paper aims to show the role of the metaphor during this period and the way in which what had traditionally been a figure of speech could manifest itself in several instances of system computers and mainly in digital graphic representation interfaces.

Introdução

“A metáfora é talvez uma das mais frutíferas potencialidades humanas. Sua força está próxima da mágica, e ela parece ser uma ferramenta de criação que Deus esqueceu dentro de cada uma de Suas criaturas quando Ele as criou.” (Gasset, 1925)

Falar do “uso de metáforas na computação” pode parecer algo incompatível. Antes de mais nada, o computador é uma máquina absolutamente lógica. Sua linguagem nativa é rígida, matemática e até mesmo estranha para os nossos padrões. O processador, uma espécie de “cérebro eletrônico” da máquina, se “comunica” em binário. Não há margem para a compreensão nas entrelinhas, para os trocadilhos, para a poesia.

A contradição se situa no fato de que os chamados tropos ou figuras de linguagem, na qual a metáfora se enquadra, sejam lembrados pelo caráter poético de suas aplicações. Mas a metáfora, assim como qualquer outro tropo, está em todos os lugares. Fazemos uso dela diariamente, às vezes sem perceber. É parte integral de nossa linguagem. Utilizamos como recurso estilístico, de retórica ou simplesmente para facilitar o ato comunicativo. E embora a metáfora pareça fazer parte de um reduto exclusivo pertencente à linguagem literária, na verdade ela está inserida no processo de representação figurada de qualquer manifestação. Seu uso se tornou tão popular que acabou por formar palavras ou criar conceitos. Com a informática, não poderia ser diferente.

A partir das leituras e reflexões que realizei em seguida, pude perceber que estes dois assuntos – o computador e a metáfora –, que por ora pareciam pertencer a universos tão distantes, possuíam um ponto em comum: o uso da linguagem. De um lado a do computador, absolutamente codificada, construída a partir de uma lógica precisa, de caráter rígido e em determinados momentos até mesmo colocada fisicamente (quando implantada no desenho de seus circuitos). De outro, a linguagem figurada, poética, conotativa, mas também cognoscitiva das figuras de linguagem em geral, e, em particular, das metáforas. Em que ponto e de que modo estariam se comunicando estas duas linguagens?

Mais adiante notei que, além das linguagens citadas, o computador utilizaria tantas outras, como a linguagem verbal, escrita nos seus processadores de texto, ou a linguagem visual da suas telas, só para ficar nas mais óbvias. Neste mundo repleto de linguagens, não só das chamadas linguagens humanas, mas também das linguagens tecnológicas, não é de se estranhar que um conceito tenha adquirido um *status* especial: o de interface.

A definição mais usual de interface diz que ela é uma “face entre” dois dispositivos, físicos ou lógicos, que não poderiam se comunicar diretamente. Mas a interface é também uma espécie de tradutora, pois põe em contato dois sistemas, equipamentos ou “faces” que não se encaixam ou, usando de uma metáfora, não “falam a mesma língua”. No contexto da Informática, a interface “traduz” a linguagem do computador para alguma linguagem que nós, humanos, possamos compreender. Mas o fato inquestionável, e importante para o estudo, é que é exatamente através da interface que o usuário se comunica com a máquina. Não é à toa, portanto, que a interface seja comumente definida simplesmente como a área que vemos na tela do computador e, para alguns, virou sinônimo para isto. Assim, se a interface é um espaço tradutor entre dois níveis de linguagem, podemos constatar que é ali que a metáfora deve estar inserida e que, portanto, é ali que o estudo deve ser focado, já que, como veremos adiante, a metáfora também é um recurso tradutor.

Neste sentido, o **capítulo 1** busca desvendar, a partir da evolução das interfaces, o modo como a metáfora se estabeleceu como tradutora da linguagem da máquina para a linguagem que o usuário compreende, em função do que ele vê na tela do computador. Terão destaque, neste sentido, a definição que se dá para metáfora por alguns autores de relevância, como Aristóteles e Paul Ricoeur, seus primeiros estudos que colocam-na como disciplina da Retórica, sua classificação, suas funções e seu papel como recurso tradutor.

Em seguida, o estudo prossegue examinando a linguagem do computador a partir de sua concepção binária, o desenvolvimento de outras linguagens de maior compreensão para o público leigo em geral e a conseqüente necessidade de criar interfaces com o usuário que disponibilizem a tradução de uma para o outra.

Finalmente, o capítulo abordará como a metáfora se inseriu nos diversos tipos de interface, cuja evolução foi ditada pelo ritmo da absorção das novas tecnologias do próprio computador. De formadora de palavras na informática à signo visual de tradução nas interfaces gráficas, a metáfora esteve presente em todos os períodos desta evolução.

De caráter mais ilustrativo, o **capítulo 2** apresentará uma pequena amostragem da aplicação de metáforas nos computadores: os nomes de dispositivos criados a partir dela, o processo de denominação de comandos de computador, a metáfora do *desktop* nos sistemas operacionais e seus elementos de composição, e o papel da metáfora a partir da adoção de um modelo de interação para certos aplicativos, como o processador de textos e a planilha de cálculo.

O **capítulo 3** abordará as tendências mais recentes do emprego de metáforas em computadores. O principal exemplo advém dos sistemas interativos hipermídia, particularmente da porção WWW da Internet. Relataremos a contribuição dada pelo tropo na formação dos navegadores de Internet e o papel que a metáfora possui quando é aplicada para minimizar a desorientação que o usuário sente ao navegar por sistemas não-lineares típicos da hipermídia. A metáfora será lembrada também pelo seu papel social na compreensão de certos conceitos e tecnologias que surgiram na esteira da rede mundial de computadores. Finalmente colocaremos um olhar no tropo a partir de interfaces baseadas na realidade virtual.

Seguindo a linha desenvolvida pela dissertação, gostaria ainda de ter abordado a adoção de hipermetáforas no computador. Mas senti que isto seria matéria para outra pesquisa, assim como a extensão do estudo para outros tropos, tão possíveis de serem empregados em interfaces computacionais como a metáfora.

— CAPÍTULO 1 —

A inserção da metáfora nas interfaces de computador

1.1 As Funções da metáfora

A metáfora faz parte de nosso dia-a-dia. É parte integral de nossa linguagem. Podemos não perceber, mas fazemos uso dela constantemente. No entanto, embora ela esteja presente nas diversas formas de expressão do homem, ela é geralmente entendida como um recurso de manifestação artística. Nem sempre foi assim.

Os primeiros estudos que se têm notícia sobre a metáfora colocam-na como parte da teoria da composição discursiva, da qual fazem parte também as outras figuras de linguagem (ou tropos). A fundamentação para essas figuras é proporcionada pela Retórica, conforme formulada por Aristóteles.¹

No dizer de Paul Ricoeur, “a retórica foi adicionada como uma ‘técnica’ à eloquência natural.” [Ricoeur, 1975:15] e sua função é a de engendrar a persuasão. O que a coloca como um domínio do discurso. Como os gregos tinham na expressão oral a sua forma de comunicação mais importante, a metáfora (e os outros tropos) era colocada também como um recurso da oralidade.

No entanto, por mais ampla que seja a extensão da função retórica, ela não abrange todos os usos do discurso. A arte do “bem-falar” exige outros domínios. Por exemplo, a poética.

“A poesia não é eloquência (...) Poesia e eloquência delineiam dois universos de discurso distintos. (...) Ora, a poesia não pretende provar nada; o seu projeto é mimético; (...) o seu objetivo é o de compor uma representação mais essencial das ações humanas; o seu modo próprio é o de dizer a verdade através da ficção, da fábula, do *muthos* trágico. A tríade *poiêsis-mimêsis-catharsis* descreve de maneira exclusiva o mundo da poesia, sem confusão possível com a tríade *retórica-prova-persuasão*” [idem, 19-20].

A base desta divisão foi lançada por Aristóteles, que afirma que a *léxis* na *Retórica* tem a função da prova, da demonstração, da função lógica, objetiva, intelectual, denotativa. A *léxis* na *Poética* tem a função da *mimesis* (imitação), a função alógica, subjetiva, emocional, a função da conotação. Por isso, o estudo da metáfora, engendrado por Aristóteles, abrange

¹ Aristóteles dividiu a Retórica em três campos: uma teoria da argumentação, uma teoria da elocução e uma teoria da composição do discurso. O estudo dos tropos era englobado dentro desta terceira teoria.

justamente estes dois domínios do discurso: a poética e a retórica. Para Aristóteles, a metáfora tem uma estrutura, mas duas funções. Mais tarde este conceito é ampliado por Turbayne:

“a função retórica representa a área do saber, do conhecimento, da ciência e é expressa pela linguagem unívoca, nocional, referencial e numa classificação de Le Guern, denotativa. A função poética representa a área do sentir, da paixão, se processa no campo emotivo numa linguagem plurívoca, polivalente, socorrida por associações extranocionais, portanto conotativa.” [Filipak, 1983:68]

Monroe Beardsley define a linguagem denotativa como de significação primária e a conotativa com de significação secundária. A primeira é cognoscitiva, a outra é emocional. Para ele, há implicitamente a idéia dual de denotação/noção e conotação/emoção. “A denotação propõe uma significação explícita e a conotação sugere uma significação implícita.” [idem ,78] Jean Cohen conclui: “Nem toda a metáfora é poética.” [Cohen, 1974:172]

O grande paradoxo, no entanto, é que a metáfora “chegou até nós através de uma disciplina – a *retórica* – que pereceu nos meados do século XIX” [Ricoeur, 1975:13]. Modernamente, a metáfora é colocada como parte da teoria das figuras do discurso, portanto separada da Retórica. Nos bancos escolares, os tropos estão atrelados à disciplina “Comunicação e Expressão”, principalmente como recurso à arte do “bem -escrever”. Portanto, há um evidente deslocamento do uso da metáfora de recurso retórico-oral para poético-visual. Não que a metáfora não fosse utilizada antes também na expressão artística. E não que hoje ela não seja utilizada na nossa fala. Como afirmamos anteriormente, muitas vezes fazemos uso dela sem perceber. Mas, hoje, ao perguntar a alguém à quem serve a metáfora, provavelmente ele responderá que ela está associada à poética e não à retórica.

Se, por um lado, pode haver saudosistas que sonham em ver a teoria das figuras discursivas novamente como parte da Retórica, há, por outro lado, um saldo positivo desta separação. As metáforas podem ser usadas mais livremente em outros domínios, pois não estão atadas ao estudo da persuasão. Elas podem ser aplicadas em outras expressões; não só na oratória. É o que vem acontecendo, por exemplo, no caso da Informática.

1.2 Metáforas e computador

Além de ser uma ferramenta, o computador é também um meio de comunicação e expressão. Tem, portanto, a sua própria linguagem nativa, que é bastante estranha à compreensão humana. Para que haja uma comunicação efetiva homem-máquina, a linguagem do computador deve ser preferencialmente “substituída” para a nossa linguagem natural.

Julio Plaza diz que “toda operação de substituição é, por natureza, uma operação de tradução – um signo se traduz em outro.” [Plaza, 1987:17] A metáfora, portanto, só poderia estar presente neste processo, porque nela está embutida também uma operação de substituição: “a metáfora é o emprego de todo termo substituído por um outro que lhe é assimilado.” [Dubois, 1991:411] Neste sentido, o uso da metáfora é uma espécie de tradução.

Evidentemente, em todo processo de tradução está implícito a necessidade de facilitar um ato comunicativo. Traduzimos porque não compreendemos uma palavra, um signo, um código ou mesmo toda uma linguagem. Não importa em que nível vai se efetivar o processo tradutório (intralingual, interlingual ou intersemiótico); ao traduzir, o usuário comum procura fazer a substituição de um signo por outro “no qual ele se ache desenvolvido de modo mais completo” [Jakobson, 1969:64], como afirmou Jakobson, citando Peirce.

No contexto deste estudo, a metáfora participa como tradutora de um mundo hostil representado pela terminologia e pela linguagem do computador para a nossa linguagem natural. Neste processo de transcodificação da linguagem do computador para a nossa linguagem, veremos que a metáfora participou sob diversas formas. A mais usual foi no processo de denominação e formação de palavras na área da informática.

1.2.1 Metáforas que se tornaram nomes

Jean Dubois no seu “Dicionário de Linguística” define a metáfora como “o emprego de todo termo substituído por um outro que lhe é assimilado após a supressão das palavras que

introduzem a comparação”. O conceito de metáfora como agente de substituição por um termo similar tem sido uma das idéias mais fortemente propagadas.

Para a Retórica clássica, do mesmo modo, a metáfora é descrita em termos de desvio. “Chamemos de metáfora a todo o “deslocamento” (*shift*) do sentido literal para o sentido figurado.” [Sacks, 1992:280] Muitas vezes, este desvio é atribuído à função de denominação. Ou seja, para identificar uma coisa (ou idéia), em vez de utilizar o seu próprio nome, podemos utilizar uma palavra emprestada, uma palavra “estranha”, conforme a terminologia de Aristóteles. Segundo Ricoeur,

“A análise racional dessa transferência de nomenclatura foi entendida como similaridade objetiva entre as próprias coisas ou similaridade subjetiva entre as atitudes relacionadas à compreensão dessas coisas. No que diz respeito ao objetivo dessa transferência, ela deveria preencher uma lacuna lexical e, portanto, servir ao princípio da economia que governa a tentativa de dar nomes adequados às coisas novas, novas idéias, ou novas experiências, ou reelaborar o discurso, e portanto cumprir a finalidade principal do discurso retórico, que é a de persuadir e agradar.” [idem, 147]

Aristóteles afirma que “o dom de elaborar boas metáforas depende da capacidade de ponderar sobre semelhanças.” A isto Paul Ricoeur acrescenta: “a clareza de boas metáforas resulta de sua capacidade de ‘colocar frente aos olhos’ o sentido por ele exposto.” [idem, 146] Donald Davidson criou a expressão “verdade metafórica” para descrever este processo: o simples fato de compreendermos uma metáfora é suficiente para considerarmos esta metáfora como “boa”. [idem, 47]

Esta capacidade de elaborar boas metáforas é muito importante, pois, como dissemos, a metáfora funciona como um elemento de tradução. Julio Plaza afirma: “toda tradução movimenta-se entre identidades e diferenças” [Plaza, 1987:29]. É essencial que uma metáfora objetive o máximo de identidade para cumprir o papel de suprir certas deficiências vocabulares. Esta função da metáfora, calcada no domínio da Retórica ou da denotação, é comumente chamada metáfora *denominativa* ou *lingüística*.

Denominação é um processo de lexicalização. É, pura e simplesmente, a substituição lingüística de uma coisa ou idéia por um termo novo. Neste sentido, a lacuna lexical pode ser preenchida por uma metáfora. O uso contínuo desta metáfora pode torná-la, enfim, uma

denominação de fato. Quando isto acontece, o significado metafórico do termo passa a ser identificado também como um significado denotativo; às vezes tanto quanto o seu significado originalmente estabelecido. Em geral isto acontece quando o termo empregado já aparece no dicionário com ambos os significados: o “de fato” (original) e o “ampliado” (ou figurado). Neste caso, com o tempo, esta metáfora perde força, pois como diz Ricoeur, não há metáfora no dicionário, só há no discurso: “sua construção pode ser repetida e pode ser tomada como uma criação lingüística. Se for adotada por uma comunidade lingüística, pode transformar-se numa significação usual e aumentar a polissemia das entidades lexicais (...) Quando a metáfora chega a este estágio, já não é mais metáfora viva, mas metáfora morta.” [Sacks, 1992:151]

Levando em conta este argumento, certamente, uma metáfora tem um ciclo de vida e em casos extremos ela pode morrer. Charles Bally diz que “muitas metáforas perderam o seu vigor primitivo, elas não são mais reconhecíveis como transposições e que para o intelecto elas estão mortas.” [Filipak, 1984:39]. Alguns, como Dubois, as chamam de metáforas gastas: “a metáfora desempenha um grande papel na criação léxica; muitos sentidos figurados são apenas metáforas gastas” [Dubois, 1991:411]. Já para alguns autores, elas são simplesmente identificadas como catacreses, um tropo que se refere justamente aos termos figurados que são aplicados na falta de um termo próprio, mas que com o tempo passam a fazer parte da linguagem formal: “no caso da existência da palavra própria, a substituição é opcional e temos um tropo, *stricto sensu*; no caso de carência de palavra própria, a substituição é obrigatória, temos uma catacrese.” [Filipak, 1983:27]

Em casos mais extremos, a origem da metáfora se perde completamente. Há exemplos de sobra. Quando chamamos um indivíduo de burro, nem sempre associamos a ofensa ao animal. Sabemos que o adjetivo “burro” tem sua origem no substantivo “burro”. Mas esta correlação está praticamente esvaziada, mesmo porque muita gente que usa a palavra na sua forma adjetiva nunca viu o correspondente animal na sua forma substantiva, embora saiba que ele exista. No entanto, nem sabe se um burro é “burro” mesmo.

A informática é um caso exemplar do emprego de metáforas denominativas. Nenhuma indústria, nenhum campo de atuação do homem tem sofrido atualmente um processo tão

vertiginoso de desenvolvimento. Diariamente surgem novos dispositivos, novos programas, novas ações relacionadas a seu campo de atuação. A necessidade de se criar novos termos para tantas inovações faz-se imediata.

Mas não foi só a velocidade da indústria computacional o fator preponderante da utilização da metáfora denominativa no processo de criação lexical da informática. Temos que considerar também a intenção de substituir um termo técnico, frio, por algo próximo da realidade do público leigo. Se fôssemos considerar a clássica divisão aristotélica das funções da metáfora, teríamos presente a função retórica, pois afinal de contas é um processo de denominação; e num nível menor, teríamos a função poética. São metáforas principalmente cognoscitivas, denotativas. Mas há algo de poético quando chamamos de *mouse* o dispositivo de apontamento; é até mesmo lúdico. Persiste, no entanto, a sua intenção principal, que é a de obter uma comunicação mais efetiva: de um mundo técnico dos computadores para o universo comunicativo de nosso dia-a-dia.

A metáfora entrou fortemente neste processo. Termos computacionais como memória, disco, *mouse*, diretório, arquivo, raiz, árvore etc foram criados a partir de metáforas (veja no capítulo 2 uma descrição mais pormenorizada). Neste sentido, dispositivo de armazenamento tornou-se memória²; dispositivo de apontamento, *mouse*; a estrutura de diretórios, árvore; e assim por diante.

Em todos estes casos, há a simples substituição de um termo por outro. Não importa o contexto ou a estrutura em que a metáfora é colocada. Há basicamente uma troca: de um termo, “dispositivo de armazenamento,” por outro, “memória”. É a metáfora na sua concepção mais simples, em que há o desvio, a transferência ou a substituição de um termo por outro. Esta maneira de empregar a metáfora, na qual está inclusa também a metáfora denominativa, é colocada dentro de uma categoria maior chamada metáfora-palavra.

² Memória é a parte do computador que guarda as informações. É um dispositivo de armazenamento

1.2.2 Metáfora-palavra

A metáfora-palavra, fundamentada sobre bases lingüísticas, consagra o elo entre a metáfora e a palavra. Ela funciona no esquema *isto é aquilo*. O que vale é a tese de substituição, que se faz notar, segundo Davidson, ‘por existir certa semelhança, freqüentemente uma semelhança nova ou surpreendente, entre duas ou mais coisas.’ [idem, 37] Filipak diz que a ‘metáfora-palavra constrói-se pela teoria substitutiva de paradigmas;(...) uma transposição de um nome para um nome estranho, isto é, de uma outra área semântica. Esta transferência para outra área sêmica provoca uma impertinência.’ [idem, 46] Quando falamos que o computador tem memória, isto em primeira mão pode provocar um estranhamento; é impertinente. Mas se esta impertinência for apreendida na sua intenção, ela torna-se uma verdade metafórica.

A informática tem sido feliz neste sentido. São muitos os casos de termos computacionais criados a partir de metáforas-palavra. Particularmente no caso de dispositivos e conceitos, como memória ou arquivo. No entanto, o processo de formação de palavras computacionais vai além. Existe um caso particular em que a denominação não se estabelece por uma aplicação clássica da metáfora-palavra. É o caso da denominação das instruções de computador.

1.2.3 Instruções de computador

O conceito de instrução é fundamental em sistemas computacionais porque efetivamente é através dele que se efetua a comunicação homem-máquina. Como o computador é uma máquina que não tem iniciativa própria, tudo que quisermos que ele faça tem que ser solicitado. Ou seja, o computador precisa ser instruído. Além disso, a autonomia intelectual da máquina é extremamente limitada se comparada com a nossa. Conseguimos entender certas mensagens nas entrelinhas. O computador não tem esta capacidade. Se desejarmos que o computador faça algo, devemos usar instruções já especificadas para a realização da operação correspondente, numa seqüência lógica e coerente. Como o computador só entende instruções, e elas devem ser solicitadas passo a passo, esta lógica é procedimental.

Portanto, para que não haja erros, o ideal é que a comunicação homem-máquina seja “programada”.

Como diz Tanenbaum, computadores são máquinas capazes “de solucionar problemas através da execução de instruções que lhes são fornecidas.” Um programa portanto é “a seqüência destas instruções que descreve como executar determinada tarefa.” [Tanenbaum, 1992:1]

Não bastasse a rigidez deste diálogo, o computador se comunica por uma linguagem diferente, bastante estranha e portanto difícil para a nossa compreensão. Para que a comunicação se realize, é necessário que haja um processo de tradução. Vamos então fazer um pequeno parêntesis a fim de conhecer os fundamentos que regem a linguagem computacional.

1.2.4 Computador: níveis de linguagem

Segundo Lévy. “o principal problema do diálogo com os computadores reside na diferença entre as linguagens formais, que regem o comportamento das máquinas, e as linguagens naturais utilizadas e compreendidas pelo homem em sua vida quotidiana.” [Lévy, 1998:30].

Este diálogo, em princípio, não pode ser imediato porque o código que rege a formação lingüística do computador é diferente do nosso. Assim posto, de duas, uma: ou o computador fala a nossa língua, ou nós falamos a dele. Historicamente, a segunda tentativa foi a adotada, pois a fabricação de dispositivos físicos análogos à nossa representação numérica e lingüística tornaram-se impeditivos em termos de precisão e custos. Assim, só nos restou a alternativa de estabelecer um diálogo com a máquina através do que seria a sua linguagem natural. Mas qual é a linguagem do computador?

Computadores são máquinas eletrônicas. Trabalham com circuitos digitais. Por conta disto, os computadores são projetados para operar somente em dois estados. Ou seja, na linha de carga elétrica, são utilizados dois pontos não-adjacentes, tipicamente chamados como saturação e corte. A carga de entrada em um circuito é de saturação (um nível alto de

voltagem) ou de corte (um nível baixo de voltagem). Como resultado, a tensão de saída também só possui dois estados: alto ou baixo.

Estes dois estados físicos, perceptíveis pelo computador, foram traduzidos simbolicamente por dois valores: zero (0) e um (1). Se os dez dedos do homem influenciaram na formação da linguagem matemática a ponto de torná-la decimal, e o número de letras utilizadas em cada alfabeto é determinado pela capacidade fonética cultural de cada povo; o computador também teve seu código estabelecido pela sua limitação física e portanto só poderia ser binário³. Para nós, acostumados com uma notação numérica que possui dez símbolos, ou com uma linguagem verbal/escrita composta por tantos caracteres, esta linguagem composta somente por seqüências binárias é no mínimo esdrúxula. Para o computador, por outro lado, é uma linguagem natural.

Programas são então construídos com enormes cadeias de zeros e uns. No entanto, a tentativa de efetuar o diálogo homem-máquina através da notação binária tornou-se rapidamente cansativa, porque, além desta linguagem ser para nós pouco convencional, a complexidade crescente do computador foi tornando os programas binários cada vez maiores e mais complexos de se entender. Isto colocou uma pá de cal na linguagem binária; pelo menos para nós. Não para o computador, obviamente, pois em última instância, é esta linguagem, a binária, que o processador vai continuar utilizando, já que sua concepção física-eletrônica continua a mesma. Para nós, no entanto, começaram a ser desenvolvidas novas linguagens, mais próximas de nossa compreensão. Evidentemente, antes de serem executadas pelo processador, estas linguagens são traduzidas para a linguagem binária. A primeira delas foi chamada de Assembly.

```
mov BX,1
mov CX, ShowLength
lea DX, Showit
mov AH,40H
int 21H
ENDM
```

Infelizmente a linguagem Assembly também não é tão fácil de ser compreendida por nós. O problema reside no fato de que esta linguagem possui uma relação 1:1 com a linguagem binária da máquina. Ou seja, embora a linguagem Assembly deixe de

³ Na verdade o computador não é naturalmente binário. Já se pensou na possibilidade de se criar dispositivos que trabalhassem com níveis de voltagem tão diferenciados que pudessem representar muito além de zeros e uns. No entanto, verificou-se que a implementação prática de um dispositivo deste tipo seria inconveniente em termos de precisão e custos.

empregar o código binário, suas instruções são ainda as mesmas da linguagem binária do computador. A única diferença é que em vez de a instrução ser representada por uma sequência binária, ela é representada por um nome-fantasia. Neste sentido, estas instruções têm mais a ver com a realidade lógica do computador do que com a nossa. Por exemplo, as instruções JMP (*Uncondition Jump* ou Salto incondicional), POP (*Pop top of stack* ou tirar do topo da pilha), DEC (*Decrement operand* ou Operação de decremento) fazem muito sentido para o computador ou, na melhor das hipóteses, para um programador⁴ muito experimentado. Mas não para o usuário leigo.

Segundo Tanenbaum, este problema pode ser atacado de duas maneiras principais,

“ambas envolvendo o projeto de um novo conjunto de instruções de uso mais conveniente para as pessoas do que o conjunto de instruções embutidas da máquina. Juntas, estas novas instruções formam uma linguagem que chamaremos de L2, exatamente como as instruções embutidas de máquina formam uma linguagem, que chamaremos de L1.” [idem, ibidem]

Como o computador, em última instância, só entende a sua própria linguagem binária de máquina, a L1, há duas técnicas para executar o programa escrito em linguagem L2. A primeira técnica consiste em substituir cada instrução de L2 pela correspondente sequência de instruções em L1. É gerado portanto um novo programa, escrito completamente com instruções em L1.

Esta técnica é chamada tradução (ou compilação). A outra técnica consiste em “escrever um programa em L1 que receba os programas escritos em L2 como dados de entrada e efetue a execução examinando uma instrução de cada vez e executando a sequência equivalente de instruções L1 diretamente.” [idem, ibidem]. Esta técnica é chamada interpretação. No entanto, tanto o processo de tradução quanto o de interpretação de uma linguagem para outra é realizada pela máquina. Num certo sentido, é o computador que passou a falar a nossa língua.

⁴ Programador é a pessoa que constrói os programas, usando para isto uma das linguagens do computador.

Na maioria das vezes, no entanto, mesmo a linguagem L2 ainda está muito longe de ser compreendida pelo usuário menos experiente. A solução óbvia é criar uma nova linguagem (L3) que abranja um outro conjunto de instruções ainda mais dirigido ao homem. Esta linguagem, obviamente, terá que ser também traduzida ou interpretada para a máquina. Assim, caso esta linguagem tenha sido criada a partir de L2, é necessário que cada instrução L3 seja traduzida pelo conjunto correspondente de instruções L2, e cada uma destas seja finalmente traduzida pelo conjunto equivalente de instruções L1. A criação de novas linguagens pode continuar ininterruptamente, sempre que for necessário se introduzir uma linguagem mais adequada para um determinado público ou para um conjunto de tarefas específicas.

Cada linguagem tem uma predecessora como base. Em geral, são classificadas pelo seu nível de dificuldade: da mais baixa, a linguagem de máquina do computador, às mais altas. A linguagem de máquina é de baixo nível porque ela é uma linguagem de baixa inteligibilidade para o homem.

Meirelles vai bastante longe nesta classificação, pois inclui também a linguagem do homem. “De um lado, no mais alto nível de comunicação, o homem (usuário), que fala sua linguagem natural ou corrente. De outro lado, está o computador, que só entende 0 e 1.” [Meirelles, 1994:264]

Tanenbaum diz que em vez de pensarmos em termos de tradução e interpretação, seria mais “adequado imaginar a existência de um computador hipotético ou máquina virtual” [Tanenbaum, 1992:2], cuja respectiva linguagem poderia ser a L2, a L3 ou a Ln. Tomamos como partida a existência da linguagem binária do computador, a L1, que é a linguagem da máquina real⁵. A partir de cada linguagem criada acima de L1, teríamos o correspondente a um computador hipotético. Neste sentido, quando uma linguagem é formada, paralelamente é estabelecida uma máquina virtual. “Cada máquina tem sua linguagem de máquina que

⁵ Na realidade, Tanenbaum diz que há ainda outros níveis abaixo de L1. O nível 0, lá embaixo, é o hardware. Os seus circuitos é que executam os programas que foram escritos em L1. E abaixo do nível 0 se situaria no campo da engenharia elétrica e é denominado nível de dispositivos, representado por transistores individuais. Finalmente poderia se questionar como funcionam os transistores por dentro, matéria que segundo Tanenbaum já pertence à física do estado sólido.

consiste em todas as instruções que a máquina pode executar. De fato, uma máquina define uma linguagem. Do mesmo modo, uma linguagem define uma máquina.”[idem, 3].

1.3 Uma evolução das linguagens computacionais e suas interfaces

A evolução tecnológica do computador multiplicou as tarefas que a máquina pode desempenhar e, concomitantemente, provocou o aumento do número de linguagens existentes. A própria noção de linguagem computacional se atualizou, pois não é mais possível reduzi-la somente à sua concepção tradicional, que a identifica como um conjunto de instruções que são codificadas em um editor de textos comum para posteriormente serem compiladas por um programa tradutor. Hoje, além das linguagens orientadas à codificação escrita, existem as linguagens visuais, cuja comunicação é orientada a cliques em menus e ícones, operações do tipo arrastar e outros tipos de interações, todas estas já pré-programadas. Há também muitas linguagens híbridas, em parte visuais, mas com algumas operações que devem ser codificadas pela escrita. Em razão disto, fica mesmo muito difícil definir o que é uma linguagem computacional, o que é um aplicativo, um programa ou um software. Numa definição mais aberta, poderíamos dizer que tudo é linguagem de computador, de uma aplicativo para tratamento de imagens ou um processador de textos, até uma linguagem tradicional, do tipo Pascal ou Cobol. De qualquer modo, esta concepção não invalida a tese de Tanenbaum, que diz que toda linguagem define um computador hipotético ou uma máquina virtual.

O fato de Tanenbaum chamar estas máquinas de virtuais não implica, no entanto, a inexistência de uma representação. As máquinas virtuais ou suas linguagens são representadas através de suas interfaces. Lexicalmente falando, interface significa a “face entre”. Aplicando ao modelo computacional, a interface é a face mediadora entre a linguagem do homem e a da máquina. Neste sentido, tem um papel fundamental no processo de tradução.

Assim como o computador, a interface teve o seu processo evolutivo, definido paralelamente pela evolução da própria máquina e de suas linguagens. As primeiras interfaces eram orientadas por linhas de comando. Os seus programas eram procedimentais

e tinham como objetivo a execução de tarefas simples, com o cálculo da trajetória de projéteis para o esforço de guerra, a tabulação de dados para um censo, a definição da média e o registro da nota de um aluno etc. Na tela, o usuário estabelecia sua comunicação através de uma palavra-chave ou de um comando, e conseqüentemente a interface era composta por muitas linhas de texto. Nesta conjuntura, foram desenvolvidas algumas linguagens de computador que tinham como objetivo criar programas específicos: ou de ordem comercial (por exemplo, controle de estoque), ou de ordem científica, ou educativa etc. Cada nova linguagem deveria criar suas instruções próprias, em geral combinações de instruções primitivas de máquina, mas que, de qualquer forma, deveriam passar por um processo de denominação.

1.3.1 Denominação de instruções a partir de metáforas em interfaces orientadas a linhas de comando

O projeto de criação de uma nova linguagem de computador de nível mais alto implica na denominação de suas instruções. A denominação de uma instrução não é somente uma referência

decorativa ou uma identificação simbólica. É também a partir do nome criado que o usuário irá “chamar” a instrução, ou, tecnicamente falando, executar a instrução. Nas antigas interfaces orientadas a linhas de comando isto era bem comum. Por exemplo, no MS-DOS⁶, o antigo sistema operacional⁷ da Microsoft, a operação de cópia de arquivos é feita a partir do comando⁸ COPY. Mas COPY é também o nome da instrução.

⁶ MS-DOS ou MicroSoft Disk Operating System é um sistema operacional criado através da linguagem BASIC, em 1981, pela Microsoft. Obteve grande popularidade na década de 80, mas hoje o DOS (como é conhecido popularmente) é considerado muito pouco amigável. Para operar este sistema operacional, o usuário necessita decorar um grande número de comandos, que são digitados diretamente na tela do computador. Através deles é possível criar diretórios, copiar e apagar arquivos ou formatar discos, dentre outras operações.

⁷ Tanenbaum em seu livro *Sistemas Operacionais Modernos* fornece uma boa descrição para sistema operacional. Segundo o autor “há muitos anos, ficou claro que dever-se-ia encontrar uma maneira de afastar o usuário da complexidade do hardware. A forma encontrada, que evoluiu gradativamente, foi a de colocar uma camada de software em cima do hardware, para gerenciar todos os componentes do sistema, apresentando-o ao usuário como uma interface muito simples de entender e de programar. Tal interface é conhecida como máquina virtual. A camada de software construída sobre o hardware nada mais é do que o sistema operacional.” O autor diz ainda que o sistema operacional é portanto um software supervisor, com o objetivo

O nome da instrução deve de algum modo expressar o significado técnico que ela apresenta no contexto da linguagem a partir da qual ela foi formada. Mas também deve identificar o resultado simbólico que nós, humanos, percebemos da operação. Esta percepção é dada dentro do âmbito da máquina virtual. De um certo modo, portanto, esta instrução tem dois significados: um para a máquina real, que é técnico, e outra para a máquina virtual, que é figurativo.

A experiência tem mostrado que é mais adequado identificar uma nova instrução pelo significado que ela tem na sua máquina virtual. Não seria possível e na realidade seria mesmo inadequado denominar uma instrução de linguagem de alto nível por uma das instruções de L_{n-1} , a linguagem utilizada para sua criação. Para entender melhor, vamos fazer uma comparação com a receita de um bolo.

Assim como programas de computador, receitas de bolo são procedimentais. Quando numa receita lemos uma instrução do tipo “Ferver a água”, sabemos de antemão que ela implica em uma série de instruções básicas, como colocar a panela com água no fogão, riscar um fósforo, ligar o gás, aproximar o palito de fósforo até a boca do fogão correspondente. Nenhuma receita, obviamente, descreve esta operação – “ferver a água” – com todos estes detalhes. Subtende-se que o “cozinheiro” saiba que “ferver a água” implica nesta série de instruções básicas, porque o “processador” interno do homem já tem estes passos internalizados.

No caso do computador acontece a mesma coisa. Estes passos básicos da receita (colocar a panela com água no fogão, riscar um fósforo, ligar o gás etc) correspondem às instruções L_1 , necessárias para o processador do computador, mas enfadonhas para os homens.

De qualquer modo, uma instrução de computador L_n , que é o resultado (ou a soma) de instruções de L_{n-1} , precisa ser referenciada. No caso da receita acima citada, cumprimos este papel quando denominamos a instrução como “ferver a água”. Neste mesmo exemplo, não poderíamos chamá-la de “riscar o fósforo” (que é como se fosse uma das instruções

também de proteger o hardware da ação dos usuários. Não é à toa que alguns autores como Meirelles comparam o sistema operacional com um guarda de trânsito. [Tanenbaum, 1995:3]

Ln-1 que foi usada para formá-la), pois este nome não descreveria a operação tão bem quanto faz a instrução “ferver a água”. No computador, também não deveríamos usar uma das instruções da linguagem Ln-1 como referência para chamar a operação da instrução de Ln. As instruções de linguagens de alto nível, em geral, não são referenciadas por uma das instruções da linguagem que as formaram. Elas são denominadas a partir da metáfora da operação que é realizada no ambiente de sua máquina virtual.

Há ocasiões, no entanto, em que o nome da instrução de algum modo identifica a operação que ela realiza tanto na máquina real (ou na máquina virtual predecessora) quanto na virtual. Por exemplo, o comando *copy* do DOS descreve uma ação de cópia, tanto do ponto de vista da linguagem Ln quanto do da linguagem Ln-1. Nos dois casos, o comando *copy* faz os bytes que ocupam um setor do disco rígido⁹ serem duplicados com a finalidade de ocupar também outro setor.

Por outro lado, o comando *delete*, também do DOS, é referenciado por um nome que representa bem a operação que é realizada nesta máquina virtual, mas não no âmbito da linguagem a partir da qual ele foi formado. *Delete* significa apagar. No contexto da máquina virtual representada pelo DOS é o que acontece: estamos apagando um arquivo. Mas no da máquina correspondente à linguagem Ln-1, o comando *delete* não apaga arquivo algum. Ele apenas deixa o espaço de um disco rígido, que estava sendo utilizado por um arquivo, à disposição para que possa ser ocupado por um arquivo novo. Portanto, os bytes do arquivo “velho” continuam a ocupar o espaço do disco rígido, mas não com o mesmo *status*. Eles ficam lá até que sejam utilizados por outro. *Delete* é apenas uma metáfora para a operação que acontece na máquina virtual representada pelo DOS. No capítulo seguinte, veremos que há uma série de nomes de instruções, em geral chamados “comandos” nas linguagens de alto nível, que foram criados a partir de metáforas.

⁸ Nas linguagens de programação de alto nível, em geral, as instruções são chamadas de comandos.

⁹ Disco rígido é um dispositivo de memória permanente do computador. Isto significa que quando os dados são guardados nele, eles não são perdidos quando o computador é desligado, ao contrário das chamadas memórias temporárias. Isto significa que os dados que estão guardados no disco rígido podem ser recuperados posteriormente. Por outro lado, a palavra “rígido” é acrescentada à disco para diferenciar os discos rígidos dos discos flexíveis, que também são permanentes. Um exemplo de disco flexível é o disquete.

Assim, no caso de um conjunto de instruções de L_{n-1} resultar na criação de em uma nova instrução de uma linguagem de um nível mais alto, L_n , a metáfora denominativa surge como uma variante. Neste caso, não temos simplesmente a substituição de um termo por outro, mas de um conjunto de termos (ou de instruções) por um outro termo – a nova instrução. Ou seja, há neste caso a tradução de um conjunto de instruções de uma linguagem de mais baixo nível para uma instrução de linguagem de mais alto nível.

Mas podemos enxergar esta tradução também como um processo de substituição do tipo 1:1, se considerarmos que o conjunto de instruções de L_{n-1} representa, na verdade, “uma” operação da máquina virtual de L_n . Neste caso, o esquema de substituição do tipo *isto é aquilo*, que identifica a metáfora-palavra, volta a ser validado, pois é efetuada a tradução de uma operação por uma nova instrução. *Isto* é o nome da instrução. *Aquilo* é a operação na máquina virtual. *Delete* (apagar) é a instrução de DOS (a máquina virtual que corresponde à L_n) que “chamamos” quando desejamos realizar a operação de L_{n-1} correspondente a “deixar o espaço ocupado por um arquivo à disposição para ser ocupado por outro”.

Um outro exemplo é o comando chamado *move*, também do MS-DOS. O comando *move* foi criado a partir de um conjunto de instruções da linguagem BASIC. *Move* em inglês significa mover. Mas no contexto da máquina virtual abaixo do DOS, a do Basic, tecnicamente, o comando *move* faz outra coisa. Ele define uma operação que atribui um outro diretório¹⁰ à entrada de um arquivo. Por outro lado, quando tiramos a entrada de diretório de um arquivo e atribuímos a outro diretório, no contexto do DOS enxergamos outra coisa. Temos a impressão que movemos este arquivo de um lugar para outro de um disco. Neste sentido, mover é uma metáfora, pois define uma operação que é executada somente na máquina virtual representada pelo DOS, mas não na do Basic. Neste nível, que está abaixo, os bits do arquivo “movido” não se movem; eles permanecem onde estão no disco.

¹⁰ O diretório é uma espécie de repositório que agrupa um conjunto de arquivos. Tecnicamente, é um arquivo especial que possui algumas informações de caráter gerencial, como a entrada e saída de arquivos presentes dentro dele. Veja mais detalhes adiante no próximo capítulo.

1.3.2 Interfaces orientadas a menus

Interfaces orientadas por linhas de comandos têm suas vantagens: alta velocidade de processamento, utilização de um espaço mínimo na tela do computador e rapidez de interação, principalmente para usuários experientes, já que é baseada na inserção de dados por digitação. Mas as desvantagens são evidentes. É necessário um conhecimento prévio para operá-las, principalmente se orientadas por comandos. Normalmente o usuário precisa decorá-los. A estrutura interfacial não é visível. O nível de abstração é muito alto. E apesar da utilização de metáforas no processo de denominação de algumas instruções, alguns comandos não fazem sentido para o usuário mais leigo.

O primeiro avanço para tornar as interfaces mais amigáveis foi através da utilização de menus. Um menu interfacial de computador é simplesmente uma lista de opções mostrada na tela para que o usuário faça uma seleção. Com os menus, o usuário pode “reconhecer o que deseja fazer.”

[Típica interface orientada a menus]

[Heckel, 1993:60'] Outra vantagem da utilização dos menus é que estamos familiarizados com o seu uso, pois eles são metáforas dos cardápios de restaurantes. A idéia que está por trás é que se sabemos consultar um cardápio, conseguimos também interagir com os menus do computador.

Os menus computacionais também passaram por evoluções (veja capítulo seguinte): começaram como menus de tela cheia e mais tarde foram adaptados a barras de menu e menus suspensos. Assim como o seu similar gastronômico, todos estes tipos de menus computacionais procuram diminuir a sobrecarga cognitiva do usuário, porque dispensam-no de aprender certas ações e comandos que são necessários na comunicação que se realiza numa linguagem de computador convencional. Além disso, sua organização é baseada em tópicos e subtópicos hierarquizados que permitem ao usuário não se sentir perdido numa estrutura de procedimentos que nem sempre é linear.

A interface baseada em menus foi apenas uma transição entre as interfaces orientadas por linhas de comando e as interfaces gráficas atuais. A evolução tecnológica dos computadores pessoais ajudou a ditar este ritmo. Os primeiros monitores eram monocromáticos e de baixa resolução, configuração suficiente para apresentar somente uma interface baseada em linhas de comando. Depois surgiram telas de computador que mostravam somente algumas cores, mas que possibilitavam a construção de grafismos mais elementares. Isto tornou possível desenvolver interfaces orientadas por menus. Quando os monitores passaram a apresentar milhares de cores e os primeiros dispositivos de apontamento apareceram no mercado, as interfaces começaram a sua evolução rumo a uma representação gráfica. Este passo foi uma revolução.

1.3.3 A GUI (*Graphic User Interface*) – A Interface Gráfica

GUI, abreviação para *Graphical User Interface*, é a denominação usualmente utilizada para definir as interfaces gráficas dos computadores. A evolução dos computadores pessoais rumo a uma interface graficamente representada foi tão natural, desejada e necessária quanto se poderia imaginar. Como afirma Dondis, “a compreensão visual é um meio natural que não precisa ser aprendido” [Dondis: 1991,186]. O computador só precisou de avanços tecnológicos suficientes (maior resolução de tela, maior capacidade de processamento e de memória) para passar a representar graficamente. Este fato é somente mais uma prova de que o homem se sente mais à vontade em se comunicar por meios visuais do que através de textos. Como diz Dondis, “para os analfabetos, a linguagem falada, a imagem e o símbolo continuam sendo os principais meios de comunicação e, dentre eles, só o visual pode ser mantido em qualquer circunstância prática. Isso é tão verdadeiro hoje quanto tem sido ao longo da história.” [idem,185].

No universo computacional, a grande massa de usuários é também analfabeta de algum modo: em relação ao funcionamento da máquina e de suas linguagens de mais baixo nível. A comunicação homem-máquina através do código binário ou, sem precisar ir muito longe, através das linguagens computacionais de alto nível (como Pascal, Basic ou C) é extremamente árdua para grande parcela dos usuários. Como afirma Julio Plaza:

“A imagem e a linguagem visual estão sendo atualmente privilegiadas pela informática, em parte devido ao caráter de condensação e síntese, que permite compreendê-la rapidamente fazendo jus à máxima: ‘Uma imagem vale por mil palavras’. Toda sorte de imagens, diagramas, metáforas e gráficos tratados pela informática, ajudam ao pensamento visual e verbal.” [Plaza, 1991:18]

Se tivéssemos que nos comunicar com a máquina por meio de sua linguagem natural, passaríamos por um processo de aprendizagem tão ou mais longo do que o que é necessário para aprender a ler e escrever. Não é à toa que hoje em dia há tantos cursos de nível superior dedicados ao ensino de programação e ao projeto de computadores.

A informação visual, ao contrário, é muito facilmente absorvida; é imediata. Pode ser percebida mesmo nas entrelinhas; pode ser sintetizada. Com diz Julio Plaza em resposta “à decomposição formal da imagem em linhas e pontos, o olho se encarregaria (pela lei da proximidade) de criar a continuidade das formas e objetos representados.” [Plaza, 1986:83]

Toda sorte de diagramas, esquemas, esboços podem ser utilizados para expressar idéias abstratas ou complexas. Tomando emprestado um termo usado por Umberto Eco, a codificação visual é fraca [Eco, 1971:122], o que significa que comparando com outros códigos, como as línguas, não há a necessidade um nível de aprendizagem tão demorado. A este respeito Dondis afirma:

“Ainda que uma descrição verbal possa ser uma explicação extremamente eficaz, o caráter dos meios visuais é muito diferente do da linguagem, sobretudo no que diz respeito a sua natureza direta. Não se faz necessário a intervenção de nenhum sistema de códigos para facilitar a compreensão, e de nenhuma decodificação que retarde o entendimento.” [Dondis, 1991:21]

“A linguagem separa, nacionaliza; o visual unifica. A linguagem é complexa e difícil; o visual tem a velocidade da luz, e pode expressar instantaneamente um grande número de idéias.” [idem, 82].

Pode-se argumentar que muitas expressões de caráter visual têm significações plurívocas, abertas. Poderíamos citar as obras abstratas ou surrealistas. Às vezes acontece ainda de uma imagem ter diferentes interpretações em função de diferentes culturas. Isto também é possível. Mas o que está se colocando em discussão é o caráter direto da comunicação visual. Bruno Munari afirma:

“Conhecer a comunicação visual é como aprender uma língua, uma língua composta só por imagens, mas imagens que têm significado para as pessoas de todas as nações e, por isso, de todas as línguas. A linguagem visual é uma linguagem talvez mais limitada do que a falada, porém, é mais direta.” [Munari, 1968:81]

Na computação, o visual também é o sentido que melhor tem obtido o efeito de transposição. A comunicação sonora também o é, mas é posterior. Quantos aos outros sentidos, somente mais recentemente pesquisas começaram a obter algum resultado, como no caso da simulação do olfato ou no mapeamento de movimentos corporais. A história tem nos mostrado que o sentido visual é o que melhor nos tem dado uma sensação do real. Basta fazermos uma comparação, imaginando-nos como uma pessoa sem olfato e depois como uma sem visão, para constataremos que sentimos mais a falta do segundo sentido. A interface gráfica, portanto, foi muito bem vinda.

Acredita-se que o pai das interfaces gráficas seja Ivan Sutherland do MIT (*Massachusetts Institute of Technology*) que em 1962 desenvolveu o SketchPad, um programa que permitia aos usuários criar linhas, círculos e pontos. No entanto, o primeiro sucesso comercial só veio muitos anos depois com o Apple Macintosh, em 1984, baseado no seu precursor de 1983, o Apple Lisa, que, por outro lado, foi criado a partir do sistema da Xerox, o Alto, do início dos anos 70 (para maiores detalhes, veja no capítulo seguinte).

A chave do sucesso do produto da Xerox está na metodologia de desenvolvimento do sistema como um todo. O Alto foi construído a partir de um modelo conceitual que leva em conta o relacionamento entre o usuário e o sistema. Este é representado por um ambiente de trabalho em um escritório, mais tarde chamada de “*desktop metaphor*” (metáfora do tampo de escrivaninha). Na tela são mostrados ícones de objetos de uso comum em uma empresa, como documentos, pastas, arquivos, lixeiras etc. No entanto, a natureza visual de uma interface gráfica não é o seu maior trunfo. Ela traz à mente do usuário todo um conjunto de relações que existe entre os objetos que compõem a interface, toda uma série de procedimentos e ações que fazem parte do modelo representado e que, portanto, são familiares.

A grande novidade das interfaces gráficas de sistemas computacionais é que ao contrário de outros espaços de representação, como a tela de um cinema ou o visor de uma televisão

analogica, elas podem ser manipuladas, ou como se diz no jargão corrente, interagidas diretamente pelo usuário.

Por serem interativas, as interfaces gráficas adquirem uma dimensão a mais. Ted Nelson, um dos mais atentos estudiosos de sistemas interativos, “percebeu a sensação de satisfação nos usuários, quando uma interface é construída pelo que ele chamou de princípio da virtualidade – uma representação da realidade que pode ser manipulada.” [Shneiderman: 1987,198] Chris Rutkowski apresentou um conceito similar através de seu princípio da transparência: “O usuário deve concentrar o seu esforço intelectual diretamente na tarefa; a ferramenta em si tem que desaparecer.” [idem, ibidem]. Este universo de conceitos e princípios, Ben Shneiderman chamou de “manipulação direta”. A interação do receptor com a interface provoca no usuário a sensação de que ele está em contato direto e real com o ambiente representado.

O grande sucesso dos videogames está em conseguir iludir o usuário, a ponto de ele perder a noção de estar, na verdade, agindo sobre uma representação. O mesmo pode-se dizer dos simuladores de voo. O usuário se sente imerso na interface. Alan Kay afirma que o interagente experimenta um estado de ilusão, de mágica (o autor estabeleceu o termo *user ilusion*). [Laurel, 1992:199] A interface é a parte integral da experiência do usuário.

[Interface de um jogo simulador. Usuário é um motorista]

Vulgarmente falando, vem daí a idéia de metáfora. É como se o receptor estivesse manipulando a ferramenta ou o objeto, mas, na verdade, ele está interagindo com a sua representação. A este respeito, Thomas D. Erickson, um importante pesquisador da Apple, afirma: “*metaphors serve as natural models; they allow us to take our knowledge of familiar objects and events and use it to give structure to abstract, less well understood*

concepts.”¹¹ [Laurel, 1992:73] Neste afirmação, o “abstrato”, que foi citado pelo autor, além de representar idéias e conceitos, pode ter ainda uma segunda interpretação: é também a linguagem computacional.

Estes modelos naturais, descritos acima, são considerados tão importantes que a IBM, em 1992, descreveu um guia (*Common User Access Guidelines*) sobre como se aproveitar deles para desenvolver o projeto da interface de um aplicativo. Os modelos mentais são definidos como “*a internal representation of how users understand and interact with a system*”¹² [Mandel, 1997:23]. Segundo Teo Mandel, um modelo mental ajuda o usuário a prever eventos futuros, a encontrar as causas de acontecimentos que são observados e a determinar as ações apropriadas que causariam mudanças desejadas. Serve também como um dispositivo mnemônico que lembre relações e eventos e ainda como um meio de entender dispositivos análogos. O autor completa:

*“Transferring knowledge of the world around them to the world of computers, users rely on models to guide their interactions with computers. This is where the concept of metaphors comes into play.”*¹³ [idem, 25]

A metáfora, se apoiada num modelo natural que conhecemos, faz o papel de tradutora, como ocorre normalmente na aplicação de metáforas na linguagem verbal. Para o meu estudo, no entanto, o desafio é saber como identificar sintaticamente este tropo em expressões de caráter gráfico-visual. A seguir, vamos descrever este processo com mais detalhes.

1.4 Metáforas em interfaces gráficas

O estudo da metáfora é principalmente pautado na linguagem verbal. São muitos os autores que dedicaram boa parte de seus esforços de pesquisa no estudo deste tropo, desde Aristóteles, passando por Paul Ricoeur, Max Black, Roman Jakobson, Jean Cohen, Hedwig

¹¹ T.A.: “Metáforas servem como modelos naturais; elas nos permitem transportar o conhecimento que temos de objetos e eventos familiares, estruturando o abstrato.”

¹² T.A.: “uma representação interna de como os usuários entendem e interagem com um sistema”

¹³ T.A.: “Transferindo o conhecimento que eles têm do mundo ao redor para o mundo do computador, os usuários contam com modelos para guiar suas interações com os computadores. É aqui que entra o conceito de metáfora.”

Konrad, Michel Le Guern, I. A. Richards, dentre outros. Um ponto em comum, que permeia o estudo destes autores, é que todos praticamente centraram sua investigação no signo verbal. Isto pode ser justificado pelo fato de a teoria tropológica fazer parte, historicamente, do estudo da Retórica, que, principalmente no passado, foi trabalhada dentro do contexto lingüístico.

Não podemos achar, no entanto, que as figuras de linguagem e, particularmente, as metáforas devam ser empregadas exclusivamente na linguagem verbal. Como afirmou Marvin Minsky, “a metáfora não é apenas um recurso especial usado na expressão literária e sim algo que permeia virtualmente todos os aspectos do pensamento humano” [Heckel, 1993:154].

Algumas pesquisas em campos teóricos paralelos podem, por outro lado, nos dar algumas pistas sobre o processo de aplicação de metáforas em interfaces gráficas. Primeiramente, a própria Teoria da Metáfora. Nela reside uma outra espécie de metáfora que prevê que o movimento de substituição que é efetuado no processo da construção metafórica pode ter o seu desvio de sentido realizado no eixo sintagmático, ao contrário do que ocorre no caso da metáfora-palavra, cujo desvio é feito no eixo paradigmático. Esta outra espécie de metáfora é chamada de metáfora-enunciado. Neste sentido mais do que a mera substituição de uma palavra por outra, a metáfora reivindica um olhar que abranja a estrutura da frase. Paralelamente, no campo da lógica de programação, temos uma nova concepção construtiva de interfaces que também leva em conta a estrutura e pode ajudar a nossa investigação: a “programação orientada a objetos”. A Semiótica tem também sua participação nos estudos de Julio Plaza relacionados à tradução intersemiótica e de outros autores que estudaram o processo construtivo da imagem, como Donis Dondis e Bruno Munari. Finalmente, não podemos deixar de lado a contribuição de Charles S. Peirce que colocou a metáfora como uma categoria de hipo-ícone.

1.4.1 Metáfora-enunciado

Em oposição à concepção clássica de metáfora, que afirma que o sentido metafórico é conduzido pela palavra, surge na década de 80 uma nova corrente formada por Ivor A.

Richards, Max Black e Monroe Beardsley, conhecida como grupo anglo-saxão. Para eles, o sentido metafórico é conduzido não pela palavra, mas pela sentença como um todo. Ou seja, o fio condutor do sentido metafórico é fornecido pelo enunciado. Por esta razão, esta metáfora é denominada metáfora-enunciado.

Max Black formula uma nova teoria semântica para a metáfora-enunciado, chamada teoria da interação. Para o teórico, na formação do sentido metafórico, há uma interação entre um sujeito e um predicado lógicos, que vai além da simples substituição de uma palavra por outra ou de um nome por outro. Como afirma Paul Ricoeur, “o problema da semelhança recebe uma nova formulação na teoria semântica caracterizada por Max Black como uma teoria da interação (oposta a uma teoria substitutiva). O condutor do sentido metafórico não é mais a palavra mas a sentença como um todo.” [Sacks, 1992:147] O que constitui a metáfora é o enunciado por inteiro, já que nem todas as palavras de um enunciado são metafóricas. Mas Black considera que a atenção se concentra sobre uma das palavras e no jogo desta com o enunciado é que se forma o sentido figurado. O autor designa essa palavra isolada como *focus* (foco) e o resto da frase, por *frame* (moldura).

A abordagem de Richards é colocada sob outras bases. Para ele, somente o discurso, ou seja o enunciado, tem condições de fornecer à palavra um significado. Ela, sozinha, não fornece sentido. As palavras só têm significação dentro de um contexto. Por isto, Richards instaura a teoria da interpenetração “das partes do discurso sobre as quais se edifica a teoria da interação própria da metáfora-enunciado.” [Filipak, 1983:117] Beardsley apresenta um ponto de vista bastante parecido. Para ele, a palavra apresenta sozinha uma significação isolada. Mas é a frase que dá um sentido completo. Um exemplo de metáfora-enunciado é a expressão “tirar o pé da lama” (Em, por exemplo: “estou cheio de dívidas, mas vou *tirar o pé da lama*”), pois, neste caso, a metáfora é definida pela sentença como um todo. Há um sentido indiviso, fornecido pelo contexto, na interação das palavras. Ou seja, a teoria metafórica não é fundamentada na lingüística, “mas na lógica e mais exatamente na lógica proposicional que leva em consideração a frase (não a palavra) e desenvolve a metáfora com as características de um enunciado, dentro do esquema da predicação.” [idem, 116]

No entanto, seria um equívoco achar que somente a metáfora-enunciado fundamenta o seu sentido na lógica. A metáfora-palavra também o faz. Paul Ricoeur afirma que a diferença é que o aspecto lógico da reestruturação desta é mais conspícuo. É a metáfora por analogia ou por proporção: ‘Fulana está para uma rosa assim como Sicrano está para um cachorro.’ Se a metáfora-enunciado tem o sentido lógico definido pela proposição com um todo, num sentido horizontal, vamos dizer; a metáfora-palavra o tem por comparação, um sentido verticalizado. É o que Saussure define como ‘os dois eixos da linguagem’: o sintagmático e o paradigmático. [idem, 63]

Numa construção lingüística, a combinação dos elementos significantes é o que Saussure chama de sintagma. Para ele, ‘o sintagma se compõe sempre de duas ou mais unidades consecutivas (por exemplo: re-ler; a vida humana; Deus é bom; se fizer bom tempo, sairemos; etc). Colocado num sintagma, um termo só adquire seu valor porque se opõe ao que precede ou ao que o segue, ou a ambos.’ [idem, 64].

O sentido da metáfora-enunciado é formado, portanto, no eixo sintagmático. A este respeito Ricoeur afirma: ‘é necessário buscar o segredo da metáfora pelo lado das uniões sintagmáticas’ [Sacks, 1992:173], ao contrário da metáfora-palavra, cujo desvio repousa no eixo paradigmático. Segundo o mesmo autor a metáfora ‘precisa ser descrita antes como uma predicação alterada do que como uma denominação alterada.’ [idem, 147] Ricoeur explica assim o significado de predicação alterada:

‘Um teórico francês no campo da poética, Jean Cohen, e m *Structure du Langage Poétique*, fala desse desvio em termos de não-pertinência semântica, querendo dizer com isso a violação do código de pertinência ou relevância que governa a atribuição dos predicados em uso normal. A expressão metafórica funciona como a redução desse desvio sintagmático através da fixação de uma nova pertinência semântica. Essa nova pertinência, por sua vez, assegurada pela produção de um desvio lexical, que é então um desvio paradigmático, ou seja, exatamente o tipo de desvio descrito pelos retóricos clássicos. A retórica clássica, nesse sentido, não estava errada, mas apenas descrevia o ‘efeito do sentido’ no nível da palavra enquanto omitia a produção dessa guinada semântica no nível do sentido. Enquanto for verdadeiro que o efeito do sentido é focalizado na palavra, a produção do sentido é transmitida pelo enunciado como um todo. É dessa maneira que a teoria da metáfora depende de uma semântica da sentença’ [idem, 147]

Podemos dizer que as relações sintagmáticas são construídas a partir de uma base una, indivisível, neste caso a sentença. Já nas relações paradigmáticas, a formação do sentido é criada em função de uma seleção. Na sentença, há um conjunto de elementos fixos, mas

uma das palavras é formadora do sentido metafórico. Esta palavra pode ser substituída para formar outro significado. E, portanto, a base como um todo adquire diversos sentidos. São sentidos que estão lá, virtualmente formados, basta escolhê-los. Jean Dubois diz que “as relações paradigmáticas são as relações virtuais existentes entre as diversas unidades da língua que pertencem a uma mesma classe morfossintática e/ou semântica.” [Dubois, 1991:453]. Por outro lado, a formação de sentido metafórico por meio de metáforas-enunciado não tem este caráter, porque os signos verbais formam uma unidade. É a visão da lingüística estrutural: “síntagma é um grupo de elementos lingüísticos que forma uma unidade numa organização hierarquizada.” [Filipak, 1982:65] Um termo do sintagma só adquire valor porque se opõe aos outros termos da sentença. Quando o valor resultante da relação entre os termos que formam a unidade semântica é inusitado, estranho como diz Aristóteles, há a possibilidades de estar se empregando uma metáfora. Há uma espécie de contraste ou de tensão entre os seus elementos.

Portanto, a construção de uma metáfora depende do modo como os elementos que estão estruturados em um enunciado são combinados. Fica claro que a estrutura da frase tem papel preponderante no caso da construção da metáfora-enunciado. Como afirma Ricoeur: “no enunciado metafórico, a ação contextual cria uma nova significação que tem o estatuto de acontecimento, posto que existe só neste contexto” [Ricoeur, 1975:15]. Cabe averiguar como este mesmo conceito se aplica numa estrutura visual.

1.4.2 A estrutura da imagem digital

Segundo Pouillon, “a estrutura é aquilo que a análise interna de uma totalidade revela: elementos, relações entre tais elementos, e a disposição, o sistema dessas mesmas relações” [Pouillon, 1966:8]. Na linguagem verbal, os elementos constitutivos de um enunciado são os caracteres, os fonemas, as palavras e as frases. Segundo Saussure, a característica da fala, e eu incluiria, num nível menor, da escrita, é a liberdade de combinações. No entanto, dependendo do elemento formador na estrutura do enunciado, há uma liberdade maior ou menor. Jakobson diz que a liberdade de combinar palavras a partir de fonemas é menor do que a de se criar frases a partir de palavras. E a liberdade de criar frases em enunciados

aumenta substancialmente, diante da ausência de regras estipuladas neste caso. A formação de palavras a partir de caracteres e fonemas exige o cumprimento de uma série de regras. O mesmo ocorre na formação de frases a partir de palavras: as combinações são determinadas pela sintaxe da linguagem.

Na expressão visual, também podemos fazer uma análise a partir da combinação de seus elementos. Segundo Dondis, os elementos da imagem são o ponto, a linha, a forma, a direção, o tom, a cor, a textura, a dimensão, a escala e o movimento. O processo formador da imagem é feito pela interação destes elementos: “...criamos um *design* a partir de inúmeras cores e formas, texturas, tons e proporções relativas; relacionamos interativamente esses elementos; temos em vista um significado.” [Dondis, 1991:30].

Mas há uma diferença fundamental entre um enunciado verbal e visual. Na estrutura visual, a liberdade de combinações dos elementos que formam uma imagem é maior. Mesmo que estes elementos sejam combinados aleatoriamente, ainda assim a imagem terá uma certa legitimidade. Uma imagem abstrata possui uma aceitação que um enunciado formado por palavras aleatórias ainda não têm. Mas engana-se quem acredita que a linguagem visual não possui sua sintaxe própria. Há estudos neste sentido. Dondis afirma que “em termos lingüísticos, sintaxe significa disposição ordenada das palavras segundo uma forma e uma ordenação adequadas. As regras são definidas: tudo o que se tem de fazer é aprendê-las e usá-las inteligentemente. Mas no contexto do alfabeto visual; a sintaxe só pode significar a disposição ordenada de partes.” [idem, 29] No entanto, “não há regras absolutas.” [idem, ibidem]. A disposição das partes ou, em outras palavras, a composição da imagem é o passo fundamental na solução de problemas visuais.

Numa tela, numa pintura, os elementos formadores (o ponto, a linha, a forma, a direção, o tom, a cor, a textura, a dimensão, a escala e o movimento) se combinam e qualquer resultado é possível. Estes elementos são fundamentais para a formação do significado. A alteração de um ponto ou de uma cor altera o significado do todo. Segundo Dondis, a base teórica é a Gestalt, cuja crença é “que uma abordagem da compreensão e da análise de todos os sistemas exige que se reconheça que o sistema (ou objeto, acontecimento etc) como um todo é formado por partes interatuantes, que podem ser isoladas e vistas como

inteiramente independentes, e depois reunidas no todo. É impossível modificar qualquer unidade do sistema sem que, com isso, se modifique também o todo.” [idem, 51]. Nas interfaces gráficas computacionais este pressuposto também é válido. A diferença é que todos os elementos formadores são sintetizados e transformados através de algoritmos.

“Essas imagens incorporam os sistemas tradicionais de representação iconográficos e criam outros decorrentes de suas estruturas. Surgem aqui, outras modalidades de criação de imagens pela codificação numérica das estruturas dos objetos e fenômenos que representam. Também pela aquisição e posterior digitalização de imagens que remetem a um novo pictorialismo e ideografia.” [Plaza, 1991:18]

A estrutura da imagem digital é hoje bem conhecida. A tela do computador é composta por diversos pontos luminosos enfileirados na horizontal e na vertical formando o espaço retangular delimitador da interface. Quanto maior o número de

maior a resolução da imagem. Num processo chamado de varredura, cada ponto da tela recebe três feixes de elétrons de diferentes intensidades; cada feixe representando um das três cores primárias. O resultado é que cada ponto será mostrado com uma nova cor, em função da fusão perceptiva das três cores primárias, ou seja, a produção da cor é óptica. Finalmente, a soma de todos pontos da tela formam a imagem final. Numa velocidade extremamente rápida para a percepção humana, uma nova varredura é executada na tela de tempos em tempos, atualizando as cores apresentadas em cada ponto da imagem de acordo com o algoritmo corrente. Como diz Arlindo Machado, “a imagem eletrônica não é mais, como eram todas as imagens anteriores, inscrição no espaço, ocupação da topografia de um quadro, mas síntese temporal de um conjunto de formas em mutação.” [Machado, 1993:52]. O aspecto fundamental é o caráter dinâmico da imagens digitais. Elas se transformam, alteram seus estados de cor, luminosidade, textura e todos os aspectos citados como elementares por Dondis. As imagens sofrem processos de mutações constantes, ao gosto de algoritmos e fórmulas matemáticas. A interface gráfica computacional está inserida numa estrutura espaço temporal. Portanto, os objetos representados ganham “vida”; deixam de ser “natureza morta”.

Neste estudo, que pretende identificar metáforas nas interfaces computacionais, não são os elementos apresentados por Dondis, tal qual cor, forma, linha, textura etc, que nos interessam na análise estrutural da imagem. O conceito fundamental, o significado almejado, é o de objeto. Na formação da metáfora, os objetos estão para as interfaces gráficas, assim como as palavras estão para o enunciado verbal. É na interação dos objetos com o usuário e na inter-relação de um objeto com os outros objetos da interface que se estabelecem as metáforas. A fusão destes objetos resulta num significado maior, que é apresentado no sintagma visual, no espaço de representação, que, no caso, é a interface.

1.4.3 Orientação a objetos

Colocamos o objeto como a entidade fundamental dentro de uma possível análise sintática da estrutura visual representada pela interface computacional. No entanto, este mesmo termo, o de “objeto”, é também empregado em outro modelo que nos interessa, só que relacionado à programação de computadores, denominado “programação orientada (ou estruturada) a objetos”. Longe de concorrer com o conceito anterior, este conceito de objeto relacionado agora à programação só vem confirmar o seu papel neste estudo. Mas o quem a ser programação orientada a objetos?

Muito além de um modismo, por sinal bastante difundido atualmente, como uma forma mais fácil de programar computadores do que os antigos modelos de codificação, a Programação Orientada a Objetos (POO) reflete os novos paradigmas computacionais que foram impulsionados pelos diversos avanços tecnológicos recentes do computador: velocidade de processamento, capacidade de memória e, principalmente, a adoção de interfaces gráficas.

As linguagens computacionais mais antigas forneciam um bom suporte para construção de programas que privilegiavam a representação de procedimentos. Em interfaces orientadas a linhas de comando isto era bastante conveniente, porque os programas eram destinados a resolver problemas que tinham a linha de *prompt* (ou de comando) como principal suporte comunicativo. O usuário digitava um comando ou respondia a uma solicitação do programa e uma ação era executada. Na verdade, este modelo lógico não foi abandonado. Os

programas continuam procedimentais. No entanto, houve uma pequena distribuição destas ações em diversas entidades, cada uma responsável por executar determinados comportamentos individualizados, mas que podem interferir no programa como um todo. Estas entidades são chamadas de “objeto”.

Este termo, “objeto”, não foi escolhido à toa. Por traz desta definição, há a idéia de que o nosso mundo real é composto por objetos. Aproveitando o fato de que as interfaces atuais são gráfico-visuais, os programas podem reproduzi-lo. Mas como isto funciona?

Um programa ou uma aplicação de computador representa um modelo ou uma solução para um problema do mundo real. Os projetistas de computador identificam dois espaços de representação: o Espaço de Soluções e o Espaço de Problemas. O desenvolvimento do programa envolve “um processo de mapeamento de aspectos de objetos pertencentes ao Espaço de Problemas para representações abstratas no Espaço de Soluções, de tal maneira que operações sobre essas representações abstratas correspondam a operações no mundo real.” [Takahachi, 1988:2] A partir daí, o projetista cria o algoritmo. Quando executado no computador, o programa deve produzir no Espaço de Soluções as ações que foram mapeadas no Espaço de Problemas.

O Espaço de Soluções pode ser representado livremente. Em geral é determinado um modelo de representação. É evidente, no entanto, que quanto mais próximo o Espaço de Soluções estiver do Espaço de Problemas, mais fidedigna será a representação. Mais segura também será a possibilidade de compreensão por parte do usuário. Um objetivo claro por parte do projetista é diminuir a distância semântica entre o modelo representado no Espaço de Soluções com as abstrações percebidas a partir do Espaço de Problemas.

Com a possibilidade que hoje o computador possui de representação visual das suas interfaces ficou mais fácil a modelização do mundo real. E como este é composto de objetos, nada mais natural do que tomá-los como ponto de partida semântico e torná-los âncoras no processo estrutural de programação. Esta metáfora de objetos tem sido aplicada no projeto de sistemas e se consolidou como um paradigma importante na computação.

Dentro deste paradigma, objetos são entidades autônomas em relação ao resto do sistema, possuidores de uma memória local independente e inacessível a outros objetos. Funcionam como uma variável de memória¹⁴, pois recebem dados que podem ser alterados. Os dados armazenados de um objeto descrevem o seu estado interno. Correspondem portanto às suas características ‘pessoais’, como cor, posição na tela, tamanho etc. Tecnicamente, estas características são denominadas propriedades.

Além das propriedades, os objetos possuem comportamentos específicos. São comumente chamados de métodos e correspondem a um repertório de ações que cada objeto possui para responder a determinados eventos do sistema. Assim como mundo real, os objetos podem e devem interferir no comportamento e nas propriedades de outros objetos. Não confundir isto com a memória interna do objeto que continua a ser indevassável, pois corresponde à sua autonomia ontológica. Embora tudo isto possa parecer demasiadamente técnico, este modelo reflete o funcionamento sistemático do mundo real. Objetos são como pessoas que possuem características (as propriedades), como peso, altura, nome, modo de vestir etc e comportamentos pessoais, como a capacidade de falar, pensar etc.

Finalmente, há os eventos. Um evento é o modo como um objeto ou o sistema interfere em outros objetos ou no sistema como um todo. É todo tipo de processamento ativado por mensagens. Quando um objeto recebe uma mensagem, ele executará uma ação específica, como alterar o seu estado interno. Assim, no exemplo ‘quando clicar em um botão, pinte a cor da imagem de branco’, ‘clicar’ é o evento. A partir deste evento, uma mensagem é enviada do botão à imagem. Ambos são objetos. Cor é a propriedade do objeto imagem. E pintar é o método. No mundo real fazemos o mesmo: ‘quando abrir o farol, vou passar com o meu carro’. ‘Abrir o farol’ é o evento. O motorista e o carro são objetos. Passar é o método. Há ainda a propriedade, não mencionada explicitamente, mas existente, como o fato de o carro estar parado ou não.

¹⁴ Variável de memória é um conceito empregado em programação de computadores para representar um espaço significativo de memória. Quando o programador percebe a necessidade de guardar um dado que está sendo utilizado no seu algoritmo, ele declara ao sistema a que precisa ocupar um determinado espaço da memória para este fim. Simbolicamente este espaço na memória é representado pela entidade chamada variável. O nome variável advém do fato de que o dado que foi guardado lá poder ter o seu valor alterado.

Vejamos um exemplo mais real de aplicação na computação. Poderíamos, por exemplo, criar um sistema simulador de vôo. A interface poderia ser a representação do painel da aeronave. Seria o Espaço de Soluções. Como o objetivo é treinar futuros pilotos de vôo, nada mais natural que definir como objetos os botões, as alavancas, a direção, o radar, a bússola, os acessórios do próprio painel e também a representação de um piloto, que funcionaria como o alter ego do próprio usuário. Quando o piloto virtual usa a direção da aeronave e muda a rota do avião, obviamente, a bússola deve refletir este novo posicionamento. Esta é um exemplo bastante simples de como um objeto (o piloto) afeta a propriedade (posição do avião, orientação da agulha magnética da bússola) de outros objetos (avião, bússola) do sistema. É um exemplo de como o Espaço de Soluções pode ser representado de modo bastante similar ao Espaço de Problemas. Nem sempre é assim, como veremos mais adiante.

Um fato inquestionável, no entanto, é que todos os objetos podem receber mensagens. Esta é a forma de interação. Sem esta possibilidade, a representação espacial seria estática, pois não poderíamos afetar os objetos diretamente. A maneira de estabelecer as conexões é feita por codificação. Cada objeto possui no seu estado interno o *script* (programa) pessoal que permite afetar o sistema e os objetos do sistema. A maioria das linguagens de programação possuem um ambiente de codificação que permite criar o programa interno de cada objeto. Este programa é como o código genético do objeto. Nele estão estabelecidas internamente todas as possibilidades do objeto, que explicitam como, quando e quais objetos serão afetados por ele. É exatamente neste ponto que os técnicos de computação descrevem a programação como “orientada a objetos”. Todo o processamento ocorre internamente, dentro do objeto, e é ativado por eventos direcionados a estes objetos. Um objeto recebe uma mensagem e reage a ela através de um processamento. O processamento é solicitado através de outra mensagem que solicita processamento a outro objeto e assim por diante. Isto tem que ter um fim. Existem, portanto, objetos chamados primitivos que não enviam mais mensagens e somente processam os dados. Estes objetos têm um caráter bastante abstrato, como, por exemplo, os valores numéricos, e às vezes nem mesmo podem ser vistos diretamente no Espaço de Soluções. O que demonstra claramente o caráter metafórico dos objetos mais complexos abstraídos do mundo real. Para o sistema mais

primitivo do computador, que só entende a linguagem binária, os objetos representados no Espaço de Soluções não passam de números conjugados. Como estes objetos primitivos podem ser combinados de forma a representar outros objetos? Isto é possível devido ao caráter atomístico do sistema. Tal como um jogo de lego, um objeto combinado com outro pode levar ao desenvolvimento de um terceiro. Cada peça do lego é um objeto. Tem tamanhos, cores e formas diferentes. Mas são objetos individuais. É assim também no mundo real. A programação orientada a objetos trabalha da mesma forma. Um programa montado no Espaço de Soluções é composto por um conjunto de objetos auto-suficientes, cada um com uma função específica dentro do sistema, mas podendo se comunicar através de comportamentos previamente definidos.

Por isso mesmo há uma hierarquia nestes objetos. Há objetos mais gerais ou mais específicos. Objetos menores e maiores. Concretos e abstratos. Mas no geral há objetos que possuem características bastante similares. Neste caso, são divididos em classes. Podemos fazer novamente uma analogia com o mundo real. Existe a classe veículo automotivo. Dentro dela há as subclasses carro, caminhão, motocicleta ou trator. Na classe carro temos as peruas, os esportivos, os de luxo etc. Num sistema computacional isto é também adotado.

Por exemplo existe a classe botão. Dentro desta classe temos algumas instâncias, como o botão de seleção (*radio button*), o botão de escolha múltipla (*check box*), o tradicional botão de apertar (*push button*) etc. Podemos constatar, a

partir disto, que as classes possuem características comuns. “O desenvolvimento orientado a objetos é centrado na identificação de objetos e na construção de uma hierarquia de classes que fatora as propriedades comuns de subclasses e de objetos.” [idem, 7]

A importância da classificação dos objetos está no momento de projetar ou programar a aplicação. Na criação de um objeto, podemos identificá-lo como parte de uma classe e construí-lo a partir dela. É como se quiséssemos criar uma vestimenta diferenciada, mas

que se assemelhasse a uma roupa existente no mercado. Em vez de desenvolvê-la a partir do zero, podemos comprar esta roupa e fazer as alterações desejadas. Na orientação a objetos isto também é possível. A partir de uma classe, pode-se construir um novo objeto, uma instância desta classe, que ignore algumas propriedades originais da classe em proveito de novas características. A classificação de objeto, portanto, permite “identificar partições naturais de sistemas e programas em objetos, e associar propriedades e comportamentos no nível adequado de abstração, evitando duplicações, unindo o que é comum e complementando os aspectos comuns através de especialização.” [idem, 8]

Alguns autores, como Goldberg, sugerem que se enxergue cada objeto como um mini computador. Possui memória, é programado para executar determinadas operações, receber e enviar mensagens etc. “O universo de objetos seria constituído por um número indefinido (posto que variável ao longo do tempo) desses computadores ligados em rede.” [idem, 6] Uma segunda maneira de ver isto, mais conservadora, mas não menos válida, é enxergar cada objeto como um miniprograma, com uma função específica dentro do Espaço de Soluções. Esta visão enfatiza a modularidade do sistema, permitindo a qualquer momento incluir novos objetos.

Existem outras características presentes na programação orientada a objetos que poderíamos ainda descrever, como, por exemplo, o paralelismo, que como o nome indica é a capacidade do sistema de poder processar ações de objetos ao mesmo tempo, ou o “ciclo de vida do objeto”, posto que os objetos podem nascer e morrer, assim como no mundo real. Não há a necessidade de entrar em pormenores destes aspectos. O mais importante é perceber que o modelo de objetos “é bastante adequado a diversas aplicações práticas que são intrinsicamente distribuídas e em que partes se comunicam através de mensagens reais.” [idem, 8].

1.4.4 A escolha de uma metáfora no Espaço de Soluções: a interface e seus objetos

Para Dondis, a fusão dos elementos pertencentes à sintaxe da expressão visual (as cores, as formas, as texturas etc) são relacionados tendo em vista um significado. Neste estudo, por

uma questão estratégica, definimos os “objetos” como elementos centrais da sintaxe da interface. Combinados eles também têm como objetivo a formação de um significado. No entanto, não podemos esquecer, que, antes de mais nada, os objetos representados na interface são signos. Signos visuais para ser mais exato.

Um dos aportes negativos dos signos visuais é que eles podem ser interpretados de forma muito mais livre do que outras formas de expressão. Os signos visuais seriam formadores de *códigos fracos*, tomando emprestado mais uma vez a expressão de Eco, já que são “muito pouco definidos, continuamente mutáveis, em que as variantes facultativas prevalecem sobre os traços pertinentes.” [Eco, 1971:122] É verdade que alguns signos visuais são expressados segundo alguma forma de codificação. É o caso de algumas sinalizações, em que mais do que uma interpretação livre, o que vale é uma convenção. São símbolos, que muitas vezes são construídos justamente por metáforas ou metonímias. Quando vemos na porta de banheiros públicos uma placa com a figura de uma cartola, sabemos de imediato que se trata do sanitário masculino, não de uma sala que vende este tipo de chapéu. Embora dificilmente encontremos algum homem de cartola no Brasil, ninguém mais pára para pensar se aquele ícone corresponde à porta masculina. A metáfora morreu, mas tornou o ícone da cartola praticamente em um símbolo.

No entanto, as interfaces gráficas não são formadas somente por símbolos. Há toda uma série de formas significantes cuja interpretação não é convencionada. São esquemas, diagramas, imagens capturadas e diversas formas de expressões iconográficas.

Qual é a importância desta discussão? Vejamos. Se a metáfora é definida por Dubois como “o emprego de todo termo substituído por um outro que lhe é assimilado”, fica estabelecida a necessidade de o signo (que neste caso é verbal, o termo) ter um significado mais ou menos convencionado. A palavra “cadeira” é cadeira e ponto; ou seja, é um objeto de material mais ou menos resistente, cuja função é assentar pessoas. Se alguém diz que “Fulano de Tal é uma cadeira” há um estranhamento inicial, até que o interlocutor interprete o enunciado dentro de um contexto diferenciado e forneça ao signo “cadeira” um novo significado. Do mesmo modo, a imagem capturada de uma cadeira, numa foto, por exemplo, também pode ser interpretada somente como o objeto cadeira. E tal qual um

enunciado, se a imagem da cadeira estiver vazia num quarto, ela pode ser interpretada como o sentimento de ausência ou de solidão.

Portanto, o processo de resignificação da metáfora não necessitaria, em tese, de códigos fortes. A expressão visual não apresenta signos que contenham esta vocação, mas isto não impede que a metáfora também seja aplicada como recurso fomentador das artes visuais. O surrealismo é prova disto. A metáfora da expressão visual se processa no eixo sintagmático, da mesma forma como foi interpretada pelo grupo anglo-saxão ao definir a metáfora enunciado. A diferença é que em vez de signos verbais temos signos visuais. O sintagma visual corresponderia à interface.

A aplicação de metáforas em interfaces computacionais opera, no entanto, de modo diferente. O significado resultante de uma tela surrealista é livre, onírico, um tanto quanto vago. O significado que a interface computacional tem em vista é preciso. Esta dicotomia significado aberto/significado fechado que a metáfora sustenta é perfeitamente possível, com atestam exemplos de enunciados verbais. Existem metáforas, geralmente de caráter poético, cujos significados devem ser alcançados livremente pelo leitor e existem metáforas criadas para definir um idéia, cuja intenção é argumentativa. É o que definimos como as duas funções da metáfora: a poética e a retórica. A primeira é conotativa, emotiva, catártica (segundo Filipak), de significação particular (Dubois), apelativa (Jakobson), emotiva (Charles Bally), empregadas conforme o que Le Guern chama de linguagem plurívoca, aberta; a segunda é denotativa, lógica, intelectual (Filipak), referencial, cognitiva {Jakobson}, empregada com o que André Martinet chama de função comunicativa.¹⁵

Nas interfaces gráficas computacionais, as metáforas não aparecem como um ruído. Elas não são notadas. Não pretendem provocar o citado estranhamento que falamos acima quando empregamos as metáforas em certas sentenças verbais. A intenção é a de facilitar a compreensão de um sistema. A aplicação de metáforas em interfaces gráficas de sistemas interativos tem a intenção declarada de facilitar a comunicação. Neste sentido são denotativas, sua função é principalmente retórica.

¹⁵ Todas estas referências a autores foram retiradas de FILIPAK, Franciso. *Teoria da Metáfora*. Curitiba, HDV, 1983, p.9.

Evidentemente, o caráter poético das metáforas não é completamente abandonado. Em certos casos, a função catártica se impõe, praticamente se torna única. É o caso de certas manifestações da Web arte, que procuram explorar ao máximo o potencial estético da hipermídia. Não se trata de catálogos de artistas presentes em páginas Web, ou *sites* de museus virtuais. Estes são apenas suportes para a apresentação de outras expressões artísticas. No caso da Web arte, o suporte é parte atuante do processo. São trabalhos que buscam o rompimento, que procuram o experimento, que anseiam pela dissolução dos elementos componentes da estrutura hipermidiática.

[Algumas manifestações de Web arte]

De uma forma ou outra, no entanto, o que está se levando em conta neste estudo é mais uma vez o papel da metáfora como agente de substituição. Segundo a definição mais usual, a metáfora se processa quando uma palavra é substituída por outra de caráter similar. Mas na interface, a expressão é visual. Se transportarmos esta definição para o sintagma da imagem, quem desempenha o papel das palavras são os diversos signos visuais representados; os objetos. Pela definição, diríamos também que os signos visuais deveriam ser substituídos por outro de caráter similar, operação esta que provoca o processo de ressignificação. Mas é neste ponto que começam a aparecer as dificuldades conceituais.

Pelo fato de operar com códigos fracos, a expressão visual torna mais diluída a percepção de substituição. Tanto a sintaxe quanto o processo de significação de palavras no enunciado verbal são muito mais fortes e definidos do que ocorre com os signos de um sintagma visual. Voltando ao exemplo da cadeira: “o homem é uma cadeira”. A primeira impressão é que alguma palavra foi colocada de maneira equivocada, ou “cadeira”, ou “homem”, ou mesmo o verbo ser. Esta percepção é muito bem definida porque as palavras possuem significados mais ou menos precisos. O mesmo se opera no caso da sintaxe. No enunciado verbal, a sintaxe é muito rígida. Frases do tipo “cadeira homem o uma é” não faz

absolutamente nenhum sentido, pois as palavras foram colocadas sem respeito às normas gramaticais.

Analisando a linguagem visual, embora Dondis insista na existência de uma sintaxe, não há dúvidas que neste caso ela é muito mais livre. Uma composição visual pode formar desde um representação completamente realista até uma tela abstrata. Isto atesta a grande dificuldade em transportar a teoria tropológica da metáfora para expressões visuais. Mas há esta possibilidade.

Voltemos à tela surrealista. É bastante conhecido o fato de que alguns destes artistas pintavam imagens de caráter onírico. Como diz Gombrich, a respeito dos sonhos e dos surrealistas: ‘hos sonhos, com freqüência experimentamos a estranha sensação de que pessoas e objetos se fundem e trocam de lugar. Nosso gato pode ser ao mesmo tempo nossa tia, e o nosso jardim ser a África. Um dos principais pintores surrealistas, o espanhol Salvador Dali (...), tentou imitar essa confusão fantástica da nossa vida onírica’ [Gombrich, 1999:592]. Somente nesta afirmação, podemos perceber a presença de três palavras-chaves da Teoria da Metáfora: estranha, objetos e troca. Ou seja, numa tela surrealista, objetos (signos) são colocados de maneira estranha; são trocados (substituídos) por outros que deveriam estar lá. Se repararmos com atenção, esta é a definição mais usual de metáfora.

Talvez por se colocarem através de uma estrutura quase realista, as imagens surrealistas permitem ao receptor perceber o enigmático, o inesperado; ele nota que há algo estranho na representação. Na obra “A Canção do amor” de Giorgio de Chirico, há, na mesma tela, uma cabeça clássica esculpida na parede de uma espécie de galpão, ao lado de uma enorme luva pendurada. Na frente de ambas, uma esfera verde colocada numa superfície plana. No fundo, há a silhueta de uma cidade. Os

elementos parecem não combinar: a luva deveria ser menor, ou nem estar lá. E a esfera verde talvez devesse ser trocada por algo relacionado ao ambiente, como uma árvore ou um banco de praça. Por outro lado, numa segunda leitura possível, poderíamos buscar o significado da composição em função das relações expostas. A racionalidade deveria então ser deixada de lado, favorecendo o que chamamos de função poética da metáfora. Como diz Gombrich a respeito da obra de Dali (mas que poderia valer para de Chirico ou outro surrealista):

“O modo de Dali fazer cada forma representar muitas coisas ao mesmo tempo pode concentrar a nossa atenção nos muitos significados possíveis de cada cor e de cada forma – de maneira semelhante àquela em que um trocadilho bem-feito nos faz compreender a função da palavra e seu significado.” [idem, 594]

Uma possibilidade é que as metáforas de signos visuais poderiam se situar, então, na terceira tricotomia de signos da teoria peirciana, que leva em conta a relação do signo com seu interpretante. Ou seja, a metáfora poderia se situar no rema, no dicente ou no argumento. Só para ficar num exemplo, o rema, para Peirce, é: “um Signo que, para seu Interpretante, é um Signo de possibilidade qualitativa., ou seja, entendido como representando tal e tal espécie de Objeto possível. Todo Rema fornecerá, talvez, alguma informação; mas não é interpretado como destinado a fazê-lo” [Peirce, 1975:102] Ou seja, a partir de certos ícones, às vezes meras formas, podemos interpretá-los com a representação de um objeto, embora não fosse esta a intenção inicial. Diante de nuvens, costumamos dizer: “Veja, é a cara de um cachorro”. O mesmo fazia m os antigos quando denominavam as constelações a partir de objetos que se assemelhavam ao conjunto de certas estrelas. Como afirma Santaella:

“Ora, aquelas formas de fato não representam essas imagens. Podem, quando muito, sugerí-las. É por isso que o interpretante que o ícone está apto a produzir é, também ele, uma mera possibilidade (qualidade de impressão) ou, no máximo, no nível do raciocínio, uma rema, isto é, uma conjectura ou hipótese. Daí, que diante de ícones, costumamos dizer: ‘Parece uma escada, uma montanha...’ ” [Santaella, 1983:65]

Podemos interpretar estes signos quase como um tipo de metáfora da representação visual. Eles cumprem uma função poética. Podem muito bem ser encontrados em pinturas abstratas ou surrealistas, mas isto não é uma característica das interfaces gráficas de

computadores. De caráter mais denotativo, estas parecem, em princípio, não aceitar o pressuposto metafórico. Mas a complexidade é maior.

Os signos visuais interfaciais têm uma intenção: eles representam um objeto. Não há outra interpretação possível. O simples fato de representarmos uma lixeira na interface não significa nada mais do que a representação icônica da lixeira. Afirmamos acima que os elementos componentes do sintagma visual da tela do computador (representado pela interface) são signos visuais que, na maioria das vezes, desempenham o papel de objetos. “As imagens são substitutos (Gibson, 1965) de objetos da realidade. Representando estes objetos, elas servem à comunicação.” [Plaza, 1986:107] Ou seja, nas interfaces, estes signos representam os objetos em si. Eles não têm a pretensão poética de substituir um signo que represente outro objeto. Com afirma Julio Plaza:

“As figuras ilustradoras tratam de captar a imagem visual que esses mesmos objetos oferecem ao olho, a partir de um ponto de vista típico e universal (válido para muitos receptores). Incluem, assim, taxas diferenciadas de normalização. Pelo seu princípio formador esquemático, elas traçam os aspectos característicos dos objetos ‘caricaturizando-os’. São signos de experiência direta na medida em que atraem a atenção para seus objetos referenciados, dos quais, aliás, decorrem na sua apresentação, levando o intérprete a se dirigir mentalmente ao próprio objeto.” [idem, 118]

Neste sentido, os signos visuais das interfaces gráficas se situam na chamada segunda tricotomia de signos de Peirce, aquela que estabelece a relação do signo com o objeto que ele representa. “Para que algo seja um Signo deve ‘representar’, como dissemos, algo diverso que é chamado seu Objeto”. [Peirce, 1975:95] De acordo com esta classificação, um signo pode ser um ícone, um índice ou um símbolo.

Na definição de Peirce, ícones são signos cuja qualidade que possui com a coisa (o objeto representado) torna-o apto a ser um Representamen¹⁶, são “aqueles signos que têm certa nativa semelhança com o objeto a que se reportam.” [Eco, 1971:99] Muitas vezes, o ícone é lembrado como um signo que possui semelhanças materiais com o objeto, como numa pintura realista. De fato, “um signo pode ser *icônico*, ou seja, pode representar seu objeto principalmente por similaridade”. Mas na verdade, “qualquer coisa tem condições de ser

¹⁶ Representamen para Pierce é um signo, “é algo que, sob certo aspecto ou de algum modo, representa alguma coisa para alguém.” [Peirce, 1975:94]

um *Substituto* de qualquer coisa que se assemelhe” [Peirce, 1975:116]; portanto ícones são mais do que “signos que operam pela semelhança de fato entre suas qualidades, seu objeto e seu significado” [Plaza, 1987:22]. Esta semelhança não é necessariamente de natureza substantiva. Ela pode se revelar através de alguma outra forma de relação análoga que possui com o objeto; por exemplo, uma expressão algébrica, um mapa, um esquema. Mas quando o ícone for representado segundo uma semelhança substantiva, “amplamente convencional em seu modo de representação; ... em si mesma, sem legenda ou rótulo” [Peirce, 1975:116], então ele é chamado de hipo-ícone¹⁷.

Um exemplo de hipo-ícone¹⁸ são as imagens, cuja característica é assemelhar-se à aparência do objeto. Neste caso, a associação é direta. Ao olharmos uma imagem percebemos imediatamente a que objeto ela se reporta. Já os diagramas são exemplos de hipo-ícones de segundo nível, porque representam, na verdade, as relações que existem entre as partes de um objeto, como, por exemplo, a planta arquitetônica de uma casa. Finalmente as metáforas são hipo-ícones que representam um objeto por meio da representação de outro, mas que possui com o primeiro alguma semelhança. Por isso são chamados hipo-ícones de terceiro nível.

Um índice “é um signo que se refere ao objeto que denota em razão de ver-se realmente afetado por aquele objeto.” [idem, 101] Por exemplo, uma pegada, que indica o objeto homem, ou uma fotografia, que tem correspondência ponto a ponto com o objeto representado, por ter sido formada pela projeção luminosa do objeto numa superfície sensível.

Já o símbolo “é um signo que se refere ao objeto que denota por força de uma lei” [idem, 102]. Há uma convenção. São exemplos, as palavras, os numerais, as siglas, os sinais, as marcas comerciais.

¹⁷ O hipo-ícone, para Peirce, não deixa de ser um ícone; é como uma espécie de ícone.

¹⁸ Na verdade, Peirce estabelece o que chama de “divisão grosseira” do hipo-ícones. Ele os classifica em três categorias: as *imagens*, que definimos quando a “qualidade de sua aparência é semelhante à qualidade de aparência do objeto que a imagem representa” [Santaella, 1983:65]; os *diagramas*, quando o que se representa são as relações análogas entre as partes; e as *metáforas*, “aqueles que representam o caráter representativo, de um Representamen, traçando-lhe um paralelismo com algo diverso.” [Peirce, 1975:117]

Neste estudo, interessar-nos-ia particularmente as metáforas, o hipo-ícone de terceiro nível. No entanto, as interfaces gráficas são representadas por toda espécie de signos. Especialmente os ícones. Provavelmente porque segundo Peirce, “o único meio de transmitir diretamente uma idéia é por via de um ícone.” [idem, 117] (Coincidência ou não, o termo *ícone* é fartamente utilizado dentro do universo computacional, como podemos atestar no capítulo seguinte)

São exemplos de ícones nas interfaces computacionais os pontos-quentes de interação, imagens, figuras e desenhos produzidos em computador e toda a sorte de diagramas, esquemas, barras, bordas que são utilizadas para estruturar o desenho da interface. Há ainda imagens indiciais capturadas e muitos símbolos, como flechas, tipografias, numerais etc. Uma interpretação possível, no entanto, e esta é a que nos interessa, é que toda espécie de signo, sem exceção, seja ele um ícone, um índice, um símbolo, ou ainda um hipo-ícone (imagem, diagrama, metáfora), é também sempre uma metáfora. E como tal, o signo é representado pela substituição de outro signo, pois na metáfora está sempre embutida um processo de tradução. Como se opera então esta substituição?

Como dissemos acima, no item ‘Orientação a objetos’, todo objeto pode ser visto também como um miniprograma. Tecnicamente falando, programas de computador executam um procedimento, uma operação, ou, em outras palavras, uma atividade qualquer. Como o objeto está inserido em um sistema composto por diversos outros miniprogramas (os dos outros objetos), o seu miniprograma define o papel que ele desempenha no sistema como um todo.

Levando em conta uma interpretação peirciana, podemos dizer que cada objeto é representado por dois signos: o código-fonte¹⁹ de seu miniprograma e o signo visual apresentado na interface. Esta interpretação é possível porque o código-fonte de um programa é também um ícone, da mesma forma que Peirce havia afirmado em relação às

¹⁹ Código-fonte é a seqüência (o texto) que contém as instruções escritas em uma linguagem de computador específica e que descreve o programa. Depois de salvar o código-fonte em um arquivo texto, ele deve ser traduzido, através de um programa compilador, para a linguagem que o computador entende. Se tudo correr bem, ou seja, se não houver erros lógicos, de sintaxe e de interpretação, o compilador vai criar um arquivo de código-objeto.

equações algébricas²⁰. Ora, assim como estas, o código-fonte mantém relações análogas com um problema do mundo real, portanto é um ícone também. Assim, no nível mais baixo de linguagem, que no nosso caso é o algorítmico, o objeto-problema está sendo representado pelo código-fonte de seu miniprograma. Mas no nível mais alto, o da interface, este objeto-problema está sendo representado pelo signo visual de um objeto que é diverso. Neste sentido, é uma metáfora. Como a tradução operada ocorre da substituição de um ícone (o programa) por outro ícone (na interface) no eixo paradigmático, esta metáfora tem analogia à metáfora-palavra de um enunciado verbal. Deste modo, o ícone que aparece na interface pode ser representado por qualquer signo visual, desde que possua de algum modo uma relação com o objeto-problema.

Por exemplo, na interface do processador de textos Microsoft Word, existe um ícone, que se pressionado, aciona o miniprograma que executa a seguinte operação: “criar um novo documento do aplicativo”. O ícone que representa o miniprograma na interface do aplicativo é a imagem de um papel em branco. Mas, no nível da linguagem computacional que o criou, este miniprograma continua a ser só um conjunto de instruções.

Temos então um caso particular de tradução intersemiótica: a substituição de um conjunto de instruções (que também é um ícone) por um signo visual da interface (que pode ser de qualquer espécie – ícone, índice ou símbolo). São dois níveis de linguagem computacionais que utilizam signos e códigos diferentes. Num nível inferior, segundo a classificação de Tanenbaum, temos uma linguagem computacional qualquer, que é a que constrói um aplicativo. Num nível superior, a linguagem visual representada na interface. O ícone da interface substitui um conjunto de instruções que seriam muito estranhas se tivessem que ser manipuladas diretamente pelo usuário. O ícone da interface, portanto, torna-se uma metáfora dentro do espaço de representação. Julio Plaza já dizia algo similar a respeito da utilização de pictogramas em videotextos: um "pictograma (extraído da projeção ótico-

²⁰ "Em verdade, toda equação algébrica é um ícone, na medida em que indica, por meio de signos algébricos (que em si mesmo não são ícones), as relações das quantidades em causa." IN: PEIRCE, Charles S. *Semiótica e Filosofia*. São Paulo, Cultrix & Edusp, 1975, p.119.

retiniana e normalizado), ilustrando o texto ou não, pode muito bem substituir o verbal (função metafórica) ou fornecer informação do todo pela parte (função metonímica).” [Plaza, 1986:119]

A imagem apresentada na interface é apenas uma metáfora para a operação realizada e, na verdade, poderia ser representada por qualquer signo visual que, de modo figurativo, representasse convenientemente a operação. Veremos a seguir que a seleção do ícone não é tão aleatória assim; em geral ela é feita criteriosamente.

1.4.5 A metáfora na interface gráfica

Em geral, um signo não aparece isoladamente em uma interface; há um conjunto de signos. Todos os objetos representados são signos. Na interface, tudo é signo. Até mesmo o fundo da tela, que, embora não pareça, cumpre um papel decisivo. Como diz Julio Plaza: “o que vemos numa imagem são relações de tensão entre figura e fundo que definem, junto com o objeto que a imagem referencia, aquilo que chamamos de ‘imagem’. Na maior parte das vezes, percebemos objetos representados e não vemos a superfície.” [Plaza, 1986:107]

Esta percepção de um signo como um objeto, em contraposição ao fundo, é fundamental nas interfaces computacionais, pois tem duplo papel: semiológico e algorítmico. Semiológico porque, como disse Julio Plaza, o receptor percebe que cada signo representa um objeto. E ao perceber o signo como um objeto o receptor fornece ao programador de sistemas as coordenadas necessárias para definir os limites do objeto algorítmico.

[O nível simbólico traduz o nível algorítmico]

Dissemos acima que o computador trabalha com níveis de linguagens computacionais. A interface é colocada num nível superior, que eu vou chamar de nível simbólico, pois é o nível percebido pelo usuário. Abaixo dele há um nível onde se coloca a linguagem computacional, a partir da qual a interface foi desenvolvida. Vou chamar de nível algorítmico. O

nível simbólico é portanto uma tradução do nível algorítmico. Do mesmo modo, no nível superior temos o objeto simbólico, que é o percebido na interface pelo receptor. Num nível abaixo, há o código-fonte, que é objeto algorítmico; para o usuário, este é somente uma abstração, pois não possui representação na interface. O projetista fecha a questão ao fazer com que o objeto simbólico coincida com o objeto algorítmico. Isto depende da seleção adequada de um signo visual que substitua o miniprograma, pois o objeto simbólico é uma tradução do objeto algorítmico. Aquele é a metáfora deste.

Mas os objetos não estão isolados numa interface. Há diversos objetos. O modo como eles estão estruturados permite ao usuário compreender uma qualidade que é maior do que a simples soma deles. Como afirma Julio Plaza:

“A organização refere-se a boa ordem das tensões entre os elementos gráficos que, por sua estrutura, dão como resultado um todo. A mensagem é completa e fechada na medida em que todos os elementos estão inter-relacionados, seguindo a premissa gestáltica: ‘O todo é mais qualitativamente diferente do que a soma das partes.’ A percepção de elementos separados carece de sentido, pois elementos separados estão descontextualizados e não participam do todo como qualidade: esta qualidade é percebida como um todo quando se apresenta a uma mesma consciência.” [idem, 158]

Esta qualidade percebida no todo é o significado da interface. A soma das partes é qualitativamente diferente do todo porque os objetos interagem. Como diz Dondis, à respeito de imagens em geral, “relacionamos interativamente esses elementos; temos em vista um significado.” [Dondis, 1991:30]. Neste sentido, a interface se aproxima do estruturalismo:

“É a seu propósito, efetivamente, que o estruturalismo (e não a estrutura) encontra a sua primeira definição: em oposição ao atomismo, que isola termos cujo conjunto é simplesmente a sua justaposição, ele consiste em procurar as relações que dão aos termos que unem o valor ‘de posição’ em um conjunto organizado, em apreender conjuntos que sua articulação torna significantes. O estruturalismo implica, pois, duas idéias: a de totalidade e a de interdependência.” [Pouillon, 1966:10]

A interdependência dos objetos acontece no nível simbólico e algorítmico. No primeiro porque, como afirma Plaza, existe uma tensão presente entre os elementos gráficos (os objetos), de modo que uma mensagem é enviada para o usuário a partir da sintaxe da interface, composta por seus objetos.

Algorítmico, porque, segundo o paradigma estabelecido pela ‘programação orientada a objetos’, os objetos se relacionam e podem interferir na qualidade de outro objeto. Tecnicamente, a qualidade de um objeto é chamada ‘propriedade’. Cada objeto se comunica com outro a partir de mensagens que são enviadas quando algum evento ocorre no sistema. São estas mensagens que permitem um objeto alterar a qualidade/propriedade de outro.

De uma certa maneira, cada objeto tem um papel a cumprir no Espaço de Soluções. O que define este papel é o miniprograma instalado internamente no objeto. É uma espécie de código genético dele, que fica em estado de inércia até que um evento ocorra e permita que ele afete a qualidade de outro objeto, de si mesmo ou do sistema como um todo. Os signos que representam os objetos, portanto, não são imutáveis. São signos de interação. Isto é fundamental, porque permite que a interface seja mais do que simplesmente uma representação gráfica estanque. Ela passa a ser também o simulacro de um modelo da realidade: um ambiente, um veículo, uma ferramenta etc. Se os objetos fossem signos estáticos, as interfaces seriam apenas uma imagem. A possibilidade de alteração do estado interno dos objetos dão vida à interface. E ela adquire funcionalidade.

Há uma dicotomia formada nos dois pólos. De um lado, a representação estática vista por um receptor passivo. Do outro, a interface interagida pelo usuário. Além de ser um ‘signo para ver’²¹ o objeto torna-se também um ‘signo para interagir’. O receptor deixa de ser passivo e passa a se tornar ativo. As interfaces são metáforas de ambientes porque o usuário interage. No entanto, os dois pólos coexistem. O caráter funcional, de interação, da interface deve coincidir com uma representação simbólica apropriada. Júlio Plaza afirma:

“... quando se trata de mensagens (como organização física de signos) utilitárias, a sua função simbólica, de produção de significado, é significar seu uso. Concordamos com Gillo Dorfles quando no diz: ‘Trata-se de uma classe de simbolismo que poderíamos definir como ‘funcional’; de um simbolismo que se identifica, com a funcionalidade do objeto.’” [Plaza, 1986:167]

²¹ Julio Plaza já havia utilizado esta definição à respeito dos signos gráficos: “O gráfico permite uma leitura rápida, uma melhor visualização e memorização, assim como ajuda a desenvolver idéias complexas com simplicidade. São signos para ver.” [Plaza, 1986:93]

Podemos dizer que a funcionalidade do objeto se situa no seu objeto algorítmico, enquanto que sua representação se situa no seu objeto simbólico. Do mesmo modo o caráter funcional do sistema encontra-se na soma de seus códigos, no algoritmo geral do sistema, no nível algorítmico. E o caráter simbólico é a interface que visualizamos. “O modelo conceitual de um programa integra a interface (com o usuário) aos recursos oferecidos pelo software, combinando forma e funcionalidade.” [Heckel 1993:153] Podemos concluir finalmente que o objeto é simbólico porque está representado numa interface cheia de significação. Do mesmo modo, o objeto algorítmico só é operacional na funcionalidade geral do sistema. Como diz Patrick J. Lynch e Sarah Horton:

“The graphic user interface (GUI) of a computer system comprises the interaction metaphors, images, and concepts used to convey function and meaning on the computer screen. It also includes the detailed visual characteristics of every component of the graphic interface and the functional sequence of interactions over time that produce the characteristic look and feel.”
[Lynch, 1999:11]²²

Por exemplo, nos sistemas operacionais atuais, há um miniprograma destinado a liberar um determinado espaço de memória para que possa ser ocupado por outros dados do sistema. Na linguagem habitual, afirmamos que estamos apagando um arquivo quando executamos esta operação. No nível simbólico este miniprograma é representado pelo seu correspondente objeto simbólico, o ícone de uma lixeira. A sua representação em si é apenas o ícone de uma lixeira. Mas o ícone faz parte de uma interface que é simbolizada por um tampo de escrivaninha e que é composta por diversos outros objetos. Quando arrastamos um documento (um objeto) até o ícone da lixeira (outro objeto) há um movimento de interação entre eles. Ou seja, o que faz com que o ícone da lixeira adquira um *status* especial de metáfora é também o fato de fazer parte de um sintagma visual.

Assim, se a tradução do objeto algorítmico pelo objeto simbólico nos leva a crer que há um movimento de substituição paradigmático típico da metáfora-palavra, o fato deste objeto simbólico (ícone da lixeira) estar inserido numa interface de um tampo de escrivaninha, construída pela composição de diversos outros objetos que interagem, nos faz acreditar que

²² T.A.: “A interface gráfica com o usuário (IGU) de um sistema computacional compreende metáforas interativas, imagens, e conceitos utilizados para transmitir funcionalidade e significado para a tela do computador. Isto inclui também os detalhes visuais característicos de todo componente da interface gráfica e a seqüência funcional de interações no tempo que produz o olhar e sentimento característico.”

há aí também um movimento que se opera no eixo sintagmático e que portanto se confunde com a metáfora-enunciado.

Vamos relembrar a sua formulação: na metáfora-enunciado, o sentido metafórico é conduzido não pela palavra, mas pela sentença como um todo, tendo a estrutura do enunciado papel preponderante na formação figurativa. Max Black definiu a base de sua metáfora-enunciado a partir do que chamou teoria da interação, segundo a qual o sentido metafórico é formado a partir da interação entre um sujeito e um predicado lógicos. Segundo este pressuposto, um termo do sintagma só adquire valor porque se opõe aos outros termos da sentença.

Na interface computacional o processo de formação da metáfora é estabelecida a partir de um modelo semelhante. A diferença é que em vez de um enunciado, temos o sintagma visual, representado pela interface. E ao contrário de palavras, temos um conjunto de signos visuais, representantes dos objetos, que se inter-relacionam. Os eventos através dos quais os objetos interagem, os signos calculadamente escolhidos para representar os objetos algorítmicos, a forma como os objetos se relacionam, tudo isto define uma metáfora interfacial que representa um modelo de algo que vemos no mundo real.

Ao adotar um modelo para a representação da interface, os projetistas mais uma vez estão utilizando de um processo de tradução: de uma interface baseada em elementos abstratos, manipulados por linhas de comando, teclas de funções e atalhos de teclado para a manipulação direta de signos visuais que representam objetos de nosso dia-a-dia. É a tradução de um sistema de representação e manipulação que possui uma linguagem abstrata por outra de caráter gráfico-visual.

Como vimos, Tanenbaum também adotou esta linha de pensamento ao descrever cada sistema como uma máquina virtual. Para o autor, há uma “relação importante entre uma linguagem e uma máquina virtual” [Tanenbaum, 1992:3]: assim como uma máquina é a definição de uma linguagem, uma linguagem define uma máquina.

Nos dois níveis, o algorítmico e o simbólico, há a definição de uma máquina virtual. Os dois níveis possuem suas respectivas linguagens. Na tradução de uma linguagem de computador, que não é compreendida pela maioria dos usuários, para uma linguagem gráfica, o programador de sistemas escolhe um modelo que torne mais clara a comunicação homem-máquina. A linguagem simbólica, que utiliza elementos gráficos, é uma face metaforizada da linguagem algorítmica, que utiliza código-fonte.

1.4.6 A popularização da adoção de metáforas nos computadores

Uma prova histórica das vantagens da utilização de metáforas foi dada pelos primeiros computadores Macintosh que adotaram o modelo de representação criado pela Xerox: a chamada metáfora do desktop (tampo de escrivaninha), hoje implementada em praticamente todos os ambientes operacionais. A partir deste momento, muitos projetistas passaram a adotar a metáfora como uma solução na criação das interfaces dos aplicativos. À este respeito, Joy Mountford, pesquisador da Apple, afirma: “... about everyone at Apple knows the phrase ‘desktop metaphor’ and fervently believes that a good metaphor is essencial to an easy-to-use human interface.”²³ [Laurel, 1992:27]

[O “desktop” do Macintosh]

Outro colaborador da Apple a apresentar as vantagens da utilização da metáfora em interfaces é Thomas D. Erickson. Em seu artigo, *Working with Interface Metaphors*, ele sugere que a utilização de metáforas em interfaces é praticamente uma técnica:

“Metaphor is an integral part of our language and thought. It appears not only in poetry and novels, but in our everyday speech about common matters. Mostly we don’t notice; metaphor is such a constant part of our speech and thought that it is invisible (...) The characteristics of metaphor in our language are the same ones that govern how metaphor works in an interface.”

²³ T.A.: “... quase todos na Apple conhecem a frase ‘metáfora do tampo de escrivaninha’ e acreditam fervorosamente que uma boa metáfora é essencial para uma interface amigável.”

*Just as metaphors invisibly permeate our every speech, so do they occur throughout the interfaces we use and design.” [idem, 65]*²⁴

Esta maneira de desenvolver interfaces computacionais a partir de metáforas passou a fazer parte do universo comum dos desenvolvedores de aplicativos. Em seu livro ‘Software Amigável’, Paul Heckel chega a listar algumas vantagens do uso da metáfora. Para o autor, as metáforas tornam as coisas mais familiares, pois “os objetos e as regras não são coisas arbitrárias a serem memorizadas, mas têm associações com o mundo real que auxiliam o pensamento.” [Heckel, 1993:160]; elas fornecem um mapa objetivo da realidade, já que “ao estabelecer uma analogia com o mundo real, tanto o usuário quanto o projetista podem trabalhar usando um mapa comum” [idem, ibidem]; elas facilitam a comunicação, “ao evocar um modelo mental comum a duas pessoas, uma metáfora fornece uma estrutura para a comunicação humana (...) depois que o usuário tiver compreendido a metáfora plenamente, ele será levado a estruturar as aplicações dentro dessa metáfora sempre que começar a pensar sobre elas” [idem, ibidem]; elas simplificam as coisas, pois “... uma boa metáfora é aquela que fornece um modelo conceitual simples (...) para as quais podem ser transportados problemas complexos.” [idem, 161]; elas unificam, ao “...fornecer uma estrutura integradora que permite avaliar classes de problemas.” [idem, ibidem]; elas auxiliam a invenção e assim “...beneficiam tanto o projetista quanto o usuário” [idem, ibidem]; e finalmente as metáforas criam limites: “...os limites de uma boa metáfora definem padrões naturais...” [idem ibidem]

Paul Heckel afirma que “uma metáfora não é útil apenas por comunicar informações, ela também é útil por causa da estrutura que fornece e das associações que permite fazer. Tudo isso possibilita que o público utilize um conhecimento que já possui.” [idem, 150].

Outro autor a ter esta percepção foi Theo Mandel: “*Metaphors aid designers because adoption of a metaphor allows them to structure aspects of the target system or interface in*

²⁴ T.A.: “a metáfora é parte integral de nossa linguagem e pensamento. Ela aparece não somente na poesia e em novelas, mas em nossa fala do dia-a-dia em assuntos comuns. A maioria nem percebe; a metáfora é uma parte constante de nossa fala e pensamento, que é invisível (...) As características da metáfora em nossa língua são as mesmas que governam o modo como elas trabalham na interface. Assim como metáforas invisíveis permeiam toda nossa fala, do mesmo modo elas ocorrem por toda as interfaces que usamos e desenhamos.”

*terms of familiar and commonly understood aspects of the source domain.”*²⁵ [Mandel, 1997:25]

Por conta disto, Paul Heckel chega a fazer uma classificação dos tipos de metáforas adotadas em interfaces computacionais:

‘Nós classificamos as metáforas de computadores em metáforas de familiarização, que facilitam a aprendizagem de um programa, e em metáforas de transporte, que fornecem uma estrutura para a qual o usuário pode transportar vários problemas de modo a pensar sobre eles e sobre a maneira de controlá-los melhor.’ [idem, ibidem]

Lembremos que Dondis já havia alertado para a importância da estrutura na formação do significado presente nas representações visuais: “... tudo o que vemos e criamos compõe -se dos elementos visuais básicos que representam a força visual estrutural, de enorme importância para o significado e poderosa no que diz respeito à resposta.” [Dondis, 1991:22]

Algo a se considerar é que a utilização das metáforas de estrutura não se restringiu aos chamados aplicativos. Elas invadiram também as linguagens de desenvolvimento. No primeiro caso, as metáforas, ao fornecer a estrutura do aplicativo ao usuário, tornam a interação mais familiar. No segundo caso, no dos programas de desenvolvimento, o usuário é também projetista. É usuário de um software que tem como objetivo o desenvolvimento de outro aplicativos.

Existe de maneira subliminar uma metáfora nas ferramentas de desenvolvimento de sistemas hipermídia, sejam elas simplesmente softwares de apresentação, como o PowerPoint, o Macromedia Flash ou o Adobe PageMil; softwares de autoria, como o Director da Macromedia e o ToolBook da Asymetrix; linguagens de marcação, como o HTML; linguagens visuais, como o Visual Basic ou o Delphi e as linguagens de alto nível mais convencionais, como o C++ ou o Java. Cada uma delas tem a sua própria metáfora ‘interna’. Todas elas usam programação orientada a objetos, em que cada objeto nativo do

²⁵ T.A.: “As metáforas ajudam os projetistas porque a adoção de uma metáfora permite a eles estruturar o s aspectos do sistema em questão ou da interface em termos de aspectos familiares e comumente conhecidos do modelo original.”

software possui relacionamento com os outros objetos do sistema. Esta forma estrutural de relacionamento gera uma metáfora interna.

O PowerPoint usa a metáfora de uma apresentação audiovisual, com uso de retroprojetores; o Flash, assim como o Director, usa a metáfora de um filme, e o ToolBook, a de um livro; o HTML usa a metáfora das plantas de diagramação; as linguagens visuais, como o Visual Basic, usam a metáfora de formulários para uso de banco de dados e assim por diante.

Vamos analisar agora o estudo de dois casos bem particulares, o do ToolBook e do Director. Os dois não escondem a adoção de uma metáfora de estrutura; pelo contrário, eles chegam a explicitá-la em seus manuais de orientação do software. Isto é feito até as últimas conseqüências, a ponto mesmo de influenciar na cara do aplicativo hipermídia desenvolvido a partir destas ferramentas.

1.5 Estudo de caso: os softwares de autoria

O desenvolvimento de um aplicativo muitas vezes requer a utilização de uma linguagem de programação tradicional, como C, Basic, Pascal etc. Um bom exemplo é o sistema operacional Windows, criado com C. Uma das vantagens na adoção destas linguagens é a flexibilidade: elas permitem construir qualquer tipo de produto, na forma que o projetista desejar. Mas tudo deve ser construído a partir do zero, cada ferramenta, cada rotina, o que torna o código do programa quase exclusivo para aquele aplicativo. Estas linguagens são muito abertas, mas requerem um nível alto de aprendizado e um processo extremamente demorado e oneroso de desenvolvimento.

De um certo modo, uma apresentação audiovisual, só para ficar num exemplo, poderia ser programada com a linguagem C. Sem dúvida, o produto final seria inédito em todos os seus aspectos: tanto no que diz respeito à sua interface, quanto na parte lógica e operacional. Mas a troco de que gastar tanto tempo e dinheiro se já existem no mercado aplicativos específicos para este tipo de apresentação, como por exemplo o PowerPoint, da Microsoft ou o Flash, da Macromedia? São aplicativos fechados, mas dirigidos a este gênero de produto e muito mais fáceis de operar. Costuma-se dizer que algo feito em PowerPoint fica

com “cara” de PowerPoint e algo feito em Flash, com “cara” de Flash. Mas o projetista prefere pagar este preço do que ter que gastar tanto tempo e dinheiro, desenvolvendo um produto a partir de uma linguagem tradicional.

Paralelamente às apresentações audiovisuais, o projeto de aplicativos hipermídia passa por este mesmo questionamento. Existem algumas ferramentas, chamadas de maneira geral “softwares de autoria”, que foram especialmente criadas para o desenvolvimento deste gênero de aplicativo, particularmente aqueles que são distribuídos em CD-ROM. São *softwares* especialistas e fechados, mas que já possuem embutidas uma série de ferramentas de produtividade. Certamente um CD-ROM hipermídia possui uma complexidade muito maior do que uma simples apresentação do gênero PowerPoint, o que justificaria, neste caso sim, a adoção de uma linguagem mais poderosa, do tipo C. Mas isto seria mais conveniente no caso da construção de aplicativos mais complexos, como uma Enciclopédia Virtual, que trabalha com uma enorme base de dados e necessita de condições pesadas de execução. As ferramentas de autoria são neste sentido uma alternativa, pois embora também possuam uma zona de codificação (por script), muitos elementos podem ser construídos por manipulação direta da interface.

As ferramentas de autoria partem de um metáfora básica. Vamos analisar dois softwares clássicos desta área: o Director da Macromedia e o ToolBook da Asymetrix.

O Director da Macromedia é uma ferramenta muito popular na construção de títulos hipermídia e na criação de animações digitais em duas dimensões. A codificação segue o paradigma de orientação a objetos. Mas o Director é montado dentro de uma estrutura que força os seus objetos a se relacionarem de uma maneira análoga ao de uma montagem teatral ou cinematográfica. Costuma-se dizer então que o software da Macromedia usa a metáfora de uma peça de teatro (ou de um filme). Toda a terminologia do Director é baseada nos jargões destes dois gêneros. Assim, o projeto final de um trabalho desenvolvido por ele é chamado de filme (*movie*) e para rodá-lo existe um painel de controle (*Control Panel*) com botões típicos de um videocassete (*play, pause, stop* etc). Os objetos simbólicos do Director são chamados ou representados como atores (*cast members*), que desempenham seus papéis (*sprites*) no decorrer do andamento do filme

(*movie*). A janela principal, onde é montada a interface, é chamada de palco (*stage*). E há ainda uma espécie de partitura que funciona como uma linha do tempo, tipicamente representada por uma janela chamada *score*. Ela é dividida em numa espécie de tabela, em que as colunas representam os quadros (*frames*)

ou cenas do filme, e as linhas representam os canais, que também apontam para as camadas (as camadas indicam quem está na frente ou atrás no palco). Numa cena, cada canal é ocupado por um dos atores (objetos) do filme (projeto). Estes objetos podem mudar um das suas propriedades (cor, posição, tamanho, deformação, giro, textura etc), mudar de canal ou simplesmente sumir, à medida que o filme roda, pois o Director é orientado pelo tempo. Isto dá a sensação de movimento, exatamente como no modelo empregado no cinema, segundo o qual temos a sensação de movimento quando visualizamos diversos quadros ligeiramente diferentes à medida que o tempo corre. A partir de uma determinada velocidade (em *frames* por segundo) não percebemos mais a existência de quadros individualizados, mas apenas um filme como uma entidade una. Costuma-se justificar este fenômeno em função de um estado psicológico chamado *pregnância do olhar*. No entanto, o Director não serve apenas à animação; ele também se presta ao desenvolvimento de sistemas interativos, o que o força a criar mecanismos de paradas no filme para que o usuário possa interagir com os elementos componentes da interface. Isto o distingue de uma ferramenta para criação de animações convencionais, mas não exclui de o colocar dentro uma estrutura que usa a metáfora de um teatro ou de um filme.

Outro exemplo de software de autoria é o ToolBook da Asymetrix. O ToolBook usa a metáfora de um livro e assim como o Director tem toda a sua terminologia orientada por ela. Assim, o *software* da Asymetrix é dividido em páginas que possuem fundos (*backgrounds*) que podem ser compartilhados por mais de uma página. O espaço de uma

página coincide com a interface com o usuário e o projeto final é chamado de livro (*book*). O método para virar uma página (ou seja, o modo como o usuário pode “navegar” entre os diversos pontos do sistema hipermidiático) é através de comandos de sua linguagem interna, chamada tipicamente como OpenScript. Os comandos em questão são o *go to page* (vá para a página) e *flip* (mover rapidamente, folhear ou virar – uma página no caso). O projetista do aplicativo hipermídia é denominado “autor” e o usuário é o “leitor”, nomes que inspiraram também a denominação dos dois modos de interação do software: o de desenvolvimento, chamado “modo de autor” e o de testagem, chamado de “modo de leitor”. O que difere o ToolBook de um livro convencional é que além de aceitar textos e imagens, ele também agrega outros recursos como vídeo ou áudio. Isto não elimina a metáfora de estrutura empregada no software, pois grande parte do modo lógico e operacional do ToolBook é baseado no modelo de construção de um livro.

O interessante nestes dois casos é que a adoção de uma metáfora de estrutura faz algo mais do que simplesmente facilitar o processo de desenvolvimento de aplicativos hipermídia. Ao conhecer o processo construtivo de um filme ou de um livro, o autor tem um modelo natural a se inspirar e se sente mais à vontade no processo de construção de um similar digital. Por outro lado, tanto um projeto feito com Director quanto um feito com ToolBook ficam com a “cara” de seu modelo. Projetos em Director costumam ser mais dinâmicos, mais “animados”; os de ToolBook têm sempre um caráter enciclopédico, com mecanismos de busca precisos.

Importante considerar também o fato de que embora cada software tenha sua metáfora de estrutura, os dois se prestam a construir a mesma coisa, ou seja, sistemas interativos hipermídia. Um dado que merece destaque nesses dois casos é o caráter não-linear do hipertexto, que

nos fornece outra estrutura, só que esta de caráter variado: a estrutura de interação, ou, como costuma-se dizer popularmente, de “havegação”. Lucia Leão caracteriza o hipertexto como uma estrutura composta por:

“...blocos de informação e por vínculos eletrônicos (*links*) que ligam estes elementos. Os blocos de informação costumam ser denominados *lexias*. O termo *lexia* foi empregado anteriormente por Barthes para designar blocos de textos significativos. Esse vocábulo foi retomado por Landow como sendo o ponto onde se está antes de seguir um *link*. Outros autores preferem usar a denominação *nó*. De qualquer forma ambos os termos correspondem às unidades básicas de informação. Uma *lexia* pode ser formada por diferentes elementos, tais como textos, imagens, vídeos, ícones, botões, sons, narrações etc.” [Leão, 1999:27].

A autora ainda destaca que os *lexias* podem adquirir vários nomes dependendo do tipo de metáfora empregada pelo software de desenvolvimento. No Director, por exemplo, ele é chamado de *frame*. No ToolBook, de página (*page*). Em outras ferramentas de autoria ocorre o mesmo processo. O HyperCard chama o seu *lexia* de *card*, pois apresenta a metáfora de uma pilha de cartas – o projeto é denominado *stack*. No IconAuthor, da Asymetrix, o projeto é construído dentro de um fluxograma e cada *lexia* é chamado de *icon*. A mesma concepção tem o Authorware, da Macromedia, pois o projeto é estruturado em uma espécie de *storyboard*. O Flash, que também é um software baseado no tempo, chama o seu *lexia* de *frame*. Acrescento ainda o PowerPoint, que, embora não seja exatamente uma ferramenta de autoria, tem a capacidade de desenvolvimento de sistemas não-lineares. Nele, o *lexia* é chamado de *slide*, pois o programa utiliza a metáfora de uma apresentação audiovisual.

1.6 Conclusão do capítulo

Em princípio, a metáfora só poderia se estabelecer em uma linguagem que contivesse um código extremamente forte e estabelecido, como o da linguagem verbal. Seus signos, que neste caso são as palavras, possuem significados muito mais precisos do que os signos de certas expressões da linguagem visual. Só assim, a partir de uma codificação mais rígida, poderíamos sentir o chamado estranhamento, citado pelos autores da Teoria da Metáfora, quando ouvimos uma frase do tipo “fulano de tal é um touro”.

A experiência tem mostrado, no entanto, que a aplicação da metáfora na linguagem visual é também possível, nas suas mais diversas formas de expressão. E isto é tão verdade que o conceito de metáfora vem sendo utilizado indiscriminadamente para identificar figurações em diversas manifestações humanas. A informática é apenas mais um exemplo.

Na computação, o histórico de utilização de metáforas acompanhou o processo evolucionário das interfaces. Nas interfaces orientadas por linha de comandos, as metáforas apareceram no processo de formação de palavras, dando nomes a dispositivos e comandos. Mais adiante, as metáforas tiveram papel preponderante no quadro formador das interfaces gráficas, primeiro nos sistemas operacionais e mais tarde nos aplicativos, vide o caso clássico da metáfora do tampo de escrivaninha (*desktop metaphor*) da Xerox Parc, depois adotado nos Macintoshs e mais tarde ainda no Windows da Microsoft. Hoje as metáforas desempenham também o seu papel na criação das interfaces de sistemas interativos hipermídia.

Assim podemos, de algum modo, enxergar um paralelo na aplicação das metáforas computacionais e o que temos visto sendo tipificado historicamente na linguagem verbal. Se alguns autores acreditam na existência de uma base teórica de classificação deste tropo, que a divide em metáfora-palavra e metáfora-enunciado, podemos constatar a ocorrência desta mesma divisão no correspondente metafórico das interfaces. A diferença é que, em vez de enunciados, temos o sintagma visual, representado pela interface. Em vez de palavras, temos os signos visuais, representantes dos objetos.

No primeiro caso, se na metáfora-palavra há uma ação de substituição de um termo por outro similar, que empresta ao corpo frásico um novo sentido, na interface computacional este movimento paradigmático se efetua numa tradução intersemiótica: de um ícone, que nada mais é do que o código-fonte do objeto algorítmico, pelo signo visual, que representa um dos objetos simbólicos apresentados na interface. O projetista completa o processo ao fazer com que o objeto algorítmico coincida com o objeto simbólico. Isto é feito a partir do momento em que o agrupamento de dados que, em conjunto, representam o objeto algorítmico sejam atrelados ao signo visual que representa o objetos simbólico. Se representados de modo conveniente para o usuário, o signo visual torna-se uma metáfora no

Espaço de Soluções. É como se cada objeto simbólico fosse uma entidade quase viva, cujo código genético seria representado pelo seu código-fonte. O objeto do nível simbólico pode se transformar, de acordo com suas regras internas, que é o seu algoritmo individual.

Por outro lado, a metáfora computacional tem também a sua interpretação na corrente de pensamento chamada metáfora-enunciado. Neste caso, a metáfora computacional está para a interface e seus signos visuais assim como a metáfora-enunciado está para a sentença e as palavras. Os signos visuais, numa leitura estática, seriam apenas representações de objetos reais. E o conjunto destes signos é simplesmente a representação de um ambiente. Mas o ambiente representado numa interface é mais do que simplesmente um quadro estanque. Ele é uma Máquina virtual. Colocados na interface, no Espaço de Soluções, estes signos se relacionam, se interagem, pois há ações associadas entre eles. É a partir desta interação, no eixo sintagmático, apoiado na metáfora individual que cada objeto simbólico possui no eixo paradigmático, que se forma o significado metafórico da interface.

Por outro lado, a boa escolha de uma metáfora interfacial fornece uma estrutura de representação que o usuário já conhece e portanto poderá controlar melhor; pode ser um ambiente, um mapa, uma ferramenta, uma máquina etc. Ela traz à mente do interagente todo um conjunto de relações que existe entre os objetos que compõem a interface, toda uma série de procedimentos e ações que fazem parte do modelo representado e que, portanto, são familiares. A maioria dos aplicativos de hoje procura ser desenvolvido a partir de um modelo de representação, que torne o usuário confortável e familiar em sua interação. A adoção de uma metáfora é, portanto, antes de mais nada, um processo de tradução necessário. De uma linguagem para outra. É a substituição de uma máquina virtual por outra mais conveniente. Neste sentido, são metáforas; traduzem um conjunto de instruções de uma determinada linguagem computacional para outra mais amigável, mais direta, mais quente e mais visual.

— CAPÍTULO 2 —

Uma Amostragem do Uso de Metáforas no Computador

2.1 Introdução

De um lado temos o computador, representante de uma tecnologia de informação praticamente consolidada no dia-a-dia de significativa parcela da população mundial. No entanto, uma máquina que nos seus primórdios era muito pouco amigável, praticamente restrita a pesquisadores e cientistas, habituados à sua linguagem, que, na época, era muito técnica. De outro lado, temos a metáfora, um dos tropos mais utilizados na nossa linguagem, como recurso estilístico ou retórico, mas que também pode ser utilizada em função de outra necessidade: a de facilitar, no sentido de traduzir, um ato comunicativo. Quando empregamos uma metáfora, nossa intenção pode ser outra: a de tornar mais claro um enunciado de difícil compreensão imediata.

As metáforas podem ser empregadas para suprir deficiências lingüísticas decorrentes de diferenças sociais, culturais ou técnico-científicas entre um emissor e um receptor. O que está em evidência nessa pesquisa é o abismo lingüístico que existe numa relação de comunicação mediada por computador, que tem de um lado os técnicos, engenheiros e cientistas da computação, e do outro, o usuário comum. E que tem a própria linguagem do computador de um lado e as linguagens humanas do outro. Essa situação nos leva a crer que a aplicação tão intensa da metáfora na computação se dá em função da necessidade de facilitar/traduzir um universo tão estranho para a maior parte das pessoas.

Em muitos casos, o emprego de certas metáforas foi tão forte que praticamente acabou por dar nomes a certos termos do universo computacional. Por exemplo, podemos citar os casos de memória, arquivo ou pasta. É o que Filipak, baseado em estudos de Aristóteles, Ricoeur e outros teóricos, chama de metáfora lingüística ou denominativa e que também classifica como de natureza denotativa. “A denominação é um processo que nomeia as coisas do mundo dos objetos ou substitui nomes imprecisos pela ação da metáfora-palavra. Seu caráter é referencial, relacionando-se com o mundo dos objetos.” [Filipak, 1983:51]

Vimos acima que além de sua natureza denotativa, a metáfora também passou a ser empregada na Informática com funções figurativas, que Filipak proclama como a natureza

conotativa da metáfora: “A linguagem figurada é essencialmente constituída pela função conotativa, a função poética... o aspecto semântico das figuras, que operam na transferência de sentido, tanto no polo da similaridade, como na da contigüidade.” [idem, 87]. É o que podemos ver numa certa medida no emprego da metáfora do *desktop*, nos sistemas operacionais modernos.

O objetivo deste capítulo é o de mostrar a existência de uma certa tradição em se utilizar metáforas no ambiente que cerca a computação. Apresentaremos uma pequena amostragem do uso da metáfora na informática; veremos como este tropo vem contribuindo para traduzir ao público leigo aquilo que para ele é tão difícil de compreender em função da complexidade técnica do computador. O capítulo foi organizado segundo uma ordenação linear-histórica, que percorre as origens do computador e chega quase nos dias de hoje, anterior ao surgimento dos programas navegadores da Internet (discutiremos este tópico em capítulo posterior).

2.2 A história do computador: metáforas, analogias e simulações²⁶

Talvez a própria origem do computador esteja ligada a uma concepção metafórica. No século passado, o professor de matemática da Universidade de Cambridge, Charles Babbage (1792-1871), após inventar o velocímetro, desenvolveu entre os anos de 1822 e 1834 um projeto de uma máquina diferencial. Este dispositivo mecânico tinha como objetivo principal auxiliar no cálculo de tabelas para a navegação, além de realizar operações simples, como soma e subtração.

Para construir e aperfeiçoar sua máquina diferencial, Babbage recebeu até mesmo financiamento do governo inglês, que mais tarde desistiu do projeto, após o matemático ter gasto enormes quantias. No entanto, apesar do dispositivo funcionar razoavelmente bem, o

²⁶ Fontes para este tópico: Andrew S. Tanenbaum. Organização Estruturada de Computadores. Rio de Janeiro, Editora Prentice-Hall do Brasil, 1992, p.10-20. Mário A. Monteiro. Introdução à Organização dos Computadores. Rio de Janeiro, LTC, 1992, p.5-17. Fernando de Souza Meirelles. Informática: Novas Aplicações como Microcomputadores. São Paulo, Makron, 1994. Larry Gonick. Introdução Ilustrada à Computação. São Paulo, Itautec, 1983.

inglês não se sentia satisfeito. Talvez porque, como diríamos nos dias de hoje, esta máquina executava somente um algoritmo.

Por conta disso, Babbage começou a gastar sua fortuna (e a de outros) no projeto da sucessora da sua máquina diferencial: a máquina analítica. Sua intenção era fazer dela um dispositivo de uso geral. Provavelmente inspirado nos teares mecânicos de Jacques Jacquard, ela deveria executar tarefas codificadas em cartões perfurados. Cada cartão corresponderia a um diferente programa.

O pesquisador chegou a contratar até mesmo uma “programadora” de cartões perfurados: Ada Lovelace Byron, filha do poeta britânico Lord Byron. Romântica como o pai, a condessa acreditou no projeto, que por fim foi um fracasso. Porém, Ada deixou para a informática um inquestionável legado: ela desenvolveu instruções de computador correspondentes aos saltos condicionais e aos laços (ou *loops*) e criou o conceito de subrotina.

A máquina analítica possuía quatro componentes: o armazenamento (memória), o engenho (unidade de cálculo), a entrada para matéria-prima (o leitor de cartões perfurados) e o produto final (saída para um dispositivo perfurador de cartões ou para uma impressora). Assim, embora estivéssemos no avançado século XIX, foi na Idade Média que Babbage se inspirou. A sua máquina tinha o funcionamento análogo ao de um moinho. Infelizmente, o matemático inglês trabalhou até 1871 sem obter grande sucesso nesta empreitada. Sua máquina estava muito a frente do tempo. Precisava de milhares de dentes, engrenagens e rodas numa escala que não era condizente com o século XIX. Morreu pobre, mas deixou milhares de desenhos com detalhes de projeto, conservados também pelas anotações de Ada L. Byron.

Faltam registros para dizer se de fato Babbage se influenciou ou não por uma metáfora – a do moinho. No entanto, até hoje, os computadores possuem uma estrutura similar à máquina analítica. Assim como o antigo dispositivo, têm quatro unidades básicas – a unidade central de processamento, a memória e os dispositivos de entrada e saída. Na sua relação sistêmica, os computadores modernos funcionam do mesmo modo.

A idéia do inglês foram a base para o passo seguinte da evolução do computador, dado por Herman Hollerith. Quando o governo dos Estados Unidos promoveu um concurso para a criação de um sistema mais rápido de contagem e tabulação para o Censo de 1890, este norte-americano pôs-se a trabalhar. O seu sistema, que por fim foi o vencedor, era capaz de registrar dados em cartões perfurados. Cada furo representava um dado que depois era classificado através de pinos de uma outra máquina, chamada “tabuladora”. É o mesmo sistema usado hoje para a correção de exames vestibulares. Se, ao contrário de Babbage, aparentemente Hollerith não se inspirou em nenhuma metáfora para construir sua máquina tabuladora, no futuro o seu nome se tornou uma metonímia. Hollerith é hoje a palavra que define em alguns lugares do Brasil o documento que recebemos como comprovante de salário.

Uma das novidades que Hollerith aplicou em sua máquina foi o uso da eletricidade. Até então todas as tentativas de se construir “calculadores” seguiam a cartilha da mecânica. Mas se a eletricidade de Hollerith nos traz direto para o século 20, foi somente no fim da década de 30 que o computador voltou a ter um novo impulso. Em 1938, o norte-americano Claude Shannon, do MIT (*Massachusetts Institute of Technology*), percebeu que existia uma analogia entre a estrutura de funcionamento dos circuitos elétricos com a álgebra booleana. Uma das intenções do inglês George Boole (1815-1864) era colocar em prática o sonho leibniziano de tratar a lógica a partir do modelo algébrico. De posse desta lógica, as grandes discussões poderiam ser resolvidas com a facilidade do cálculo matemático. Por trás deste conceito, há uma idéia: a de que os processos mentais podem ser simulados pelos circuitos do computador.

Alguns anos mais tarde, o nosso intelecto serviu novamente de inspiração para o desenvolvimento do computador. Foi o matemático John von Neumann (1903-1957) quem

deu o grande impulso que faltava para a modernização da máquina. Por outro lado, o esforço de guerra já tinha servido de estímulo para se criar muitos protótipos de computadores – alguns deles colocados em prática. No entanto, uma das constatações óbvias era que a programação das máquinas era lenta e falha, porque realizada diretamente num painel de chaves e cabos que alteravam o seu diagrama de ligações em função da tarefa a ser executada.

Von Neumann se perguntava: como nosso cérebro pode responder tão rapidamente a nossos estímulos intelectuais. Qual o limite de nossa memória e como a utilizamos? Como podemos controlar as nossas ações? E como o computador pode ser construído segundo o modelo humano de pensamento?

O matemático percebeu que assim como o computador, o nosso cérebro é repleto de programas: trocar uma lâmpada, amarrar o sapato, ou outros procedimentos de maior complexidade, como, por exemplo, realizar uma cirurgia. Às vezes, por não conseguirmos decorar alguns deles, acabamos consultando algum tipo de registro. Por exemplo, os manuais – que mostram como operar um videocassete – ou receitas – como fazer um bolo. Por termos de consultá-los, o processamento é mais lento.

Foi esta percepção que fez von Neumann entender que os procedimentos mais rapidamente executados são aqueles que já estão armazenados em nossa mente. Por analogia, para aumentar a velocidade do computador, os programas deveriam ser carregados na memória. Além disso, von Newmann percebeu que cada programa armazenado na memória pode simular outra máquina. Ou seja, cada programa seria uma máquina virtual. Se quiser fazer o computador somar carregue o programa correspondente; se quiser fazê-lo tabular, carregue outro. O computador se auto-reproduz.

De algum modo, Babbage, Shannon e von Neumann utilizaram analogias com nossos processos mentais para idealizar o funcionamento do computador. De algum modo, embora não completamente, os processadores tornaram-se uma ‘metáfora’ de nosso cérebro.

2.3 Formação de palavras na computação: algumas metáforas que viraram nomes

Não precisamos retroceder tanto assim no tempo, desvendando a história da Informática, para notar que as metáforas e o computador vêm caminhando lado a lado. As metáforas vêm sendo empregadas até hoje para dar nomes a dispositivos, comandos e elementos de interface, a ponto de nem prestarmos mais a atenção nas suas origens tropológicas.

Como mencionamos no capítulo anterior, esta forma de se aplicar a metáfora – que dá nomes a idéias ou objetos – segue o princípio da economia, segundo o qual o preenchimento de uma lacuna lexical é realizado por uma metáfora. Elas são chamadas de *metáforas denominativas*. Em casos extremos, perdemos a noção que o termo foi criado a partir de uma metáfora e tornam-se *metáforas mortas*.

A computação é um caso exemplar na utilização de metáforas de denominação. Sem tempo para dar nomes a todos os novos dispositivos, programas e ações correlatas que surgem no campo da Informática numa velocidade estonteante, as metáforas acabam por ter um papel fundamental neste sentido. Desde que o computador é computador, as metáforas têm sido utilizadas para fornecer denominações.

Um bom exemplo é a memória. Na Informática, a memória representa todo o dispositivo de armazenamento, conservação e recuperação posterior de dados, seja este dispositivo temporário ou permanente, magnético ou eletrônico, não importando a sua capacidade, nem sua velocidade de acesso. O fato de a palavra “memória” fornecer a denominação de um dispositivo tão abrangente nos ajuda a desvendar onde se encontra a similaridade entre a memória do computador e a nossa memória cerebral. Não está na sua temporalidade, nem no seu meio material; também não está na sua capacidade, nem na sua velocidade, pois em todos estes quesitos as próprias *memórias* do computador diferem entre si. Vale lembrar também que a nossa memória não é um órgão, uma parte corpórea; a memória cerebral é um conceito. Portanto a similaridade entre a memória física do computador e a nossa memória, que é não-orgânica, só pode estar relacionada a uma abstração: à nossa faculdade de reter informações, idéias, impressões, representações e conhecimento.

Por outro lado, a memória do computador, pelo menos a chamada principal²⁷ ou RAM, é dividida em células. Ora, não é possível saber se no nosso cérebro cada lembrança, imagem, palavra, cheiro, cor, enfim cada informação é exatamente guardada em células da nossa mente. De qualquer modo, no computador, é nas chamadas células de memória que são guardados as informações. E, equivocada ou não, esta compreensão é uma verdade metafórica²⁸.

Já, para recuperar um dado armazenado na memória, utilizamos do ‘endereço de memória’. É necessário que haja alguma maneira para identificar em qual célula um dado está guardado. Portanto, cada célula deve ter um endereço. Até que se prove o contrário, células humanas não têm endereços. Mas as do computador têm. E de qualquer modo, a compreensão de que endereço é a referência que identifica cada célula de memória também é uma verdade metafórica. Cada uma tem um endereço que é único, assim como temos o nosso próprio endereço que identifica onde moramos. Não fosse assim, um carteiro teria sérias dificuldades em saber como entregar nossas correspondências. E o processador sentiria o mesmo martírio para encontrar um dado na memória.

Outro dispositivo cujo nome foi criado a partir de uma metáfora é o *mouse*. Neste caso a metáfora foi definida pela analogia física do objeto com o animal. O *mouse* possui um formato bastante similar ao corpo de um ratinho e a associação se completa porque o cabo do dispositivo se parece com a cauda do animal. Provavelmente essa associação foi marcada por influência de desenhos animados e *cartoons*. É muito comum encontrar personagens de História em Quadrinhos e animações com a representação de camundongos simpáticos. Fosse a associação feita com o animal que conhecemos na sua realidade,

²⁷ “Uma das principais características definidas no projeto de arquitetura de sistema de Von Neumann, o qual se constitui na primeira geração de computadores, consistia no fato de ser uma máquina ‘de programa armazenado’. O fato de as instruções, uma após a outra, poderem ser imediatamente acessadas pela UCP é que garante o automatismo do sistema e aumenta a velocidade de execução dos programas.” [Monteiro, 1996:93] A memória que armazena o programa que é executado pelo processador é a chamada memória principal. Popular e equivocadamente é chamada de memória RAM – *Random Access Memory* ou Memória de Acesso Aleatório (pois não é só a memória principal que é RAM).

²⁸ Donald Davidson assim define ‘verdade metafórica’: pelo simples fato de ‘compreendemos uma metáfora, podemos chamar o que compreendemos de ‘verdade metafórica’ e (até certo ponto) dizer qual é a ‘verdade metafórica’. No entanto, mais tarde, no texto, o autor se mostra cético quanto a tentar estabelecer significados para a metáfora. Para o autor, “... as idéias não explicam a metáfora, a metáfora as explica”.

provavelmente a metáfora não se completaria tão facilmente. Principalmente porque outros roedores mais “inofensivos” – como esquilos, castores, *hamisters* – possuem o mesmo formato de corpo. De qualquer modo, assim como a memória, o *mouse* é uma metáfora de denominação.

[O disco rígido lembra o disco vinil]

Outro caso é o do disco rígido. Seu nome também foi criado a partir de uma metáfora: em função da semelhança física entre o dispositivo e o formato de um disco. O nome, composto, se completa com a palavra rígido, para diferenciar de outros dispositivos afins, como o disquete, que são flexíveis. Esta denominação – disco rígido – se tornou a mais usual, mas este dispositivo tem ainda algumas variações para o seu nome, como seu correspondente em inglês,

hard disk, ou sua abreviação, HD. Além disso, a peça ainda é conhecida como *winchester*. E isto é muito curioso, porque *winchester* é, na realidade, o nome de um famoso rifle. A relação é estabelecida porque o primeiro disco rígido da IBM armazenava 30 megabytes de informação de cada lado. Como o *drive* era 30-30, as pessoas chamavam-no de disco *winchester*..

Na prática, os dados são guardados nos dispositivos de armazenamento em forma de “arquivos”. Tecnicamente falando, um arquivo de computador é um registro de um certo número de bits que em conjunto formam uma informação significativa. A palavra arquivo foi formada a partir da metáfora de um arquivo convencional, em geral de ferro, com gavetas, que encontramos normalmente em escritórios. A semelhança subentendida entre o arquivo de um computador e um arquivo convencional está na estrutura organizacional de armazenamento de suas informações: através de diretórios, no caso do arquivo de computador; e de gavetas e pastas, no arquivo convencional. Do mesmo modo ambos são

referenciados e identificados por nomes (no caso dos arquivos convencionais, os nomes são apresentados nas etiquetas).

Memória, *mouse*, disco, arquivo são, assim, casos de metáforas de denominação. Antes de apresentarmos outros casos semelhantes, vamos fazer um pequeno parêntesis para descrever dois termos fundamentais no universo computacional: o de *hardware* e o de *software*. Estas palavras não foram criadas a partir de metáforas de denominação, mas a metáfora participou de outro modo na criação de seus nomes.

2.4 Hardware e software

O computador é usualmente dividido em duas partes: *hardware* e *software*²⁹. *Hardware* é identificado como a parte física do computador. *Software* é o termo geral para designar os programas de computador, ou seja, o conjunto de instruções arranjadas logicamente para serem executadas pela máquina.

Na realidade, a palavra *hardware* foi formada pela justaposição de dois termos do inglês: *hard*, que significa duro, resistente, firme, rígido, compacto + *ware*, usado geralmente no plural como significado de mercadorias, produtos. *Software*, por outro lado, é composição de *soft*, que em inglês exprime macio, mole, suave, brando, ameno + *ware*. Portanto *hardware* é mercadoria rígida e *software* é a mercadoria suave. Trocando em miúdos, *hardware* são os dispositivos ou as peças do computador, como, por exemplo, o disco rígido, as placas de expansão, ou os periféricos, como o teclado, o *mouse* e o monitor. *Software* é a porção suave, volátil do computador representada pelos seus programas.

Em geral supõe-se que cada parte desta divisão não pode funcionar satisfatoriamente sem a outra. Ou seja, um computador que só tenha seus componentes físicos não tem utilidade

²⁹ Atualmente alguns autores têm alertado para a importância de se incluir uma terceira parte nesta classificação: a de *peopleware*. *Peopleware* é a parte do computador relacionada às pessoas que usam o computador. O principal argumento para a sua inclusão nesta classificação é que da mesma forma que o computador não pode funcionar somente com a existência do *hardware*, nem só com a existência do *software*, tampouco pode funcionar sem o componente humano. Por exemplo, um computador não pode ser ligado se não houver a participação do seu usuário.

prática se não tiver *software*. Por outro lado, um programa não pode ser executado sem *hardware*. Mas isto pode não passar de mera ilusão.

Como muitas das classificações existentes, a divisão do computador em *hardware* e *software* pode parecer demasiadamente rígida. Há um limite segundo o qual não podemos dizer com precisão se uma determinada entidade do computador é uma coisa ou outra. A prova disto é apontada por Tanenbaum.

Segundo o autor, existe um nível no computador, chamado de lógica digital, na qual pequenos componentes chamados portas são construídos, a partir de transistores, e podem ser modelados com precisão como dispositivos digitais. “Cada porta possui uma ou mais entradas digitais (sinais representando 0 ou 1) e fornece como saída funções simples destas entradas, tais como *as funções lógicas* (grifo meu) AND e OR.” [Tanenbaum, 1992:3]. Estas portas podem ser combinadas em forma de circuitos combinatórios, que são fabricados e vendidos individualmente em unidades chamadas circuitos integrados, pastilhas ou *chips*. Aqui, neste ponto, podemos questionar a propagada divisão do computador entre *hardware* e *software*.

Circuitos são dispositivos, portanto *hardware*, mas a sua combinação (ou o seu desenho) representa uma pequena instrução de máquina, que não deixa de ser um pequeno programa. Este programa, ao contrário do que ocorre normalmente, não é armazenado. Ele está embutido na própria pastilha. Ou seja, não é o caso de dizer que o *software* precisa do *hardware* para funcionar, ou vice-versa. Neste caso, eles não se separam.

Embora, este pequeno exemplo, possa parecer um caso isolado, isto é mais comum num sistema computacional do que se pode imaginar. Circuitos deste tipo se encontram no computador aos milhares.

Na verdade, o que conhecemos em geral como *software* é representado pelos programas que são armazenados na memória principal e que muitas vezes são construídos através de uma linguagem que o próprio processador não compreende. Para que este possa entender, é necessário que seja traduzido ou interpretado para a linguagem que ele de fato entende – e

que é conhecida como linguagem de máquina. Tanenbaum, portanto, vai ainda mais longe quando diz que cada linguagem de computador (excetuando a linguagem de máquina, que é a linguagem do computador real) representa uma máquina virtual. “Cada máquina tem a sua linguagem de máquina, que consiste em todas as instruções que a máquina pode executar. De fato, uma máquina define uma linguagem. Do mesmo modo, uma linguagem define uma máquina.” [idem, 3]

Talvez a divisão tão explícita do computador em *hardware* e *software* seja devido à implementação definitiva da chamada memória temporária nos computadores modernos. Antigamente, os programas estavam embutidos em componentes físicos, como cartões perfurados ou mesmo discos magnéticos. Hoje eles são armazenados, carregados na memória temporária do computador, e talvez este fato tenha servido como inspiração para a criação da palavra *software*. Por ser carregado e por não conseguirmos vê-lo a olho nu, o programa parece ser volátil, etéreo, suave, leve, ou seja *soft*. Em contraposição temos a parte rígida do computador, representada por seus componentes – o *hardware*.

Soft e *hard* são metáforas para a sensação que sentimos em relação aos componentes do computador. *Software*, ou mercadoria suave, é como sentimos a existência dos programas. *Hardware*, ou mercadoria rígida, é como enxergamos a parte física do computador.

2.5 Outras partes do *hardware* e do *software*

Na formação de palavras computacionais, tanto em *hardware*, quanto em *software*, surgem outros exemplos do uso da metáfora, em especial as chamadas metáforas de denominação.

Um exemplo são as portas de comunicação. Portas são pequenas peças de *hardware*, que podem ser um ponto de acesso a um computador, a uma rede ou a qualquer dispositivo elétrico. Podem ainda ser definidas como uma interface entre um processo e um sistema de comunicação. Olhando por outra perspectiva, na verdade, as portas dão acesso a novos dados. As informações trafegam de uma máquina ou dispositivo para outro através de cabos, que são conectados por estas portas. Portanto a metáfora é evidente: os dados “entram” e “saem” e os limites são dados pelas portas.

Outro exemplo do uso da metáfora para criar nomes de peças de *hardware* é o da placa-mãe. O dispositivo é o circuito impresso principal do computador e contém soquetes para receber placas adicionais. Serve, portanto, de suporte para outras placas. Por extensão, a placa-mãe é chamada também de placa principal. No entanto, por trás do nome pode estar inscrita também uma outra metáfora: a de geradora. Ao receber outras placas, a placa-mãe faz nascer novas possibilidades na máquina. Com uma placa de som, o computador pode transformar ondas sonoras em dados digitais e vice-versa, para que possamos escutar seqüências de som; com uma placa de rede, o computador pode trocar dados com outros computadores, além de poder dividir dispositivos, como a impressora; com uma placa de captura de vídeo, o computador pode digitalizar um filme para que possamos vê-lo diretamente do monitor.

O que também recebeu sua denominação a partir de uma metáfora foi o termo arquitetura. Todo sistema computacional tem sua arquitetura. No caso, é uma metáfora para a estrutura que define o padrão para todos os dispositivos conectados e todos os *softwares* que rodam no computador. É também uma metáfora para o “projeto” da máquina. Tanenbaum descreve a arquitetura como o conjunto formado pelas instruções de máquina, pela organização da memória, pela estrutura de entrada e saída de dados e pela estrutura de barramento. [Tanenbaum, 1995:5]

Além de fornecer nomes para peças de *hardware*, a metáfora tem servido como fonte de inspiração para a criação da terminologia do *software*. Por exemplo, *driver*.

Driver é um pequeno programa de computador que controla dispositivos de *hardware*. O seu significado em inglês, condutor ou motorista, não poderia portanto ser mais sugestivo: o *hardware* está para o automóvel, assim como o *driver* (o programa) está para o motorista. Uma peça não funciona sem o seu *driver*, assim como um carro não anda sozinho sem o seu condutor. Da mesma forma, um motorista não existiria, se não houvesse carros. E um *driver* não seria desenvolvido se não existisse a peça que ele vai controlar. Em português, *driver* foi traduzido para controlador, que não deixa de ser um condutor: o motorista controla o carro.

Outra denominação criada a partir de uma metáfora é o chamado *plug-in*. *Plug-in* quer dizer algo como fazer uma ligação elétrica a partir de uma tomada. O que os *plug-ins* estariam ligando em um sistema computacional? Ora, eles estariam ligando outros programas. Em geral, os aplicativos vêm com um pacote fechado de ferramentas. No entanto, hoje, a maioria deles permitem que você acrescente ou conecte ferramentas adicionais. Para disponibilizá-las, basta “ligar” estes pequenos programas na “tomada” do *software*. Ou em outras palavras, basta instalar os *plug-ins*.

2.6 Diretórios, Árvores, galhos, raiz e ações associadas

Cada dispositivo (unidade) de armazenamento permanente de um computador (disco rígido, disquete, CD-ROM) recebe do sistema operacional uma identificação que é determinada por uma letra. Em geral, as letras “A” e “B” são reservadas para identificarem as unidades de disquete. A letra “C” identifica a primeira unidade de disco rígido. A “D” pode ser outro disco, ou uma unidade de CD-ROM. A partir destas unidades, o sistema operacional do computador começa a organizar os diretórios.

O diretório é um arquivo especial que possui algumas informações de caráter gerencial, mas, na prática, ele é enxergado como um repositório, que agrupa um conjunto de arquivos. A palavra diretório vem do inglês *directory*³⁰. Outra tradução possível para o português poderia ser diretoria. De qualquer maneira, foi a palavra diretório que se popularizou. No seu significado mais convencional, do nosso dia-a-dia, tanto diretório quanto diretoria identificam o local em que se encontra um diretor. A aplicação desta metáfora tem uma explicação. Cada diretório de um disco rígido controla a entrada e saída de arquivos presentes dentro dele. Tecnicamente falando, para encontrar um arquivo, o

³⁰ É sempre bom lembrar que muitas palavras da terminologia informática são traduções do inglês. Em muitos casos a tradução é adequada, para efeitos de uma análise que leva em conta uma interpretação da criação de termos a partir de metáforas (ou de outro tropo similar, como a metonímia). Por exemplo, no inglês, *memory* é identificada como uma metáfora para um tipo de dispositivo de armazenamento de dados do computador do mesmo modo que acontece com a sua tradução para o português, “memória”. Em outros casos, como no de “diretório”, há mais de uma tradução possível.

sistema operacional procura na entrada do diretório o número do *cluster*³¹ deste arquivo. Portanto a idéia que está por trás da metáfora “diretório” é a de controle.

Uma segunda interpretação para a palavra diretório pode ser obtida da própria estrutura de diretórios de um disco. Embaixo de cada diretório pode haver um ou mais arquivos e um ou mais subdiretórios, ou, como poderíamos chamar, subdiretórias. Abaixo destes subdiretórios podem existir arquivos ou ainda outros subdiretórios. E assim por diante. Percebe-se então que a organização estrutural dos diretórios de um disco possui uma analogia com o desenho de uma árvore, que possui uma raiz e galhos. Dos galhos podem sair outros galhos, de onde podem sair outros galhos etc. Não é à toa, portanto, que a estrutura de diretórios de um disco é chamada justamente de árvore; e a base desta estrutura, de onde são criados hierarquicamente os primeiros diretórios, é denominada raiz. Também não é coincidência que, no sistema operacional MS-DOS, o comando executado para visualizar a estrutura de diretórios de uma determinada unidade é *tree*.

No entanto esta estrutura de diretórios pode ainda esconder uma segunda leitura: a de um organograma. O organograma é representado por quadros compostos de tal maneira que possamos enxergar a estrutura organizacional e hierárquica de uma organização. Existe a instância superior representada pelo quadro de maior hierarquia. Abaixo dele começam a surgir novos quadros e assim por diante. O quadro superior corresponderia à raiz, se comparássemos com a estrutura da árvore. Só que em vez de começarmos a subir com os galhos, começaríamos a descer com quadros.

Esta outra forma de visualizar a estrutura dos discos de armazenamento do computador permite interpretar a metáfora a partir da outra versão que traduz a palavra *directory*. Poderíamos enxergar cada quadro hierárquico de um organograma como uma “diretoria”. *Directory* seria a metáfora para a diretoria de cada quadro de uma organização hierárquica.

³¹ *Cluster* é a menor unidade de alocação de espaço reservado para colocar um arquivo pelo sistema operacional. Um arquivo pode ter que usar mais do que um *cluster*, mas ele não poderia utilizar, por exemplo, meio *cluster*. *Cluster* em inglês quer dizer agrupamento. Neste caso, agrupamento de bits. Este termo não foi traduzido para o português.

Como havíamos afirmado no capítulo anterior, uma metáfora, além de ser descrita pela analogia que possui com um termo (palavra, ícone etc), pode também ser descrita pela similaridade que tem com a estrutura em que se encontra este termo; é o caso das metáforas para árvore e diretório.

Outros termos também foram criados a partir da metáfora da estrutura de diretórios. Listar, por exemplo, é o comando que mostra os arquivos e subdiretórios que se encontram abaixo de um outro diretório (Por exemplo, no sistema operacional Unix, o comando para listar é *ls* – do inglês *list*). Embora

[Uma listagem típica do DOS]

```

Prompt do MS-DOS

C:\WINDOWS\SYSTEM>dir *.exe

0 volume da unidade C não tem nome
0 número de série do volume é 1816-1BD5
Diretório de C:\WINDOWS\SYSTEM

DAGIECFG EXE      48.640  15/02/99   0:00 DAGIECFG.EXE
RAPLAYER EXE     75.264  24/08/96  11:11 RAPLAYER.EXE
SAGE EXE         3.856  19/05/00   1:50 sage.exe
TAPIEXE EXE       1.784  24/08/96  11:11 TAPIEXE.EXE
TAPIINI EXE       7.632  24/08/96  11:11 TAPIINI.EXE
5 arquivo(s)      137.176 bytes
0 diretório(s)    467.664.896 bytes livres

C:\WINDOWS\SYSTEM>

```

uma lista seja uma relação ou um rol que mostra os elementos componentes de um conjunto, e quando falamos “listar”, imaginamos esta relação vista de um modo linear, os arquivos de um disco não são armazenados ou organizados desta maneira. Eles são guardados aleatoriamente. A política de armazenamento leva em conta principalmente os setores que estão livres e nem sempre eles são sequenciais. Mas quando enxergamos o rol de arquivos de cada diretório, temos a impressão que os arquivos também estão armazenados como na listagem mostrada.

Esta mesma ilusão temos quando movemos um arquivo de um lugar para outro. Na verdade, os bytes deste arquivo não são movidos, de fato, de um setor para outro do disco³². Ele continua no mesmo lugar. O que acontece é que a entrada do arquivo passa a estar vinculada a um outro diretório. Temos a ilusão que o arquivo se moveu, por isso a metáfora utilizada para explicar esta operação é a de movimentação. O comando é chamado *move*. Mas o arquivo só se move na estrutura que enxergamos metaforicamente, não na estrutura real.

³² Há ocasiões em que de fato os bytes de um arquivo são movidos. Isto acontece quando pedimos para mover um arquivo de um dispositivo de armazenamento para outro (de um disco para outro ou de um disco para um disquete, por exemplo). Mas isto não acontece quando movemos um arquivo de um ponto para outro do mesmo dispositivo de armazenamento.

Outro exemplo é o comando *delete*, apagar, que, da mesma forma que *move*, não explica o que de fato acontece na estrutura de diretórios de um disco. Na realidade, o comando *delete* não apaga um arquivo. O espaço que ele ocupa na memória é deixado à disposição para que possa ser utilizado por outros dados. Dependendo do caso, os bytes “apagados” podem até mesmo ser recuperados. Mas o usuário leigo, em geral, acha que está de fato apagando um arquivo e isto provoca a idéia equivocada de que não pode recuperá-lo.

Portanto se uma metáfora pode ser benéfica por facilitar a compreensão de certos termos e estruturas, como no caso dos diretórios, por outro lado, ela pode provocar certas ilusões. É o caso dos comandos *list*, *move* e *delete*. As metáforas listar, mover e apagar nos ajudam, de certo modo, a compreender melhor uma ação. Mas, dependendo do caso, esta ação pode estar relacionada somente à sua estrutura metafórica, mas não à sua estrutura real. Nem todas as ações dentro de uma estrutura metafórica têm semelhança com a de sua correspondente estrutura real. E, sendo assim, às vezes uma metáfora utilizada para descrever uma ação, como nos três casos citados, pode gerar confusões conceituais.

Talvez o grande problema tenha sido utilizar como metáforas termos que já tenham um significado também dentro da estrutura real de diretórios. Quando dizemos “apagar” um arquivo, a primeira coisa que vem à nossa mente é que de fato estamos eliminando-o da estrutura de diretórios de um disco, pois tecnicamente isto é possível. Não passa pela nossa cabeça que ele poderia ser recuperado – como acontece de fato. Portanto apagar um arquivo não é uma boa metáfora. Vamos ver agora um caso – o do comando salvar – que não gera confusões porque esta palavra não tem significado algum dentro da estrutura real de um sistema computacional. É evidentemente uma metáfora.

2.7 Salvar, armazenar, carregar e importar dados

Como já dissemos acima, a memória é todo dispositivo de armazenamento e recuperação de dados de um sistema computacional. Mas, num sistema computacional, quem executa (ou processa) os programas é a UCP (Unidade Central de Processamento) ou processador³³.

³³ A UCP (Unidade Central de Processamento), popularmente conhecida como processador, é o componente de um sistema computacional responsável pela execução dos programas. Sua função consiste em: a) buscar

Toda instrução³⁴ executada pelo processador deve ser antes buscada e, depois de processada, guardada em uma das memórias do computador³⁵. Por motivos tecnológicos, alguns destes dispositivos de armazenamento não são suficientemente velozes³⁶; é o exemplo do disquete, do disco rígido ou do CD-ROM³⁷. Como o processador executa suas instruções em velocidades altíssimas (em nível de nanossegundos), é necessário então que haja um dispositivo que além de guardar os dados, tenha uma velocidade de acesso bem mais alta. A memória principal do computador cumpre este papel. No entanto, embora este dispositivo resolva a questão da velocidade, ele tem um pequeno inconveniente: é uma memória temporária. Ou seja, quando desligamos o computador, os dados que estavam lá armazenados são perdidos e devem então ser deslocados para algum outro dispositivo de memória que os “guarde” permanentemente – como o próprio disquete ou o disco rígido. Portanto, guardar uma informação é registrá-la na memória. E, na verdade, um dado só é permanentemente guardado até que outros dados ocupem este mesmo espaço de memória utilizado por ele; ação solicitada pelo usuário, evidentemente.

uma instrução na memória (operação de leitura), uma de cada vez; b) interpretar que operação a instrução está explicitando; b) buscar os dados onde estiverem armazenados, para trazê-los até a UCP; d) executar efetivamente a operação com o(s) dado(s), guardar o resultado (se houver algum) no local definido na instrução; e finalmente, e) reiniciar o processo apanhando nova instrução. [Monteiro, 1996:134]

³⁴ Uma instrução de máquina é a especificação de uma operação básica (ou primitiva) que o *hardware* é capaz de realizar diretamente. [Monteiro, 1996:147]. Por exemplo, SOMAR é uma instrução. O conjunto de instruções ordenados de forma lógica forma um programa.

³⁵ Para que a execução de um programa pelo processador tenha início, é necessário que antes ele esteja carregado em algum dispositivo de armazenamento, conhecido como memória. Existem diversos tipos de memória, que se distinguem em função de sua velocidade de acesso, capacidade de armazenamento, tecnologia empregada, temporalidade, custo ou volatilidade. A velocidade de acesso se caracteriza como o tempo que a memória leva para colocar uma informação que está armazenada nela até o processador. Quanto maior a velocidade de uma memória, melhor para o sistema como um todo. Isto porque o processador é um componente que devido às suas características tecnológicas trabalha a uma velocidade altíssima. Se o tempo que uma memória leva para deslocar um dado para a UCP for muito alto, o processador tem que “esperar”, e todo o sistema fica comprometido com esta “demora”. Memórias como disco rígido, disquete e CD-ROM são lentos neste sentido. A memória principal do computador (a chamada memória RAM) tem velocidade de acesso maior, portanto é mais adequada para trabalhar diretamente com o processador.

³⁶ Quanto se diz que estes dispositivos são lentos, na verdade, quer-se dizer que o seu tempo de acesso é lento. O tempo de acesso indica “o período de tempo decorrido desde o instante em que foi iniciada a operação de acesso (quando a origem – em geral é a UCP – passa o endereço de acesso para o sistema de memória) até que a informação requerida (instrução ou dado) tenha sido efetivamente transferida. É um dos parâmetros que pode medir o desempenho da memória.” [Monteiro 1996:89]

³⁷ O disquete, o disco rígido e o CD-ROM foram categorizados, em conjunto, como não velozes, mas na verdade eles também têm velocidades de acesso muito diferentes. O tempo de acesso de um disquete é muito mais lento do que o de um disco rígido. Ainda assim, ambos, como o CD-ROM, são muito lentos para trabalhar com a UCP.

A ação de guardar um dado em memórias permanentes tem um sinônimo na computação, na verdade mais popular, provavelmente por evidenciar uma ação muito mais forte do que simplesmente o de guardar: é o de “salvar” um dado. Provavelmente o termo salvar é o resultado de uma certa ingenuidade dos primeiros usuários de computador, que trabalhavam horas a fio, sem se importar em guardar os dados em dispositivos permanentes. Qualquer energia elétrica interrompida fazia o computador ser desligado, o que poderia significar a perda de um dia de trabalho. Olhando por esta perspectiva, mais do que guardar um dado ou um arquivo, estes deveriam ser salvos – a imagem mental que fazemos deste ato é tão forte que talvez explique o sucesso desta metáfora.

Já, armazenar um dado é mantê-lo em um circuito de computador que tenha a capacidade de segurar o sinal elétrico por um período de tempo superior à simples passagem da corrente. Em geral, a corrente elétrica literalmente entra e sai do circuito e sua carga mantém-se por um tempo infinitesimal. Para que o sinal binário se mantenha no circuito há tradicionalmente duas tecnologias: ou coloca-se um capacitor neste circuito, cujo objetivo é recarregar o sinal de tempos em tempos para que ele não seja perdido, tipicamente chamados circuitos de memória DRAM (*dynamic RAM*); ou desenhar um circuito que não deixe que a carga saia, como se fosse um cachorro mordendo o próprio rabo; são os chamados circuitos de memórias SRAM (*static RAM*). Mas a palavra armazenar significa guardar algo em um armazém. O armazém do computador, como já visualizava Babbage há muito tempo atrás, é a memória. Na Informática, portanto, armazenar é colocar um dado na memória para que ele posteriormente possa ser extraído e quem sabe processado.

Por outro lado, “carregar” é, na Informática, o ato de transferir um dado de um ponto a outro do sistema computacional. Mas geralmente é associado ao caminho que vai de um local qualquer à memória volátil (a principal ou RAM) do computador. A metáfora carregar pode ser atribuída a duas imagens diferentes, porém complementares. A palavra carregar é associada à idéia de levar, transportar. Por outro lado, também significa encher, pesar sobre, ou, como diz sua variação imediata, sobrecarregar. Na informática, portanto, carregar tem estes dois sentidos: levar um dado para a memória e também sobrecarregá-la com mais um dado nela. Não é à toa que, atualmente, o verbo “carregar” seja tão utilizado para descrever

o processo de transferência para a memória dos pesados softwares que existem hoje. Dizemos normalmente “carregar um programa na memória”.

Outro termo relacionado à manipulação de dados é o de importar. No dia-a-dia esta ação está relacionada à idéia de fazer vir ou trazer algo de outro lugar ou país. Na informática, “quem” vêm são os dados e o “lugar” para onde eles vão é um ponto da memória que foi solicitado por um *software*. Ou seja, quando importamos uma informação ou um arquivo, estamos trazendo para um programa dados que foram gerados por outro.

Dependendo do caso, estes dados sofrem um processo de transformação. Neste caso, mais do que simplesmente estar importando um produto final, estamos importando matéria-prima para ser processada. O dado, assim, passa por um processo de ressignificação, pois ele é importado, processado e transformado.

2.8 Redes

Na Informática, uma rede é a interconexão de diversos computadores. As máquinas podem se comunicar, recebendo ou transmitindo dados de um para outro, ou compartilhando periféricos³⁸. A metáfora foi criada a partir da idéia de que a estrutura de uma rede é semelhante à estrutura dos computadores interconectados, em que cada terminal representaria um nó. Esta metáfora pressupõe, no entanto, a idéia de que na estrutura

não há uma hierarquia entre os nós, porque é baseada no desenho da rede. Ou seja, nenhum nó tem um *status* especial em relação a outro. Colocado desta maneira, numa rede de

³⁸ Periféricos são os dispositivos de entrada e saída do computador, ou seja, dispositivos que enviam ou recebem dados. A denominação “periférico” é uma metáfora que indica que estes dispositivos estão periféricamente localizados em relação ao sistema central de um computador, representado pelo processador e pela memória. Exemplos de periféricos: teclado, monitor e mouse. Compartilhar periféricos significa literalmente partilhar o seu uso com mais de um sistema computacional. Uma impressora, por exemplo, pode ser utilizada por mais de um computador.

computadores, cada máquina deveria ter o mesmo *status* hierárquico do que as outras. Mas nem sempre é assim.

Há casos em que um computador, chamado servidor, em geral com maior capacidade de processamento e de armazenamento, abastece os outros computadores com dados e programas. Esta rede é convenientemente chamada de cliente/servidor. Ou seja, o computador que abastece ou “serve” os outros é o servidor. Os terminais, que recebem, e que são abastecidos com dados, são os clientes. A metáfora neste caso não provém da idéia feudal de servidão, pois afinal de contas um servo não tem clientes e sim senhores. A metáfora é proveniente de uma relação comercial, em que, por exemplo, num restaurante, o garçom, ou servidor, abastece o cliente com sua mercadoria – a refeição. Mas, no caso de uma rede de computadores a mercadoria são os dados.

As redes de computadores têm algumas topologias diferentes. As mais comuns são a topologia em anel, em estrela e de barramento. O nome das topologias vêm da analogia que a estrutura da rede possui em relação ao um determinado desenho. Assim, na rede em estrela, os computadores são direcionados a uma estação central que possui chaveadores que gerenciam a conexão entre os nós. Na configuração em anel, todos os computadores estão conectados a um mesmo circuito em forma de anel. Uma mensagem circula continuamente por ele, buscando dados em cada nó por que passa. Na topologia de barramento, todos os computadores se ligam a uma linha comum (um barramento). Quando uma mensagem é enviada ao barramento, ela passa por todos os nós, até encontrar o endereço correto.

2.9 User-friendly

Como pudemos constatar com os exemplos apontados acima a metáfora foi personagem fundamental como facilitadora no processo comunicativo na emergência do universo computacional. Evidentemente, é muito mais simples falar *mouse* do que dispositivo de apontamento. Historicamente, então, as metáforas desempenharam um importante papel de tradutora ou mediadora: de um mundo hostil representado pelos dispositivos e pela

terminologia técnica do computador para um ambiente mais próximo do universo comunicativo do homem.

Ainda assim, até pouco tempo atrás, menos de dez anos, os usuários de computadores tinham que decorar uma série de comandos para operar a máquina. Embora as metáforas fossem utilizadas como definidoras destes comandos, o usuário ainda não se sentia à vontade. Este tipo de procedimento criou, inclusive, o mito – que vem perdendo força – de que o computador é um aparelho difícil de operar, se compararmos a uma geladeira ou a uma televisão. Naquele tempo, ele era considerado uma máquina para poucos, para iniciados, técnicos, cientistas, ou pessoas que tinham paciência ou motivo suficiente para manipulá-lo. Para muitos, o computador estava mais para um objeto de ficção científica do que para um eletrodoméstico.

Com o tempo, e não demorou muito, o computador avançou tecnologicamente o suficiente para tornar injustificável a necessidade de usuários terem que decorar uma lista enfadonha de comandos. O monitor já tinha cores e resolução suficientes para se criar um ambiente gráfico. Percebendo esta nova realidade, a Apple, juntamente com a Xerox³⁹, foram pioneiras em criar o que se convencionou chamar *user friendly interface* ou *graphic user interface* (GUI)⁴⁰, duas maneiras para definir o ambiente gráfico amigável apresentado pelos sistemas atuais e que usamos diariamente.

Grande parte deste esforço pode ser credenciado a Douglas Engelbart, do laboratório PARC/Xerox, que definiu as interfaces de sistemas operacionais como a metáfora de um

³⁹ Embora a Apple seja comumente conhecida como a precursora da chamada GUI (*Graphical User Interface*), acredita-se que o avô desta proposta tenha sido Ivan Sutherland do MIT (*Massachusetts Institute of Technology*). Em 1962, ele desenvolveu um programa chamado Sketchpad que desenhava linhas, círculos e pontos. Também foi o primeiro software baseado em janelas e o primeiro a fornecer características gráficas e a atribuir relações entre objetos. Mas o programa não passou de pesquisa: não foi comercializado. Em 1970, a Xerox desenvolveu um sistema interno para os seus computadores chamado Alto. Já tinha janelas, menus e usava o mouse. Por volta de 1976, ícones foram acrescentados na área de trabalho. O primeiro software comercial desenhado com as prerrogativas da GUI foi o Xerox Star em 1981. [Mandel, 1997:157-158]

⁴⁰ “A característica básica de uma interface baseada no GUI é a integração de um número de elementos que trazem as tarefas e o trabalho que nós fazemos no computador para a vida. Em termos simples, a GUI é a representação gráfica de, uma interação com, programas, dados, e objetos na tela do computador. Mas ela vai além. A GUI tenta fornecer aos usuários ferramentas e aplicações que fazem de uma tarefa algo mais fácil do que uma lista de comandos.” [Mandel, 1997:160]

tampo de escrivaninha (o *desktop*); e a Ted Nelson, que, além de ser o criador do termo hipertexto, já em 1980 percebeu um estímulo positivo no usuário quando uma interface é construída pelo que ele chama de princípio da virtualidade – uma representação da realidade que pode ser manipulada.

Os engenheiros da Xerox PARC (os criadores da metáfora do *desktop*) eram sensações em pesquisa e tecnologia, mas não no ramo comercial. O desenvolvimento do primeiro sucesso comercial de um sistema baseado na GUI foi obtido pela Apple. Steve Jobs visitou a Xerox e trouxe sua equipe de *designers*. Em 1983 desenvolveu o Apple Lisa e em 1984 criou o Macintosh. [Mandel, 1997:159]

Neste processo, os comandos deram lugar a uma interface gráfica. Botões, ícones, menus tornaram-se metáforas para os próprios comandos. Na realidade por trás de cada botão existe uma série de instruções. Ao clicar em cima de um menu ou ao passar o mouse por cima de um ícone estamos solicitando que se execute um determinado programa. Já vimos no capítulo anterior que cada item de menu, cada botão, cada tecla de função dispara um pedaço de programa que está contido dentro de um programa maior.

Thomas D. Erickson, um dos mais renomados pesquisadores da Apple, que conheceu de perto as profundas alterações que emergiram do aparecimento da interface gráfica na Apple, através da metáfora do *desktop*, descreve assim este processo:

*“Just as we use military terms to make the rather abstract process of arguing more tangible, so we use object and container terms to make the Macintosh file system more concrete. And just as we experience arguments as real conflicts, most Macintosh users believe that when they move a document icon from one folder to another, they are really moving the document itself (what is ‘really’ happening is that a pointer to the file is being moved – of course, is a metaphor too...)”*⁴¹ [Laurel, 1992:66]

⁴¹ T.A.: “Assim como usamos termos militares para tornar o processo abstrato de argumentação mais tangível, nós usamos objetos e termos para fazer o sistema de arquivos do Macintosh mais concreto. E assim como

Na verdade a grande contribuição da Apple e da Xerox/PARC foi a de colocar comunicadores no processo de desenvolvimento de interfaces, papel que era desempenhado por programadores. A especialidade dos programadores é “a comunicação com computadores, uma atividade substancialmente diferente da comunicação com seres humanos” [Heckel, 1993:1]. Como Heckel afirma, os filmes só prosperaram quando os técnicos perderam o controle para os artistas – o mesmo aconteceu com os computadores pessoais.

“A principal novidade que Griffith trouxe para o cinema foi uma mudança de ênfase: daquilo que era apresentado para como era apresentado – de um trabalho técnico para um artístico. (...) Para tornar nossos produtos úteis para os usuários, temos de identificar e desenvolver os equivalentes, ao nível do *software*, das técnicas de cinema – *close-up*, plano de movimento, edição e assim por diante. Já temos algumas dessas técnicas – os menus e o controle de cursor, por exemplo – mas só isso não é suficiente.” [Heckel, 1993:7-8]

Esta mudança provocou resultados bastante positivos. Por exemplo, a metáfora do *desktop* jogou fora a antiga concepção de apagar um arquivo (ou documento) para algo mais próximo do que ocorre na realidade. Os novos comunicadores apresentaram uma nova metáfora para esta ação. Heckel descreve assim:

“Primeiro, o comando *delete* não remove realmente um arquivo, que é a expectativa do usuário (...) Uma metáfora útil – a de lata de lixo – melhorará a situação. O comando *wastebasket* (lata de lixo) sugere jogar algo fora, mas também deixa aberta a possibilidade de que as coisas possam ainda estar disponíveis, e comandos como *search wastebasket* e *empty wastebasket* podem se encaixar naturalmente no modelo do usuário.” [idem, 36]

Assim, em vez de decorar comandos, o usuário pressiona botões. E isto é muito mais natural, porque tem uma analogia com operações usuais e intuitivas do nosso dia-a-dia.

A compreensão de uma interface gráfica é facilitada porque a maioria dos seus elementos também são metáforas ou representações: os botões virtuais da interface implementam a ação física de um clique no botão do mouse; os ícones, com suas diversas reproduções, tornam-se auto-explicativos; os menus são listagens de opções, como os menus de um

usamos argumentos como conflitos reais, a maioria dos usuários de Macintosh acreditam que quando movem o ícone de um documento de uma pasta a outra, eles estão de fato movendo o documento em si... o que está acontecendo ‘realmente’ é que é a entrada do arquivo que está sendo movida – certamente, isto é uma metáfora também...”

restaurante; os campos de texto são elementos herdados dos formulários de preenchimento; os botões do tipo *radio* simulam chaveamentos que apresentam a dualidade liga-desliga etc.

Inegavelmente, as interfaces gráficas, ao se tornarem *user friendly*, colocaram de vez o computador na lista de eletrodomésticos básicos do dia-a-dia. Ele tornou-se “amigável” porque inteligentemente soube se aproveitar de metáforas para se fazer compreender. Ted Nelson afirmou que o trabalho do *designer* digital deve concentrar-se 10% na concepção visual ou apresentação, 30% no projeto de interatividade e 60% na determinação de uma metáfora. “Transferindo a experiência do mundo real ao seu redor para o mundo dos computadores, os usuários contam com modelos para guiar a sua interação com computadores. É aí que entra o conceito de metáfora.” [Mandel, 1997:25]

2.10 A metáfora do *desktop*

À medida que o tempo foi passando, tornou-se difícil definir a função do computador. Ou classificá-lo dentro de um universo comum: eletrodomésticos, meios de comunicação, máquinas de cálculo, jogos eletrônicos, instrumentos de trabalho etc. O computador pode ser ao mesmo tempo meio de produção e de veiculação – cria-se nele para se ver nele: é o caso de trabalhos feitos para CD-ROMs hipermídia e das produções Web.

No entanto, sem dúvida alguma, o computador se popularizou como ferramenta. Enquanto era uma máquina de cálculo científica, ele ficou restrito aos ambientes de pesquisa. Massificou-se quando entrou nas empresas e nos escritórios para digitar cartas e memorandos, produzir impressos, fazer orçamentos, criar apresentações e guardar dados. Somente depois disto, o computador invadiu o ambiente doméstico, entreterendo e auxiliando nas tarefas do dia-a-dia.

[Uma das primeiras interfaces baseadas na metáfora do *desktop*]

Se o computador se popularizou justamente no ambiente corporativo, nada mais apropriado do que colocar a escrivadinha de trabalho dentro dele. O sucesso da metáfora do *desktop* se

deve, em grande parte, a isso. (Caso o computador tivesse se notabilizado num ambiente doméstico, talvez a sala ou a cozinha da casa fossem a metáfora dos sistemas operacionais. Se as crianças fossem levadas mais a sério, talvez a metáfora fosse a de um parque de diversões).

“A metáfora do tampo da escrivaninha foi construída a partir da crença de que usuários conhecem seus hábitos do escritório – eles estão familiarizados com o ambiente da empresa, sabem como utilizar os objetos ao seu redor (pastas de papéis, gabinetes, telefone, bloco de notas etc), e se sentem confortáveis com a idéia de um tampo de escrivaninha de escritório como espaço de trabalho. Os *designers* aplicaram esta metáfora ao mapear as coisas que os usuários fazem no computador conforme eles fariam (no escritório), se eles não estivessem usando o computador.” [idem, 25]

“A metáfora de *desktop* criada no Xerox PARC e divulgada pelo Macintosh consiste em uma metáfora de familiarização: ela se baseia nas experiências anteriores do usuário para tornar a aplicação do sistema de arquivos mais fácil de aprender e usar.” [Heckel, 1993:155]

“Ela (a interface) representa um ambiente de trabalho, onde residem projetos correntes e recursos para acessá-los. Na tela são mostrados imagens de objetos de escritório, como documentos, pastas, gavetas de arquivos, cestas. Este objetos são mostrados como pequenas imagens ou ícones.” [Smith, 1982]

Mas não basta só representar o ambiente de trabalho. Como o computador é uma máquina interativa, toda tarefa deve ser metaforizada. No entanto, o conjunto de acessórios do computador se resume a poucos dispositivos de entrada, como o teclado, o *mouse*, o *modem*, o *tablet*, o *scanner*, o microfone, a câmera de vídeo. E é através deles que o usuário vai executar as tarefas que estaria realizando numa escrivaninha de escritório. Algumas operações são praticamente iguais. Por exemplo, para gerar um documento, digitar no teclado do computador é como datilografar numa máquina de escrever. Assim como, se você tiver um *tablet*, desenhar ou assinar o seu nome com uma caneta óptica produz quase o mesmo efeito do que com lápis e papel. O mesmo vale na hora de fazer cálculos: você pressiona os números do teclado numérico como se estivesse digitando num teclado real de uma máquina de calcular. Mas quando você usa o *mouse* para abrir o programa “agenda” ou utiliza o teclado para “virar” uma página de um documento num processador de textos, isto não é nem um pouco natural. É apenas uma metáfora da ação. Mas não há saída. Ao transportar as tarefas do dia-a-dia para o mundo do computador, o ambiente virtual tem que disponibilizar também estas operações. Uma alternativa, por ora, é encarar a interface mais

como uma área de comunicação do que como um ambiente simulado e tentar tornar as tarefas realizadas dentro dela as mais naturais possíveis. Como diz S. Joy Mountford:

“Consider the desktop interface. What else could a desktop be used for? People often use a real desk as a space for communication – laying out papers and showing intermediate stages of their work, marking them up together, putting them in a different order. The desktop could be used similarly as a space for shared work as a communication.”⁴² [Laurel, 1992:27]

Evidentemente, aos poucos, tudo vai sendo digitalizado. Se através do computador, um funcionário escreve um documento de solicitação de trabalho, ele o transportará via disquete ou via rede para outro funcionário. Este poderá arrastar o documento diretamente para a área de trabalho do computador (o tampo da escrivaninha) ou guardá-lo em uma pasta para executá-lo posteriormente. O mensageiro ou *office-boy* que levaria o documento impresso, agora leva o disquete. Ou seja, o envelope ou o malote de transporte é substituído por dispositivos de armazenamento. Se o documento for enviado via rede, a figura do intermediário (o mensageiro ou o *boy*) poderá ser eliminada. A energia física do intermediário será substituída pela energia elétrica que envia dados via cabo.

O importante, no entanto, é que estas metáforas não se encaixam de modo algum nas chamadas metáforas denominativas. Não estamos mais substituindo uma palavra por outra, devido a uma lacuna lexical. A metáfora é aplicada em função da analogia existente entre uma estrutura, a concebida virtualmente por programação e que vemos na tela do computador, representada como um tampo de escrivaninha, e a outra, a real – o tampo da escrivaninha em si. Trata-se de um caso típico de metáfora-enunciado. Diz Black, “na metáfora-enunciado, estribada no *tá skhêmata tés lexeos*, o *focus* metafórico não é monolítico, não é monístico, não é singular, não é lexical, mas é interativo, global e contextual.” [Filipak, 1983:113].

O que prevalece agora não é mais a palavra, mas sim a sentença, o contexto. Quando Black fala de interação, o sentido descrito por este conceito é o da interação entre o sujeito e um

⁴² T.A.: “Considere a interface do tampo de escrivaninha. Para o que mais a escrivaninha poderia ser usada? Pessoas freqüentemente usam a escrivaninha real como um espaço de comunicação – expõem papeis, mostram fases intermediárias de seu trabalho, juntam documentos, colocam em ordem. O tampo da escrivaninha poderia ser usado de modo semelhante, como um espaço que compartilha trabalho, como um área de comunicação”

predicado lógico. Ou da interação entre dois (ou mais) termos. Para ele, “a produção do sentido é transmitida pelo enunciado como um todo” [Sacks, 1992:147].

Por outro lado, quando falamos em linguagens de computador, não temos enunciados, mas sim uma estrutura com objetos. É interessante notar que paralelamente ao aperfeiçoamento do computador, novas técnicas de programação foram criadas com o intuito de facilitar a vida do programador. Como dissemos no capítulo anterior, esta técnica, que pensa a programação de um forma mais modular, é colocada dentro de um rótulo chamado “Programação Orientada a Objetos”. O que interessa nesta discussão é que tanto na estrutura de um enunciado, quanto na de uma interface gráfica de computador (construída por programação orientada a objetos), a produção de sentido é gerada pela interação entre os seus termos (ou elementos). E assim como num enunciado literal, o modo como os termos se relacionam pode levar a uma leitura de caráter metafórico; na interface gráfica, a maneira como os objetos interagem na estrutura também coloca o usuário diante de uma metáfora.

Neste sentido, a metáfora do *desktop* pode ser enquadrada numa espécie de metáfora-enunciado. Só percebemos esta metáfora porque os elementos componentes da tela, que são as pastas, os documentos, os programas aplicativos, se inter-relacionam.

2.11 Pastas e documentos

Um dos benefícios da metáfora do *desktop* foi a de (pelo menos tentar) eliminar o conceito de diretório⁴³. Diretório é um nome de certo modo distante, frio, até mesmo um pouco tecnicista. E na interface gráfica, reelaborada pela metáfora do tampo de escrivaninha, o diretório não tem lugar. No ambiente corporativo, as pessoas guardam os seus documentos em pastas (do inglês *folder*).

⁴³ Para Theo Mandel, o conceito de diretório no Windows não foi eliminado. “Os usuários ainda têm que entender como os *drives* e os diretórios funcionam, mesmo se eles começaram a partir de uma interface gráfica”. O autor completa que para o usuário usar os ícones presentes nas interfaces gráficas, eles têm que entender os conceitos de hierarquia dos diretórios em um dispositivo de armazenamento do computador e as diferenças entre os tipos de arquivos, como os de programas e os de dados. [Mandel, 1997:163-164]

Mais do que renomear “diretório”, a novo termo, “pasta”, reconceituou a entidade. A metáfora hierarquizante presente na estrutura de diretórios dá lugar à metáfora de organização de pastas de um escritório. Neste caso, não se fala mais que um arquivo de computador está embaixo de um diretório e sim que um documento está dentro de uma pasta. Embora organogramas e hierarquia também façam parte dos ambientes corporativos, a metáfora da pasta é inspirada na rotina de trabalho do tampo de escrivaninha.

O próprio termo “arquivo” vai dando lugar à palavra documento, dependendo do contexto. A metáfora “documento” é mais apropriada do que “arquivo” para descrever os dados que foram salvos a partir de um processador de texto ou de uma planilha. No ambiente real de trabalho em que as pessoas trabalham sentados em frente a uma escrivaninha, é mais comum guardar documentos em pastas do que arquivos em diretórios. Mas por força da tradição computacional, os termos arquivo e diretório não foram de fato eliminados; pelo contrário, ainda se mantêm bastante vivos.

2.12 Janelas

Os sistemas operacionais⁴⁴ atuais são organizados por janelas. Esta característica é tão marcante que um destes sistemas é chamado *Windows*. Mas “o conceito de janela como um meio de visualizar uma parte de uma folha de papel grande foi desenvolvido por Douglas Engelbart no laboratório NLS do Stanford Institute no final dos anos 60. Seu sistema utilizava duas janelas na tela. A equipe de Alan Kay da Xerox Parc ampliou esse conceito para janelas sobrepostas nos anos 70.” [Heckel, 1993:187]

[Janela típica, com barras de rolagem]

⁴⁴ Tanenbaum em seu livro *Sistemas Operacionais Modernos* fornece uma boa descrição para sistema operacional. Segundo o autor “há muitos anos, ficou claro que dever-se-ia encontrar uma maneira de afastar o usuário da complexidade do hardware. A forma encontrada, que evoluiu gradativamente, foi a de colocar uma camada de software em cima do hardware, para gerenciar todos os componentes do sistema, apresentando-o ao usuário como uma interface muito simples de entender e de programar. Tal interface é conhecida como máquina virtual. A camada de software construída sobre o hardware nada mais é do que o sistema operacional.” O autor diz ainda que o sistema operacional é portanto um software supervisor, com o objetivo também de proteger o hardware da ação dos usuários. Não é à toa que alguns autores como Meirelles comparam o sistema operacional com um guarda de trânsito. [Tanenbaum, 1995:3]

As janelas estabelecem limites. Podem ser entre processos, ou programas, ou ferramentas ou qualquer outra entidade de um sistema. Mas não há exatamente um padrão. *“It is hard to trace the first explicit description of windows, but many designers discovered that meaningful groups of information could be assigned to specific windows.”*⁴⁵ [Shneiderman, 1987:343]. Quando uma janela é aberta, é muito comum o usuário entrar em novo ambiente. Não é à toa que cada aplicativo roda em uma janela diferente.

Numa análise preliminar, poderíamos dizer que a metáfora ‘janela’ se baseia na sua semelhança física com as janelas reais. Mas um olhar mais aguçado nos leva a crer que há algo mais por trás.

Se levarmos em conta a metáfora do tampo de escrivaninha, as janelas poderiam ser a entrada para outros departamentos. Quando abrimos uma planilha, estamos entrando no ambiente contábil do escritório. Ao carregar um processador de textos, penetramos no universo do secretariado. Se rodarmos um aplicativo gráfico nos vemos num *atelier*. O programa navegador da Internet é uma espécie de departamento de comunicação.

Embora as janelas computacionais tenham suas idiossincrasias – como a possibilidade de maximizar ou minimizá-las, de redimensioná-las e de movê-las – elas, assim como as correspondentes do mundo real, são buracos por onde podemos ver um outro ambiente. E também podem ser abertas ou fechadas, embora, no caso do computador, através de um modo bastante particular: com um clique ou o pressionar de uma sequência de teclas.

Outra peculiaridade das janelas computacionais é a existência ou não de barras de rolagem. Tecnicamente estas aparecem quando a área mostrada através da janela é maior do que o tamanho do seu vão. Como não seria possível enxergar tudo o que a janela tem para mostrar e como não podemos atravessar a nossa cabeça para ver o que há além, como acontece no caso de uma janela real, as barras de rolagem permitem deslizar a área que é apresentada. Em vez de estendermos o nosso pescoço, arrastamos a barra. Esta é uma metáfora para o movimento de nosso olhar. (Algumas pessoas chamam a barra de rolagem

⁴⁵ T.A.: “É difícil traçar a primeira descrição explícita de janelas, mas muitos projetistas descobriram que grupos significativos de informação poderiam ser atribuídos a janelas específicas.”

de elevador, por causa da semelhança gráfica da barra de rolagem vertical com o elevador de um prédio. Evidentemente, isto também é uma metáfora. Mas além da semelhança gráfica, há também o fato de tanto a barra de rolagem quanto um elevador de um prédio mostrarem níveis – andares – diferentes de uma entidade).

Por outro lado, quando as janelas computacionais são maiores do que a área mostrada através dela – algo impossível no mundo real, em volta da imagem apresentada, aparece uma moldura. Quanto menor a área mostrada, maior a moldura, e vice-versa.

2.13 Menus

O nome menu é uma referência direta aos menus de restaurante. Segundo Theo Mandel [1997,148], os menus do computador são usados basicamente para dois propósitos. Eles permitem ao usuário navegar de um ponto para outro do sistema, ou de um menu para outro. Eles permitem também selecionar itens de uma lista. Estes itens estão usualmente relacionados a propriedades ou comportamentos de objetos que os usuários escolhem para alterar. Portanto a metáfora para os menus está fortemente relacionada com a idéia de escolha.

Na Informática os menus aparecem em diversos formatos. O nome veio, no entanto, do modelo mais antigo de menu computacional: o menu de tela inteira. Antes mesmo do aparecimento da metáfora do *desktop*, muitos programas já apareciam com menus. Na verdade, um quadro apresentava uma lista de itens para serem escolhidos. Quando um deles era selecionado, um programa era executado e uma tarefa era realizada. Às vezes, antes disso, você navegava de menu a menu – ou mais especificamente de tela a tela, até chegar à escolha final. Somente mais tarde, surgiram os menus com a sua configuração atual, através de barras de menu. Hoje é possível encontrar ainda uma outra variação: os menus do tipo “*pop-up*” ou menus suspensos.

Todos os menus, no entanto, possuem uma característica em comum: são hierarquicamente estruturados. Ou seja, cada menu pode ter um ou mais submenus, que também podem ter os

seus próprios submenus e assim por diante. Por exemplo, um menu de restaurante pode ser dividido em ‘Entradas’, ‘Bebidas’, ‘Sobremesas’, ‘Pratos Frios’, ‘Pratos Quentes’. E este item em ‘Carnes’, ‘Peixes’ e ‘Massas’. E ainda está em ‘Lasanha’, ‘Espaguete’ e ‘Nhoque’. A metáfora do menu computacional é estabelecida pela semelhança estrutural com o menu de um restaurante. Um menu de restaurante é útil porque nos mostra o que há à disposição para comer e torna a escolha mais fácil. Da mesma forma, “os menus podem ser uma ferramenta útil porque, ao apresentar as opções de que o usuário dispõe, eles lhe dão uma oportunidade de reconhecer o que deseja fazer.” [Heckel, 1993:60]

2.14 Botões

Os botões são representações de botões reais, portanto são botões virtuais. Mas estes também podem ser entendidos também como uma metáfora do botão do mouse. Ou uma extensão deste botão. Ao clicar no botão da interface, estamos executando virtualmente a ação física de clicar no botão do mouse.

Em geral, no entanto, os botões virtuais são representações dos diversos tipos de botões que encontramos no dia-a-dia: botões de painéis, de aparelhos elétricos, de portões eletrônicos etc. O apuro visual dos botões virtuais é tal que existem representações para os três estados do botão: no seu estado normal (ou inercial), quando ele é pressionado (ou invertido) e quando está desabilitado. Em geral, todos são desenhados em três dimensões, para que a representação fique ainda mais realista. A representação do estado normal e do invertido, em geral, é obtida da simulação do que ocorre quando um feixe de luz incide na superfície do botão; isto provoca sombreamentos diferentes. No caso do estado desabilitado, o botão é representado numa cor acizentada, com os desenhos da superfície em baixo relevo. Portanto, mais do que metáforas, estes botões são representados, simulados, porque, ao clicar em cima do botão, a imagem do estado normal é substituída pela do estado invertido; e ao liberar o botão do mouse, a imagem do estado normal volta a aparecer. Esta ação provoca no usuário a sensação de que o botão virtual foi de fato pressionado.

Os sistemas operacionais atuais ainda usam mais dois tipos de botões: os *radio buttons* e os *check boxes*. Os primeiros lembram os orifícios dos painéis de chaves das antigas

telefônicas. Os *check boxes*, por outro lado, têm o formato de quadrados. E são metáforas das caixas de opção, típicas de formulários em papel e muitos utilizados também nos exames vestibulares para questões de múltipla escolha. Portanto, ao clicar neles, o usuário deixa uma marca de “x”. Ambos são botões de seleção. Mas enquanto os primeiros só aceitam uma escolha por vez, os *check boxes* são multi-selecionáveis.

2.15 Ícones

No capítulo anterior, analisamos a inserção de ícones (no sentido peirciano do termo) em computadores, tendo em vista a aplicação por ora generalizada de interfaces gráficas em sistemas computacionais.

No entanto a palavra ícone adquiriu um significado próprio para os usuários de computador, ligeiramente distinto de sua definição peirciana. Os ícones do computador são ícones que funcionam como pontos-quentes. Ou seja, são pontos de interação. Podem ser clicados (ou duplo-clicados) para que se execute alguma ação.

Mas por serem ícones também no sentido peirciano eles possuem uma representação. Esta representação de alguma forma tem algo a ver com a ação a ser executada (que é o objeto de fato para qual ele foi representado, o *Segundo* conforme Peirce). Neste sentido, poderíamos dizer que o ícone do computador é, na verdade, um índice, porque ele “indica” o que vai ser realizado quando clicamos nele. Mas uma segunda interpretação possível é que eles operam como metáforas, pois representam e substituem instruções ou programas de computador e suas ações correlatas. Ou seja, nas interfaces computacionais, o ícone não só faz o papel de um botão (porque é clicado), mas também sua representação é contextualizada dentro de um ambiente metaforizado.

Assim, ícones representam diferentes tipos de objetos (programas, pastas, arquivos, unidades de discos, dados, o atalho de um arquivo) que executam diferentes ações: ao clicar (ou duplo-clicar) no ícone representativo de um programa, este é aberto; ao clicar em um ícone que representa uma pasta, esta é aberta, mostrando os arquivos que estão dentro dela;

e assim por diante. A grande vantagem de se trabalhar com ícones é que eles, por serem ícones no sentido peirciano, de alguma maneira identificam a ação a ser executada.

Portanto, o ícone de um caderno pode ser um *link* para um editor de textos, o ícone de uma porta pode muito bem identificar a saída de um aplicativo e assim por diante. A representação visual da porta, identificada no ícone do computador, não deixa de ser o ícone de uma porta no sentido peirciano do termo. Mas quando se clica no ícone computacional de uma porta e um programa é fechado, este ícone, no sentido peirciano do termo, passa a ser uma metáfora. Estamos utilizando a imagem de uma porta não para representar uma porta, mas para representar uma ação associada: a idéia de abrir e fechar. Ou seja, a metáfora está presente na idéia de que assim como fechamos uma porta; também fechamos um aplicativo.

2.16 Recortar-Copiar-Colar

Hoje, muitos dos programas podem compartilhar os dados entre si. Ou seja, o usuário pode digitar um texto em um editor de textos, salvá-lo, e depois inseri-lo ou importá-lo num programa gráfico. Ou vice-versa, pode criar uma imagem num *software* gráfico e colocá-la em um processador de textos. Além disso, os sistemas operacionais atuais são todos multitarefas, o que significa que um usuário de microcomputador pode rodar simultaneamente mais de um programa ao mesmo tempo. Mas estas duas características em conjunto (compartilhar dados e trabalhar com programas simultaneamente) permitem que um dado possa trafegar de um programa para outro sem a necessidade de salvá-lo. Ou seja, “pegamos” o dado de um programa e o “levamos” diretamente para outro programa. No procedimento anterior, antes de ser inserido por um comando do tipo “importar”, o dado tinha que ser guardado em um outro arquivo, que fazia o papel de intermediário.

Na realidade este ato de “pegar” uma informação, como foi descrito acima, pode ser feito por dois caminhos: na primeira possibilidade, pegamos, de fato, o dado, isto é, “tiramoss” o dado do lugar onde estava; numa segunda possibilidade, “levamos” uma cópia deste dado, “deixando” o original no lugar em que se encontrava originalmente. O dado “pegado” é sempre levado para um lugar intermediário, chamado “Área de Transferência”. A partir do

momento que o dado está nesta área, quem precisar, pode ‘pegá-lo’, quantas vezes quiser, e levá-lo para outro lugar (programa). Mas se um outro dado passar a ocupar esta área de transferência, um anterior, que estave lá, deixa de ocupá-la, fazendo valer a máxima que diz que dois corpos não podem ocupar o mesmo espaço.

Este complexo operacional foi traduzido nos sistemas por uma metáfora que esclarece muito bem o processo: a do ‘papel em branco’ (do inglês, *plain-paper metaphor*). Para compreendê-la basta rememorarmos as tardes perdidas em nossa infância com as brincadeiras de colagens. Pegávamos um papel, desenhávamos nele, depois com uma tesoura recortávamos parte dele e colávamos em outro papel. Ou então, quem não se lembra da ‘mágica’ do tiner: es fregando um paninho banhado neste líquido em cima de uma imagem, poderíamos obter a mesma representação num papel em branco que foi colocado embaixo dela. Portanto, segundo esta lógica, a partir de um desenho, podemos ‘reimprimi-lo’ num novo papel em branco, através de sua cópia ou de um recorte, acompanhado de uma colagem no novo papel.

Na metáfora computacional do papel em branco, o usuário seleciona um conjunto de dados. A partir de então, ele pode copiar ou recortar os dados selecionados. No primeiro caso, uma cópia é gerada e levada à Área de Transferência, portanto os dados originais são mantidos em seu programa de origem (como no caso do tiner acima descrito). No segundo caso, os dados são levados à Área de Transferência, mas como bem diz o comando, você recorta os dados originais: eles são suprimidos do programa em que foram gerados (como no caso da tesoura). A partir do momento em que os dados estão colocados na Área de Transferência, o usuário pode colá-los em qualquer outro programa que os aceite⁴⁶.

Os comandos copiar, recortar e colar tornaram-se bastante populares e parte deste sucesso deve-se ao fato de terem sido apresentados por uma metáfora: a do ‘papel em branco’. A aplicação destes comandos, hoje, é tão comum que alguns usuários descrevem-nos pela

⁴⁶ Na verdade nem sempre é possível compartilhar dados entre programas. Por exemplo, os editores de texto, como o Bloco de Notas do Windows, simplesmente não aceitam imagens. Eles foram desenvolvidos para trabalhar somente com textos. Neste caso, você só pode colar textos neste tipo de programa.

concatenação de seus nomes: “faça um *copy-paste*” (de copiar -colar, em inglês), que é, na verdade, uma espécie de sinédoque⁴⁷.

Mas a metáfora do “papel em branco” representa algo mais do que simplesmente um jogo de colagem. Como afirma Mountford,

*“Typically, different applications have to be launched separately depending on the kinds of data to be created; writing, drawing, and painting are all separate applications and different data types. Individual elements must be pasted together, sometimes requiring that they be transformed into new data types, in order to create a finished document. The observations above have led to the further refinement of a set of interface ideas often referred as the plan-paper metaphor, likening the screen to a blank sheet of paper. Currently, users must break work up into separate tasks and perform each within its associated environment – taking the tasks to the tools. The plain-paper metaphor would be designed to correspond to ‘real -world’ uses.”*⁴⁸ [Laurel, 1992:26]

2.17 Processador de Textos

O processador de textos é “a máquina de escrever em cima do tampo da escrivaninha”. Mas provavelmente a maior herança das antigas máquinas está no teclado. Teclas como *Tab* e *Caps Lock* atestam isto claramente. A tecla *Tab* da máquina de escrever, por exemplo, é a responsável pelas tabulações, que marcam os pontos em que o carro da máquina deve avançar.

⁴⁷ “Do grego: *syn* = juntamente + *ekdeskhestai* = tomar, recolher. É uma figura de metáfora aristotélica (metáfora-palavra) que, partindo do “*pars pro toto*” = da parte para o todo (sinédoque particularizante) e do “*totum pro parte*” = do todo para a parte (sinédoque generalizante), imprime à figura um significado figurado.” [Filipak, 1983:Glossário]. Na definição de Cherubim, “a sinédoque é um tropo imperfeito porque não oferece uma real e perfeita translação de sentido, mas tão-somente uma simples extensão do sentido vocabular, baseada em relação de contigüidade, de vizinhança ou de coexistência para alastrar ou restringir a significação normal das palavras. São vários os casos em que ocorre: 1. *tomando-se a parte pelo todo* [...]. 2. *tomando-se o todo pela parte* [...]. 3. *tomando-se o singular pelo plural* [...]. 4. *tomando-se o plural pelo singular* [...]. 5. *tomando-se o indivíduo pela espécie* [...]. 6. *tomando-se a espécie pelo indivíduo* [...]. 7. *tomando a espécie pelo gênero* [...]. 8. *tomando-se o gênero pela espécie* [...]. 9. *tomando-se o abstrato pelo concreto* [...]. 10. *tomando-se a matéria pela obra*.” [Cherubim, 1989:61]

⁴⁸ T.A.: “tipicamente, aplicações diferentes têm que ser carregadas separadamente dependendo do tipo de trabalho que vai ser criado; escrever, desenhar, e pintar são executadas por aplicativos separados, com diferentes tipos de dados. Elementos individuais devem ser colados no mesmo aplicativo, algumas vezes transformados, para criar o documento final. As observações acima nos conduzem a um refinamento que vai além das idéias interfaciais que freqüentemente explicam a metáfora do papel em branco, ou seja, uma tela que representa um pedaço de papel em branco. Normalmente, os usuários devem fragmentar o trabalho em tarefas separadas e executar cada uma em seu ambiente relacionado – cada tarefa com suas ferramentas. A metáfora do papel em branco teria sido desenhada para corresponder às aplicações do mundo real.”

Isto é muito útil para a criação de tabelas.

Pesquisas posteriores atestam que os teclados poderiam suprimir algumas teclas e mesmo mudar a sua configuração. Alguns novos teclados têm surgido, mais inteligentes, mais ergonômicos e mais enxutos. Com o tempo, o teclado vai se flexibilizando para atender novas prerrogativas e novas necessidades, não só de processadores de textos, mas também de outros programas. Mas talvez nenhuma outra relação, entre aplicativos e dispositivos, tenha sido tão indissolúvel quanto a que existiu entre o teclado e os processadores ou editores de texto⁴⁹. Um influenciou na criação do outro.

Mas se os processadores de texto são uma metáfora das máquinas de escrever, a superioridade do correspondente virtual tornou-se tão evidente que praticamente sepultou a existência da máquina real. As vantagens são inúmeras e poderíamos escrever um tratado só sobre elas. Com diz Levy:

“Os programas de edição de textos permitem suprimir com facilidade uma letra, uma palavra, um parágrafo, e fazê-los reaparecer aqui ou acolá, com a paginação se reorganizando automaticamente. As adições, correções e modificações não exigem mais reescrever toda uma página ou até um texto inteiro, ou ainda o laborioso manuseio de cola e tesoura. Mal a alteração foi feita e a apresentação fica clara. Torna-se muito fácil alterar as margens, o espaçamento entre as linhas, a divisão em parágrafos, a tipografia, todos os elementos para que contribuam para a valorização de um texto. Graças às capacidades de armazenamento e restituição da máquina (o computador, *grifo meu*), podem-se facilmente reaproveitar frases, trechos de texto ou parágrafos inteiros em contextos diferentes, com a simples mudança de algumas palavras para evitar uma total reiteração.” [Lévy, 1998:17]

2.18 Planilha

A planilha eletrônica é um dos softwares mais populares nos dias de hoje. “A metáfora de planilha evoca associações com as planilhas feitas em papel, as habilidades que milhões de pessoas têm de resolver problemas em planilhas desse tipo, e os milhões de problemas que foram resolvidos com métodos manuais de planilhas.” [Heckel, 1993:188]. Assim um dos

⁴⁹ Há uma pequena diferença entre um editor de texto e um processador de textos. Enquanto este fornece um verdadeiro conjunto de funções, como formatação de palavras, margem, configuração de páginas etc; o segundo é apenas “um programa para gerar e editar texto. Ajuda a redigir e é um dos componentes de um processador de textos” [Meirelles, 1994:597]

méritos das planilhas eletrônicas é o de trazer à mente do usuário sua experiência anterior com planilhas e as coisas que ele pode e não pode fazer com elas.

No entanto, a possibilidade de usar células para fazer cálculos através de fórmulas e o seu posterior recálculo natural é a imagem mais fortemente identificada com este tipo de *software* e em grande parte o segredo de sua popularidade⁵⁰. A planilha eletrônica é uma história de sucesso porque utilizou a mesma estrutura das planilhas convencionais; e em cima desta metáfora colocou de maneira inovadora o que há de especificidade na informática: a possibilidade do cálculo automático, de ordenação, de classificação e de análise dos números em função de diversos cenários, a previsão por metas e a criação de auditorias.

2.19 Outros programas

Como vimos, as interfaces gráficas se aproveitaram das metáforas para se fazer compreender. Como nos casos dos processadores de texto e das planilhas, os programas aplicativos que surgiram a seguir tiraram proveito desta característica, mesmo porque a partir de então quem não seguisse esta nova prerrogativa seria discriminado.

Assim foram criados programas de editoração eletrônica, cuja “metáfora faz associações com o processo de corte e colagem de textos e de *layouts* gráficos, com a tipografia de diferentes tipos e tamanhos de fontes, e com as habilidades artísticas de *designers* gráficos.” [idem,188].

Ao seguir o modelo de banco de dados relacional, os programas de BD transformaram-se em metáforas de tabelas. Os programas de apresentação seguem a mesma estrutura das apresentações realizadas com retroprojektor. E os exemplos não param por aí. Ao se criar um programa que é a metáfora do aparelho, da máquina, do procedimento ou da tarefa que era

⁵⁰ A idéia de incluir equações em células parece ter sido desenvolvida pela primeira vez por Pardo e Landau (que a patentearam), sendo usada em computadores de tempo compartilhado no início dos anos 70. Dan Bricklin e Robert Frankston reinventaram a planilha de Rahenkamp e as fórmulas de célula de Pardo e Landau e incluíram janelas para criar a primeira planilha de sucesso no mercado: o VisiCalc. Fonte: [Heckel, 1993:188]

realizada de modo convencional, o usuário identifica rapidamente a sua maneira de operar, porque tem familiaridade com o processo anteriormente realizado sem o computador. E este é o grande segredo da utilização de metáforas para a construção e estruturação da interface do *software*. Os *softwares* são os equivalentes virtuais das máquinas e aparelhos de nosso dia-a-dia. E nada é mais explícito neste sentido do que a calculadora do sistema operacional: ela é uma apresentação praticamente realista de uma calculadora de bolso. Praticamente deixa de ser metáfora para ser uma realidade virtual.

Neste capítulo pudemos perceber que a metáfora esteve fortemente presente no decorrer da história do computador. E não é por coincidência. Ela, de uma maneira ou de outra, desempenhou o papel de facilitadora; esteve presente na interface entre o mundo dos circuitos e da linguagem da máquina, de um lado, e o da nossa percepção, de outro. No entanto, a presença da metáfora sofreu processos mutatórios que acompanharam o avanço tecnológico do computador.

No início, a aplicação da metáfora era basicamente lexical, porque a interface do computador também era literal. Prosperaram as chamadas metáforas denominativas. Quando as interfaces se tornaram gráficas, as metáforas estiveram presentes dentro de um contorno estrutural. Os ambientes apresentados pela tela do computador eram metáforas de espaços reais: o tampo de uma escrivaninha, uma planilha de cálculo etc. Entramos num terreno explorado pela metáfora-enunciado.

Agora, estamos partindo para uma nova fase. No capítulo seguinte exploraremos mais esta questão. Veremos que o computador, graças a esta sua capacidade de explorar a virtualidade, pode conceber máquinas que simulam aparelhos e ambientes reais. As representações estão se aperfeiçoando de tal modo, que, tal como é descrito em uma peça de ficção científica, não poderemos mais distinguir o que é real e o que é virtual.

— CAPÍTULO 3 —

**Tendências recentes no uso de metáforas:
das interfaces gráficas à realidade virtual**

3.1 Introdução

Na última década, o computador entrou definitivamente no cotidiano das pessoas. O aumento de sua velocidade de processamento e de sua capacidade de memória colocou-o como uma ferramenta de otimização para a realização de diversas tarefas domésticas e corporativas. Como ele, o usuário escreveu cartas ou relatórios, fez seu planejamento contábil, projetou edificações, utilizou-o no processo de criação e geração de impressos, editou seqüências de vídeo, alterou imagens fotográficas e até mesmo usou-o como suporte para escutar música. Nesta onda, decretou a morte da máquina de escrever, tornou obsoletas as calculadoras científicas e destruiu a fidedignidade da fotografia como prova (pois, hoje, uma foto pode ser digitalizada e depois completamente alterada por computador). De um modo geral, o computador vai ficar marcado na década de noventa como um ferramenta, particularmente uma ferramenta de uso pessoal.

Mas esta década viu surgir também um novo movimento, que começou discreto no início, mas que se popularizou de forma definitiva no seu fim: os computadores entraram em rede mundial e passaram a se tornar também poderosos meios de comunicação. O nome desta rede: Internet. Por traz de uma estrutura gigantesca, uma idéia muito simples: usar o seu computador pessoal para se comunicar com pessoas de qualquer ponto do planeta, bastando para isto um meio de transmissão como a linha telefônica.

Paralelamente, e sem muito estardalhaço, a computação continuou avançando nas pesquisas sobre a “Realidade Virtual”. Após um período de euforia com uma tecnologia que poderia materializar o sonho humano de experimentar realidades não palpáveis, como o ato de voar, o mergulho no mundo microscópico ou a viagem pelo túnel do tempo, este novo paradigma de utilização computacional obteve os seus primeiros resultados concretos no fim da década passada e no início desta.

Neste capítulo, vamos analisar a evolução das interfaces computacionais e suas metáforas a partir destes dois focos: a popularização da Internet e a evolução da Realidade Virtual.

3.2 O avanço tecnológico que levou à Internet

A Internet é o resultado da combinação de dois conceitos elementares na computação: o computador pessoal e as redes de comunicação. As primeiras pesquisas para a interligação de redes de computadores começaram no início dos anos sessenta pelos militares. Mas somente no final daquela década, estes obtiveram resultado com a ARPA (*Advanced Research Projects Agency*), o nome que o Departamento de Defesa norte-americano deu para a sua rede. O objetivo era interligar diversos centros de pesquisas sem que houvesse um comando central de controle, pois, na época, a Guerra Fria vivia momentos bastante delicados. Era necessário que todos os pontos da rede tivessem o mesmo valor hierárquico, de modo que se um deles fosse destruído, isto não provocaria a desativação da rede como um todo. Gilles Deleuze e Felix Guattari apresentaram um maneira bem interessante de entender a estrutura desta rede. Compararam-na com um rizoma, pois este pode ser quebrado em um ponto, sem que seja destruído no todo; e mais do que isto, a partir de qualquer um de seus nós pode-se construir novos nós. “A Internet tem, como rede, uma grande capacidade de autogênese.” [Leão, 1999:22]

Nas décadas de setenta e oitenta, a idéia de uma grande rede mundial evoluiu sob diversos nomes e tecnologias: Usenet, Csnnet, Decnet, Bitnet, Janet, NSFnet etc. Mas neste período, a rede mundial era um reduto quase exclusivo da comunidade acadêmica. Basicamente sua função era a transferência de arquivos e a implementação de algumas sessões remotas. Uma das maiores limitações para popularização da rede em caráter mundial era a falta de padronização entre as diversas tecnologias do computador, pois estes, além de serem de diversos portes, usavam sistemas operacionais e interfaces muito diferentes. Uma saída foi a adoção de um protocolo de comunicação comum que conseguisse “falar” com todos os tipos de computadores: o TCP/IP. TCP (*Transmission Control Protocol*) é o protocolo que a ARPA tinha criado para assegurar que as mensagens seriam apropriadamente encaminhadas de um ponto a outro e que estas mensagens chegariam intactas. Posteriormente, a ARPA criou também o protocolo IP (*Internetworking Protocol*) que definia os detalhes de comunicação entre diversas redes locais de comunicação. A fusão

deste dois protocolos deu origem à Internet como a definimos hoje. Para a popularização da Internet, a conexão por telefone também teve um papel relevante. Como era uma estrutura mundial já instalada, para o usuário “entrar na rede”, bastava ter um computador pessoal, comprar um modem e pagar por um provedor de acesso.

Com o protocolo TCP/IP o usuário da rede poderia usar diversos serviços, como correio eletrônico, grupos de discussão, transferência de arquivos e conexão remota. Para isso, ele só precisava instalar no seu computador os programas que executassem cada um destes serviços. Era um avanço significativo, mas ainda um tanto quanto sem sal para os padrões comerciais. Para se ter uma idéia, para usar a Internet nesta época, o usuário às vezes precisava tomar conhecimento de UNIX, um sistema operacional baseado numa interface por linhas de comando. Mas no início dos anos noventa, Tim Berners-Lee, do CERN, um laboratório de pesquisas europeu, na Suíça, colocou em prática uma idéia que popularizou definitivamente o uso da rede: a WWW (*World Wide Web*). Ao contrário dos outros serviços, a WWW é baseada numa interface gráfica. Com ela, o usuário usa manipulação direta e interage com a interface através de ações do *mouse*. Além disso, ela pode ser composta por elementos multimídia, como texto, som, imagens, animações e filmes⁵¹.

A idéia de Tim Berners-Lee era bastante simples, mas revolucionária: criar um programa universal no sentido funcional, mas que tivesse uma versão específica para cada computador e sistema operacional. Este programa, tipicamente chamado de *browser* ou navegador, poderia receber grandes blocos de texto puro⁵², que depois seriam convenientemente formatados na página da interface, de acordo com a intenção do autor. Para que o *browser* pudesse interpretar o modo de visualização da página, foi criada uma linguagem comum: o HTML (*HyperText Markup Language*), que fica embutida no meio do texto. Assim, além do texto, que aparece normalmente na tela, o criador da página deveria

⁵¹ Na verdade, no início, nem todos estes elementos apareciam com freqüência. Por serem muito grandes, arquivos sonoros e de vídeo demoravam muito para serem transferidos numa Internet baseada principalmente pelas lentas conexões por linhas telefônicas convencionais. Somente no início desta década, começaram a ser oferecidos para o público em geral outras formas de conexão mais rápidas, como cabo, fibra óptica ou linhas telefônicas com tecnologia de alta velocidade.

⁵² Texto puro é um texto sem formatação, que só apresenta o código binário de cada caracter. Cada símbolo (letras, numerais, caracteres especiais) possui um código binário convencionado. Há diversas convenções que estabelecem qual o código de cada caracter. A mais utilizada é a ASCII (American Standard Code for Information Interchange)

conhecer também uma série de palavras-chaves, marcadas entre um sinal de maior e menor (<>), mas que não seriam mostradas na página. Estas palavras-chaves, chamadas de *tags*, serviriam apenas para indicar os aspectos de formatação do texto e de diagramação da página. Por exemplo o envio da seqüência “Título” faz o texto “Título” aparecer formatado em negrito (*Bold*, em inglês) na página, devido à *tag* .

Cada página produzida em HTML tem um endereço dentro da rede mundial. Para que um usuário de qualquer ponto do planeta possa visualizar uma das páginas, basta ele digitar o endereço dela num campo de inserção que existe no *browser* e depois clicar o botão “Ir”. Este endereço é chamado de URL (*Uniform Resource Locator*) e é definido segundo regras lógicas estabelecidas internacionalmente. Um URL típico seria <http://www.usp.br/index.html>.

Mas esta não é única maneira de “ir” para uma página WWW. A linguagem HTML permite definir no corpo da página alguns elementos que funcionam como pontos de interação para outros locais da rede. Tipicamente são chamadas de *links* (elos ou nós). Qualquer elemento da página pode ser um *link*, desde uma palavra, uma imagem ou até mesmo uma animação. Esta característica da linguagem HTML coincide com a estrutura rizomática da rede. Quando o autor programa a existência de um *link* em sua página, o usuário pode se transferir para o endereço associado a ele, através de um dos eventos usuais da manipulação direta, normalmente um clique. Obviamente, uma página pode ter diversos *links*. Para tornar evidente o *status* de *link* de um elemento interfacial, geralmente ele é destacado com uma propriedade diferente. No caso de ser uma palavra, o *link* é geralmente formatado em azul e sublinhado.

3.3 Hipermídia e navegação

Uma das conseqüências mais fantásticas desta estrutura é que a *Web* permite que cada leitor construa um modo muito pessoal e livre de leitura, porque a organização dos blocos de informação é não-linear. Na verdade, isto não é uma novidade. Muitos artistas, cientistas e teóricos já tinham tido o sonho de implementar uma leitura não-linear muito antes da implantação da Internet. Um deles, o físico e matemático Vannevar Bush, apresentou, a

partir de seu artigo “*As we may think*”, os conceitos do que seria um texto interativo. Sua tese central era que a leitura de um texto deveria se processar do mesmo modo como intercalamos as informações em nossa mente: por associações. Para tornar clara esta idéia, o autor chegou mesmo a conceber um dispositivo que poderia armazenar e ligar estas informações, de forma a colocar em prática este tipo de leitura: o *Memex*. Anos mais tarde, Ted Nelson inventou o termo “hipertexto” para especificar esta forma permutativa de apresentação de informações. Mais tarde, este conceito foi atualizado para *hipermídia*, uma vez que a apresentação e os elos de associação das informações não eram mais limitados ao texto, pois já poderiam ser colocados por outras formas de representação. Como acontece hoje na WWW, nos CD-ROMs de caráter multimídia ou nos atuais DVDs.

A boa nova que esta nova estrutura de apresentação de informações nos trouxe foi a colocação do leitor como uma espécie de co-autor da obra interagida, que nos remete a outro dos sonhos de muitos artistas: a arte interativa. Como afirma Arlindo Machado, no artigo “Hipermissão: O Labirinto como Metáfora”, “a diferença introduzida pela informática é que ela dá um aporte técnico ao problema” [Machado, 1997:145], que antes era só teoricamente introduzido. O autor complementa:

“A disponibilidade instantânea de todas as possibilidades articulatórias do texto verbo-audiovisual favorece uma arte da combinatória, uma arte potencial, em que, em vez de se ter uma ‘obra’ acabada, tem-se apenas seus elementos e suas leis de permutação definidas por um algoritmo combinatório. A ‘obra’ agora se realiza exclusivamente no ato de leitura e em cada um desses atos ela assume uma forma diferente, embora, no limite, inscrita no potencial dado pelo algoritmo. Cada leitura é, num certo sentido, a primeira e a última. O texto verbo-audiovisual já não é mais a marca de um sujeito (visto que o sujeito que o realiza é outro: o leitor-usuário), mas um campo de possíveis, de que o sujeito atualizador realiza parte de suas possibilidades. Pode-se inclusive dizer, que com a obra combinatória, a distribuição dos papéis na cena da escritura se redefine: os pólos autor/leitor, produto/receptor cambiam de forma muito mais operativa.” [idem, 146]

A contra-indicação desta estrutura permutativa e combinatória de apresentação das informações é que o usuário pode se sentir perdido dentro de um emaranhado de dados que não possuem uma linha orientadora e definida de leitura. A atualização das informações na tela do computador é feita a partir da seleção mais ou menos livre de um clique dentre tantos que o leitor pode ter à sua disposição; e depois por outro clique e outro clique etc. Como afirma Machado, “o processo de leitura é designado pela metáfora bastante apropriada de navegação, pois se trata realmente de ‘navegar’ ao longo de um imenso mar

de textos que se sobrepõem e se tangenciam.” [idem, 147] Outra metáfora bastante difundida atualmente para demonstrar o processo bastante particular de leitura na hipermídia é o ato de surfar pela *Web*. Há muitas imagens que talvez expliquem a utilização deste conceito, mas destaco particularmente duas: o surfe é um esporte radical, praticado geralmente por jovens, assim como a *Web*; e a Internet é considerada uma nova onda, “tão legal de surfar”, quanto as ondas reais.

É interessante notar que tanto “navegar” quanto “surfar” são ações ligadas ao mar. O mar nos remete ao volume, à quantidade, à imensidão. Indiretamente, portanto, a Internet é colocada como uma metáfora do mar, presente em todo o mundo, mas que nos fornece em vez de água, um número incomensurável de dados.

Por outro lado, Umberto Eco considera que o aspecto negativo é a própria abundância de informações: “o excesso pode ser péssimo, porque não se consegue encarar-lo e escolher o que presta.”⁵³. Navega-se portanto por um dilúvio de dados. Temos a sensação de que informações excessivas é o mesmo que não tê-las. E como “navegar, em linhas gerais, é a arte de encontrar um caminho que leve de um lugar a outro” [Leão, 1999:122], também nos sentimos tão desorientados com o excesso de caminhos quanto nos sentiríamos na ausência deles (um deserto, por exemplo). De modo muito pertinente, portanto, Lucia Leão lembra também do sentido poético que a metáfora da navegação apresenta, pois ela “nos remete a imagens como desbravamento, aventura, jornadas etc” [idem,123]

Outra metáfora adequada para demonstrar o processo de desorientação que o leitor sente diante do excesso de informações apresentadas por estruturas permutativas como a Internet é o da “hipermídia como um labirinto”. “A melhor metáfora para a hipermídia é a do labirinto, pois a hipermídia reproduz com perfeição a estrutura intrincada e descentrada deste último.” [Machado, 1997:149] A interatividade, neste sentido, pode ser uma armadilha, pois do mesmo modo que nos oferece mais dados, pode nos levar a caminhos sem volta. Lucia Leão, em seu livro “O Labirinto da Hipermídia: Arquitetura e Navegação no Ciberespaço”, explora o tema da complexidade da leitura não-linear na hipermídia. Em

⁵³ “O dilúvio da Informação”. IN: Veja: Vida Digital. Encarte Especial da Revista Veja. São Paulo. Ano 33, nº 52, p.11-15.

determinado momento ela expõe a própria experiência de pesquisa que teve na Internet para a realização de seu livro:

“Muitas vezes senti a experiência do labirinto ao recolher tantos artigos polêmicos e divergentes sobre o tema. Enfim, durante a pesquisa realizada para este livro, vivi profundamente o aspecto duplo e paradoxal da experiência hipermidiática. Por um lado, convivi com a euforia de estar conectada com o resto do planeta, discutindo questões emergentes e colhendo informações preciosas. Por outro, sofri para conseguir manusear uma profusão de dados. Senti a vertigem e a necessidade de voltar o tempo todo à já conhecida e segura tecnologia do texto impresso linearmente.” [Leão, 1999: 25]

Por outro lado, a vertigem sentida pelo leitor, uma espécie de sensação simultânea de prisão e infinitude, não é sentida pelo projetista. “Para seu arquiteto, o labirinto é finito. Para o viajante, por sua vez, devido às similitudes das encruzilhadas, aos caminhos aos quais retorna mesmo sem querer, o labirinto se apresenta como infinito.” [idem, 114] É verdade que muitas vezes o usuário quer se deixar perder, investigar sem compromissos e responsabilidades a infinitude de informações de sistemas hipermídia. No entanto, nem sempre é desejável para o leitor se perder num mundo de encruzilhadas e becos. Às vezes ele quer um mapa, pois deseja percorrer o labirinto da hipermídia com um objetivo determinado. O mapa, ao mesmo tempo, que mostra a visão geral do labirinto e quebra a característica de infinitude, mostra os caminhos para tirar o leitor da prisão. Só o arquiteto-projetista pode fornecer este mapa. Por isso, uma das técnicas costumazes de eliminar a vertigem do usuário é a elaboração de mapas e metáforas para a navegação em sistemas hipermídia.

3.3.1 Mapas e metáforas de sistemas hipermídia

Romero Tori [Tori, 1996:28] apresentou uma interessante analogia entre o usuário de sistemas hipermídia e o turista. Por um lado, existem os turistas que preferem viajar através de excursões, por serem mais seguras e por ter um roteiro pré-definido. Por outro lado, há aqueles que preferem traçar livremente seus próprios caminhos, escolher o seu hotel e definir o período em que vai ficar em cada cidade. Os mais radicais e aventureiros gostam de redefinir constantemente o seu roteiro inicial e não se incomodam de se perder. Em sistemas hipermídia ocorre algo similar. De um lado, há os usuários que se sentem completamente perdidos numa estrutura não-linear que caracteriza estes sistemas. Por outro

lado, há os navegadores que se deixam levar pelo sabor dos cliques e não se incomodam em saborear a falta de orientação da hipermídia.

Mas todo viajante, do mais tradicional ao mais rebelde, gosta de ter alguma noção de direção em determinados momentos. Algumas vezes nosso objetivo não é se deixar levar por caminhos livres; particularmente quando temos pouco tempo e nosso objetivo é bem específico: encontrar uma informação; é quando queremos ir direto ao ponto. Passar por *links* e locais que não nos interessam nos deixam estressados e pode tornar falha a nossa compreensão. Thüiring, Hannemann & Haake afirmam que: *“comprehension is often characterized as the construction of a mental model that represents the objects and semantic relations described in a text. The readability of a document can be defined as the mental effort spent on the construction process.”*⁵⁴ [Thüiring, 1995:57] Segundo estes autores dois fatores são decisivos para a construção de um modelo mental. Um deles é positivo: a existência de coerência entre as informações. O outro fator é negativo: a sobrecarga cognitiva.

A coerência é classificada por eles em coerências locais, que ocorre da conexão entre dois *links*, ou globais, formada pela associação de todas as informações lidas. Mas, “a simples atitude de seguir uma série de *links* não é suficiente para criar um todo coerente na mente do leitor” [Leão, 1999:129], principalmente se no caminho houver uma série de ruídos de comunicação que tiram a atenção dele. Isto pode ocorrer pela sobrecarga de informações sem interesse que o usuário cruza no seu caminho.

A sobrecarga cognitiva ocorre também quando o leitor tem que dividir a leitura de um documento com outras atividades necessárias para a sua realização. Quando lemos um livro ou assistimos a um filme praticamente nos concentramos na simples tarefa de recepção dos sinais. Ao folhear um livro, o máximo que precisamos fazer é virar as páginas. No caso de um filme, o esforço é ainda menor, principalmente quando estamos no escuro de uma sala de cinema. Toda nossa atenção se concentra exclusivamente na tela de projeção. Já a

⁵⁴ T.A.: “A compreensão é freqüentemente caracterizada como a construção de um modelo mental que representa os objetos e as relações semânticas descritas em um texto. A capacidade de leitura de um documento pode se definida como o esforço mental despendido neste processo de construção”

hipermídia, ao nos colocar como co-autores, acrescenta outras obrigações no processo de leitura. “A atenção do navegador fica dividida entre várias preocupações, tais como: assimilação do conteúdo, operação do sistema, orientação no hiperespaço e organização da pesquisa e de seus objetivos.” [Tori, 1996:29] A esta série de problemas que ocorrem no processo de leitura de hiperdocumentos costuma-se denominar “desorientação”. *‘It occurs when readers do not know where they are, how they got there or where they should go next.’*⁵⁵ [Thüring, 1995:59]

Esta sensação pode ser atacada pela criação por parte do projetista de mapas e metáforas de estrutura. É como se o arquiteto fornecesse o mapa do labirinto. De um certo modo, isto acarreta a subtração do aspecto surpresa que caracteriza a navegação sem orientação de sistemas hipermídia. Ao ter o mapa, o navegante sai do labirinto. “O labirinto só passa a existir como tal, como construção da complexidade, na medida que alguém o penetre e o percorra. Para o seu construtor, que tem a visão geral do projeto, que tem o mapa, o labirinto não se impõe como metáfora do obtuso, do complexo.” [Leão, 1999:114]

No entanto, a navegação livre é interessante quando o usuário deseja brincar, se deixar levar pelo labirinto da hipermídia. Quando deseja ir direto a um ponto, o labirinto pode ser um fator de estresse. Com um mapa na mão, o leitor tem a opção de usá-lo ou não, como faz um turista. Quando quer encontrar um lugar específico utiliza o seu mapa, mas quando quer apenas perambular pela cidade coloca-o no bolso. Na hipermídia ocorre o mesmo processo. “Mesmo aqueles que não abrem mão da liberdade de escolha, nem do prazer da descoberta, podem ter uma navegação mais tranqüila se apoiados por vários recursos de orientação, como mapas, trilhas, placas, marcos, guias ou marcadores de páginas.” [Tori, 1996:28]

Thüring, Hannermann & Haake [1995:59] afirmam que não basta criar uma estrutura simplificada, é necessário também transmitir a estrutura ao usuário. E os mapas cumprem bem esta função, pois oferecem uma visão geral do documento hipermidiático. Há ainda outra vantagem: ele já é um modelo conhecido de orientação de espaços complexos. De

⁵⁵ T.A.: “Ela ocorre quando os leitores não sabem onde estão, como chegar em algum lugar ou para onde deveria ir depois”

modo geral, as pessoas já estão habituadas com o seu uso, e isto diminui ainda mais a sobrecarga cognitiva, pois implica num esforço a menos para o usuário. Além disso, pode ser perfeitamente adaptado na hipermídia, devido ao caráter espacial da interface dos computadores atuais.

O fato de as interfaces computacionais serem dinâmicas acrescenta aos mapas hipermidiáticos algumas vantagens em relação ao seu modelo clássico. Uma delas é que o mapa de um sistema hipermídia pode ser apresentado numa configuração tridimensional, pois embora a interface seja espacial, é possível simular um terceira dimensão no computador. O usuário poderia através de um clique girar o mapa para visualizar outros pontos não adjacentes do espaço definido pelo eixo (x,y). Este é um aspecto bastante interessante uma vez que a estrutura de um sistema hipermídia não é necessariamente colocada numa concepção bidimensional. Outra idiosincrasia dos mapas de computador é que eles podem ser interagidos, ao contrário do modelo em papel. Assim, além de orientar, o mapa pode também fornecer a navegação, porque cada elemento funcionaria como um ponto-quente ou *link*. O próprio aspecto do mapa pode ser alterado visualmente em função do estado de navegação do usuário, algo que é já é feito de modo semelhante quando um usuário clica uma palavra-chave da Internet e esta muda a sua coloração. Esta característica pode obviamente ser estendida de modo muito mais abrangente e complexo num mapa hipermídia. Finalmente o mapa pode agregar elementos multimídia, como áudio ou animações.

Por outro lado, como já afirmamos acima, os mapas de sistemas hipermídia de uma certa forma subtraem substancialmente o charme existente da navegação livre. Mas talvez devêssemos enxergar o mapa apenas como um porto seguro, não como um inibidor da tal liberdade de navegação. Como afirma Tori, “os projetistas de sistemas de hipermídia devem gerenciar adequadamente o conflito entre redução da carga cognitiva e maximização da liberdade de decisão do usuário.” [Tori, 1996: 30] O mapa poderia resolver este conflito ao ser colocado simplesmente como uma opção e não como uma necessidade de navegação.

3.3.2 As metáforas de orientação

“Uma das formas de se reduzir a sobrecarga cognitiva e em sistemas de hipermídia é através de estruturas hierárquicas, que por sua vez embutem decisões do autor em detrimento da liberdade do usuário.” [idem, ibidem]. Se estas estruturas forem baseadas em uma metáfora espacial, ainda melhor. Como afirma Mountford *‘Metaphors have two distinct but related uses in interface design: as cognitive aids to users, and as aids to creativity for designers. Metaphors can help designers to use their own, often unconscious, expectations to create new information links and mental structures.’*⁵⁶ [Laurel, 1992:26]

De mais a mais, modelar a estrutura a partir de uma metáfora é uma maneira mais sutil de orientar usuários na navegação de sistemas não-lineares do que usar mapas. Ao contrário destes, as metáforas não mostram de modo tão explícito os caminhos e encruzilhadas que o interagente poderia estar percorrendo em um trabalho de caráter hipermídia. Na verdade, elas disponibilizam um ambiente virtual que os usuários conhecem a partir de um modelo já existente no mundo real, em geral hierarquizado ou sem grandes ambigüidades. Por exemplo, a série de CD-ROMs *“The Great Museums of Europe”* apresenta o acervo de vários museus, como o Louvre, o Prado, o do Vaticano numa estrutura visual de navegação similar ao que existe nos corredores dos próprios museus. Se o usuário estiver habituado a visitá-los, ou, na melhor das hipóteses, conhecer pessoalmente um deles, vai se sentir muito mais confortável no seu similar virtual. Além de o usuário não se perturbar, já que a sobrecarga operacional se reduzirá bastante, as metáforas tornam o ambiente de navegação mais lúdico, e o usuário poderá se concentrar mais naquilo que lhe interessa: o conteúdo.

[No CD-ROM *The Great Museums of Europe*, o usuário navega por uma interface que representa o próprio ambiente do museu]

⁵⁶ T.A.: “As metáforas têm dois distintos, mas relacionados, usos no projeto de interface: ajuda na orientação ao usuário e auxilia no processo criativo do projetista. As metáforas podem ajudar o projetista a usar suas próprias expectativas, freqüentemente inconscientes, para criar novos *links* de informação e estruturas mentais.”

Outro aspecto vantajoso do uso de metáfora no apoio à navegação é que elas acrescentam maior coerência global no sistema. Por se tratar de uma estrutura baseada em um modelo, os elementos visuais componentes das telas do sistema estão todos, de algum modo, amarrados. O usuário sente que há um contexto no hiperdocumento; e o contexto facilita a percepção de um modelo unificado na mente dele. *“The preservation of context conveys a sense of continuity across nodes that is very important for comprehension.”*⁵⁷ [Thüring, 1995:58]

É interessante acrescentar também que as metáforas podem coexistir aos mapas. *“While orientation facilities are meant to help readers find their way, navigation enable readers to actually make their way.”*⁵⁸ [idem, 60] O mapa, que estaria mais para um auxílio à orientação, ajuda o usuário a encontrar o seu caminho; enquanto a metáfora, uma espécie de auxílio à navegação, ajuda o leitor a fazer o seu percurso. No entanto, o exagero de facilidades também pode confundir o leitor. Criar muitos mecanismos de ajuda simultâneos, como mapas, menus, metáforas, atalhos etc criam uma sobrecarga de conhecimentos operacionais que nem sempre o leitor pode apreender e podem eliminar a coerência global da estrutura.

3.3.3 Desorientação na Web

No que se refere à estrutura, a Internet deve ser considerada um caso à parte. A WWW não possui uma organização hierarquizada e historicamente tem estado em expansão. *Sites*⁵⁹ são criados e eliminados facilmente sem que a estrutura da rede seja abalada. Num sistema tão dinâmico, fica impraticável disponibilizar um mapa geral de toda a rede mundial.

⁵⁷ T.A.: “A preservação de um contexto transmite uma sensação de continuidade entre os nós que é muito importante para a compreensão.”

⁵⁸ T.A.: “Enquanto as facilidades de orientação pretendem ajudar os leitores a encontrar o seu caminho, as facilidades de navegação na verdade capacitam os leitores a fazer o seu caminho.”

⁵⁹ Denomina-se *site* um grupo de páginas WWW relacionadas que em conjunto formam um hiperdocumento individualizado. *Site* em inglês quer dizer lugar ou sítio.

Não bastasse isto, outras características da WWW ainda auxiliam para tornar o usuário ainda mais desorientado. Por não ser obra de apenas uma equipe, ela não possui uma unidade de composição gráfica. O conteúdo é bastante eclético, porque a Internet serve às mais diversas necessidades: editoriais, comerciais, de serviço, artísticas, de pesquisa etc. A *Web* não tem a mesma coerência formal e de conteúdo que têm os CD-ROMs multimídia, por exemplo⁶⁰.

Felizmente a WWW apresenta alguns padrões. Uma delas é que para visualizar páginas WWW todo usuário precisa de um *browser* (ou navegador), o software que tem capacidade para interpretar os códigos HTML. E apesar da batalha comercial existente entre as empresas que desenvolvem este tipo de aplicativo⁶¹, há uma tradição operacional comum a todos eles. Só este fato já ajudaria a diminuir a sobrecarga cognitiva do usuário. Mas além disso, há um conjunto de ferramentas presentes nos *browsers* que substituem o mapa e fornecem uma certa orientação no ciberespaço. Pode-se atribuir a existência de um padrão entre os navegadores ao fato de todos eles terem sido modelados a partir de uma mesma metáfora: a de visualizador de páginas.

3.3.4 A metáfora do *browser*

Browse, em inglês, tem o significado figurado de folhear ou passar os olhos sobre as páginas de um livro. No caso da Internet, o livro é a WWW. O *browser*, portanto, é o folheador ou visualizador das páginas WWW. Algumas das ferramentas deste aplicativo refletem este conceito

⁶⁰ Embora adoção da metáfora de página na Web e a utilização por parte dos projetistas das mesmas linguagens de desenvolvimento (como HTML, Flash etc) tenham padronizado a produção da informação e estandardizado muitos dos aspectos visuais dos *sites*.

⁶¹ O primeiro *browser* comercial foi o Mosaic da NCSA (*National Center for Supercomputing Applications*) da Universidade de Illinois. O Mosaic reinou absoluto nos primeiros anos da WWW, quando a Internet ainda era um reduto de pesquisadores. Mas foi o primeiro *browser* gráfico em cores, e isto foi um dos fatores que permitiram a posterior popularização da *Web*. A partir de 1995-6, a Netscape Communication começou a tomar conta do mercado com o seu *browser* Netscape Navigator, chegando a dominá-lo quase completamente nos anos seguintes, a ponto de seu nome (*navigator*) tornar-se um sinônimo de *browser*. Quando mais de 80% dos usuários já navegavam pelo Netscape, a Microsoft resolveu entrar fortemente no setor de navegadores e criou o seu próprio *browser*, o Internet Explorer. Na entrada da década seguinte a versão da Microsoft já praticamente controlava o mercado. Nestes anos, Microsoft e Netscape travaram uma batalha sem precedentes pelo controle dos navegadores Web. A cada lançamento as duas empresas introduziam novos recursos, que se

Um deles é o *bookmark*, um recurso dos *browsers* que permite ao usuário marcar uma página WWW de seu agrado. É uma metáfora para os marcadores de páginas de livros. Se os consideramos tão úteis para a nossa leitura convencional, o que dizer então no caso da WWW, um espécie de “livro-labirinto” cujas páginas não estão ordenadas de modo linear e

cujos endereços URL precisam ser digitados cada vez que queremos vê-las novamente.

Outro recurso interessante é o botão *Back* (Voltar) que permite ao usuário ir retornando às páginas por qual passou. Talvez seja o mecanismo de orientação mais útil na Internet. Se o usuário quiser voltar para a página imediatamente anterior, ele dá um clique neste botão. Se quiser voltar duas páginas antes, ele dá dois cliques; e assim por diante. Com este botão, mesmo que o usuário se perca, ele pode ir retornando até um local que considere uma espécie de porto seguro. Lucia Leão [Leão, 1999:100] visualiza no botão *Back* uma analogia bastante adequada com as pedrinhas deixadas pelo Pequeno Polegar, para saber como retornar da floresta, ou das migalhas de pão que João jogou no chão, para mostrar o caminho de volta no conto de fadas “João e Maria”. O *Back* acumula um histórico das páginas WWW pelo qual o usuário passou e tem uma analogia precisa com um trilha. As trilhas são um dos mecanismos mais naturais de orientação em espaços complexos como florestas, parques nacionais e desertos. Atualmente apresenta também uma conotação de aventura relacionada às caminhadas de viajantes audaciosos. Isto é interessante porque demonstra mais uma vez o quanto os aventureiros têm de comum com o público da própria *Web*. Ambos gostam de navegar, se deixar levar, mas também sabem como encontrar os seus atalhos.

Outro recurso WWW importante é o botão “*Go To*” (“*Ir*”). Vimos acima que cada página WWW possui um endereço particular, chamado de URL, que permite ao usuário carregá-la no *browser*. O navegante digita o endereço no campo correspondente e clica o botão “*Ir*”.

por um lado servia para tornar cada vez mais amigável o ambiente WWW, por outro lado colocou um fim na padronização que era uma intenção inicial da Internet.

Portanto, se num livro cada página tem uma numeração, na WWW cada página tem um endereço. Mas talvez deveríamos amplificar este conceito de modo a enxergar a *Web* não somente como um livro, mas sim como uma biblioteca⁶². Do mesmo modo que cada livro de uma biblioteca convencional tem o seu código, na Internet, cada página WWW tem um endereço URL.

A WWW colocada como a metáfora da biblioteca vai ao encontro de um dos desejos mais fortes presente no imaginário de muitos escritores, intelectuais e pesquisadores: a existência de uma biblioteca que contenha todos os livros do mundo. A Internet poderia ser a solução técnica desde sonho, se todos os documentos criados pela humanidade fossem digitalizados⁶³.

António Fidalgo lista uma série de desvantagens existentes nas bibliotecas convencionais que não encontraríamos nas correspondentes virtuais, “desde uma má organização do catálogo e dificuldades de empréstimo de livros, a reduzidos horários de abertura e à má vontade dos bibliotecários” [Fidalgo, 1999:282]. Já na biblioteca digital, “leva -se o livro da estante e ele permanece lá, pronto a ser levado por outro, e assim sucessivamente. E quantas mais vezes ele for levado, mais aumenta a segurança de que esse livro ficará penção (sic) da biblioteca universal” [idem, 286]. O autor chega a fazer uma brincadeira ao dizer que a Internet seria, na verdade, uma espécie de “fotocopiadora universal, a uma velocidade estonteante, de textos, imagens e sons.” [idem, ibidem]

Outra vantagem da biblioteca virtual é que ela é, pelo menos teoricamente, mais democrática⁶⁴, pois qualquer pessoa pode disponibilizar um documento de sua autoria no enorme banco de dados da Internet. Não é de se estranhar a quantidade de informações que já foi colocada na rede mundial em tão pouco tempo de existência. Mas isto gera algumas

⁶² Uma das comparações mais recorrentes é justamente colocar a Internet como uma grande biblioteca mundial.

⁶³ Digitalizar é o processo de tradução e codificação de uma informação qualquer (som, imagem, texto, vídeo etc), representada através de meios mais convencionais, geralmente de caráter analógico, em sinais que contenham somente os dois estados típicos dos sistemas computacionais, o zero e o um.

⁶⁴ No entanto, a Internet ainda é reduto de uma minoria, se considerarmos o mundo todo. Apesar da euforia, há aqueles que visualizem a Internet como mais um fator de exclusão social e desconcentração de renda. De mais a mais, a criação de um *site* não é tão simples quanto o ato de telefonar, fotografar ou gravar um vídeo. Além disso, a maioria esmagadora dos *sites* é muito pouco ou nada consultada.

contra-indicações, pois volume de dados nem sempre é sinal de qualidade. Uma delas é discernir entre o que são boas ou más fontes. Não há uma “bibliotecária” que gerencie a entrada dos dados para verificar quais documentos são realmente fidedignos. A WWW é uma rede cuja organização (inclusive do conteúdo) não é rigidamente regulada, nem formalizada por uma instituição centralizadora. Portanto, outro problema é saber como e onde procurar uma informação. A Internet põe à disposição do usuário uma biblioteca tão gigantesca, mas tão pouco regulamentada e arranjada, que um leitor pode se perder facilmente na própria pesquisa.

Por isso, atualmente existem um série de *sites* cujo objetivo é ajudar o leitor a buscar uma informação na porção WWW da Internet. São chamados de *sites* de busca. Cada *site* de busca possui um cadastro de diversos outros *sites*. Um banco de dados guarda, de cada *site*, algumas informações consideradas significativas, como o seu endereço URL, uma palavra-chave, o título da apresentação etc. Qualquer pessoa, empresa, órgão público etc que queira se divulgar pode ter o seu *site* cadastrado. Quando um usuário entra no *site* de busca, ele digita uma palavra-chave para que um algoritmo investigue no banco de dados se algum dos *sites* que foram previamente cadastrados tem esta informação. O *site* de busca corresponde, de certo modo, aos arquivos de aço que são consultados pelo visitante de uma biblioteca convencional, à procura de um documento. Infelizmente, nos dias de hoje, não há um *site* de busca que contenha um cadastro de todos os *sites* e informações da Internet. Principalmente porque ainda ninguém criou um algoritmo capaz de realizar uma busca realmente universal; uma busca que independa da existência do cadastro em um banco de dados. Seria o correspondente a um arquivo de aço que contivesse o conteúdo de todas as bibliotecas do mundo. Enquanto isto não ocorrer, a WWW vai se parecer mais como um conjunto de bibliotecas interligadas do que uma biblioteca de fato universal.⁶⁵

⁶⁵ Existem alguns *sites* de busca que aglutinam a maioria dos outros *sites* de busca convencionais. São chamados de *sites* de metabusca. Na verdade, quando o usuário digita uma palavra-chave o *site* de metabusca efetua a procura no banco de dados dos *sites* de busca. Embora eles tenham a pretensão de se tornar um *site* de busca realmente universal, ainda há a dependência do cadastro prévio por parte de um projetista de páginas num dos *sites* de busca. Ou seja, muitos *sites Web* simplesmente não aparecem na lista por não terem se cadastrado. Por outro lado, alguns *sites* de busca não trabalham com cadastros, mas com varreduras em servidores. Mas estes também não conseguem abranger toda a comunidade *Web*.

3.4 O mundo interligado: da metáfora do tampo de escrivaninha para a do navegador

Cada dia que passa os computadores adquirem maior capacidade de interconexão. No início, a Internet só se tornou popular porque se aproveitou de uma infra-estrutura de conexão já instalada: as linhas telefônicas. No entanto, esta tecnologia de comunicação não conseguia acompanhar a velocidade de processamento oferecida pelos computadores pessoais, e isto gerava um gargalo na atualização das informações. Hoje é possível estabelecer conexões com outras tecnologias de comunicação bem mais velozes: por terra, via fibra óptica ou cabo; por ar, via satélite, radiotelefonia ou ondas eletromagnéticas e mesmo por linhas telefônicas mais rápidas. Tudo indica que ainda possam surgir novas possibilidades de conexão rápida. Caracteriza-se este conjunto como tecnologias de banda larga. Por enquanto não se enxerga um limite para este avanço tecnológico, mas para o homem se satisfazer, basta ele ter a sensação de que a comunicação se efetua em tempo real.

Costuma-se dizer portanto que a Internet vai alterar a sensação de distância, pelo menos como a percebemos hoje. Já não faz muita diferença entre buscar um dado que está num computador instalado na Austrália, no Japão ou no seu vizinho. Com a digitalização dos próprios objetos de informação, como o livro, o CD de música ou a “fita” de vídeo, as compras também poderão ser feitas indiferentemente em qualquer ponto do mundo. Na verdade, com a porção WWW da Internet estamos invertendo o processo de locomoção. Em vez de irmos para os lugares, estamos colocando eles dentro de casa: uma loja, um órgão público, um supermercado, todo o espaço que possa ser representado e virtualizado. Não é à toa, portanto, que a representação de um hiperdocumento na *Web* seja na verdade chamado de *site*.

Site em inglês quer dizer lugar ou sítio. Interessante a utilização desta metáfora, pois ela teria muito mais a ver com a sensação tridimensional presente na realidade virtual⁶⁶. Mas parece que, de algum modo, houve a percepção de que a Internet poderia disponibilizar muito mais do que simplesmente documentos. Por outro lado, a metáfora do *site* está muito

⁶⁶ Realidade Virtual é a denominação comumente definida para descrever o universo de pesquisas e aparatos tecnológicos realizados no sentido de virtualizar ambientes construídos por algoritmos de computador e que

mais relacionada à idéia do *browser* como um navegador, do que como um “folheador” de páginas. Ou seja, o usuário estaria, com este aplicativo, navegando por lugares virtuais e não só por páginas. Coincidência ou não, o *browser* é cada vez mais chamado de “navegador”. Esta mudança de paradigma aproxima a porção WWW da Internet da realidade virtual. As páginas de um *site* corresponderiam aos diversos ambientes de um mundo virtual e os *links* seriam as “portas” para os outros lugares. Atualmente, alguns *sites* chegam a se autodenominar “portais”, porque se consideram a entrada para o conhecimento de algum assunto específico. Existem os portais de saúde, de esportes, de lazer etc. A lógica da metáfora é seguinte: entrando no site *x*, o usuário vai encontrar tudo que quiser saber sobre esportes, por exemplo. Lá dentro ele vai também ter *links* para outros *sites* que falam sobre o mesmo assunto. Mas obviamente os portais vão fazer de tudo para você não sair dele, porque a maioria é sustentada por muita publicidade. Alguns chegam a criar algoritmos que impeçam que o usuário vá muito longe, fazendo-o retornar sempre que possível ao seu *site*. Isto é uma pena, porque começa a desvirtuar a idéia fundamental da Internet, que é a de interligar computadores mundialmente.

De qualquer modo, nesta nova conjuntura, em que as máquinas estão em uma rede mundial interligada, os computadores deixam de estar isolados, praticamente perdem a característica de ser PCs⁶⁷ para se tornarem IPs (computadores interligados). Caminha-se rapidamente para uma mudança de paradigma metafórico: o do tampo de escrivaninha para o de navegador. Neste sentido, os computadores não são mais somente ferramentas corporativas, são também e cada vez mais meios de comunicação. Como afirma Theo Mandel:

*“Most PC operating system and application interfaces today follow the common office desktop metaphor. The Internet breaks away from the desktop metaphor where users keep things in all-purpose or specific containers, such as folders, cabinets, and workspaces. Instead, the Internet follows the metaphor of viewing and navigating through a browser. This type of interface allows users to see onto the limitless resources of the global network.”*⁶⁸ [Mandel, 1997:388]

quando atualizados colocam o receptor imerso num mundo imaginário, embora ele ache que esteja interagindo no real. Leia mais sobre realidade virtual adiante neste capítulo

⁶⁷ Abreviação para “Personal Computer” em inglês, que significa computadores pessoais.

⁶⁸ T.A.: “A maioria dos sistemas operacionais e interfaces de aplicativos dos PCs de hoje seguem e em comum a metáfora do tampo de escrivaninha. A Internet liberta-se da metáfora do tampo de escrivaninha, em que usuários guardam coisas em todo de recipientes específicos, como pasta, gabinetes e espaços de trabalho. Em vez disto, a Internet segue a metáfora da visualização ou navegação pelo *browser*. Este tipo de interface permite ao usuário enxergar pelas ilimitados recursos da rede global.”

Alguns dos novos sistemas operacionais de computadores começam a prestar a atenção nesta realidade. Ao ligar o microcomputador, o usuário entraria diretamente no *browser*. É como se este aplicativo fosse uma janela para o mundo. Todas as operações realizadas na máquina seriam ditadas conforme a metáfora do navegador, como já tinham sido feitas pela do *desktop*. Alguns estudiosos do assunto acreditam mesmo que não vai mais haver a necessidade de ter programas de computador gravados nos discos rígidos das máquinas. Caso o usuário queira digitar uma carta, ele poderá carregar um processador de textos direto da Internet. Ele usaria o programa, salvaria ou imprimiria o documento e depois descarregaria o *software* de sua memória. Para isto pagaria uma taxa de locação. A Internet, além de ser um repositório de documentos e objetos digitais, também estaria disponibilizando ferramentas virtuais.

Costuma-se dizer inclusive que muitas tecnologias, como telefone, televisão, computador, agenda eletrônica, rádio, videogames, jornal, videocassete etc serão fundidos numa única tecnologia. Há sinais de que isto já está ocorrendo. Não é de hoje que o computador substitui ou se funde ao jornal ou ao videocassete, que incorpora o telefone ou a agenda eletrônica; não é de hoje também que a televisão, por outro lado, tenta ser interativa como o computador etc. É como se o jornal eletrônico mostrado no computador fosse uma metáfora do seu correspondente em papel. Ou os softwares que reproduzem vídeos ou áudios digitais fossem metáforas do videocassete e do tocador de CDs. Aliás, a incorporação de um meio por outro, ou de uma manifestação por outra, não é tão nova assim na nossa história. No início, o cinema parecia teatro filmado e a televisão tinha todos os cacoetes de um programa radiofônico. Em manifestações que usam o computador como meio aconteceu um processo semelhante.

3.5 Metáforas de outros meios

Como diz Alan Kay, o computador “*is a medium that can dynamically simulate the details of any other medium, including medium that cannot exist physically*”⁶⁹ [Laurel, 1992:224], pois a matéria-prima do computador é a linguagem. *‘Logic and languages, symbols and*

semantic systems, are the thinking tools for making thinking tools that shifted human evolution into overdrive.”⁷⁰ [idem, 225] Através do *software*, é possível representar outros meios no computador. “O computador, como notou Bret, seria algo assim como uma meta-ferramenta, uma ferramenta (*hardware*) com a qual se pode ‘manufaturar’ outras ferramentas (*software*)...” [Machado, 1993:39]

Diversos aplicativos no computador cumprem esta função e chegam a ser quase representações perfeitas dos aparelhos convencionais. Suas interfaces são, assim, representações de máquinas virtuais, e, portanto, são metáforas de máquinas reais. Por exemplo, programas como o *RealPlayer*, o *QuickTime* ou o *Media Player* foram criados para reproduzir seqüências de vídeo e áudio. São chamados genericamente de “tocadores de mídia”. Estes programas têm como objetivo dialogar com a parcela multimídia dos sistemas operacionais. Mas a representação é simbolizada por metáforas de videocassetes ou tocadores de CDs de música. Possuem ícones que apresentam o aspecto visual dos botões deste aparelhos: botão Tocar (*Play*), Parar (*Stop*), Pausa, Avançar (*Forward*), Rebobinar (*Rewind*) etc. Nas versões mais recentes destes aplicativos é possível selecionar estações de rádio virtuais como se o usuário estivesse girando o botão de sintonia dos rádios convencionais.

A despeito da facilidade de interação e compreensão oferecida ao usuário, devido ao caráter quase 1:1 da representação do aparelho na interface, a aplicação de uma metáfora ao extremo da simulação completa corre o risco de desprezar algumas adaptações que poderiam ser colocadas em função das especificidades fornecidas pelo computador. Por

⁶⁹ T.A.: “É um meio que pode dinamicamente simular os detalhes de qualquer outro meio, incluindo os meios que não podem existir fisicamente.”

exemplo, há uma série de *sites* na Internet cujo objetivo é funcionar como uma estação de rádio. Possuem controles convencionais, como botão para ligar, para parar uma música, para aumentar ou diminuir o volume, painéis do tipo LCD etc. Mas no seu modo de funcionamento e interação, praticamente nada de muito significativo o diferencia de uma estação de rádio normal. Seria interessante que o usuário pudesse ao menos escolher a sua seleção musical, porque o computador, como meio digital, permite este tipo de experiência inovadora. A apresentação de notícias pela WWW entra neste mesmo conjunto.

Na WWW podemos encontrar diversos jornais eletrônicos, em geral chamados de “jornais *online*”. André Manta, da UFBA, diz que a utilização desta expressão deve ser entendida como uma metáfora, pois “a noção de jornal sempre esteve associada a um suporte material (o papel)”, enquanto que “no ambiente das redes comunicacionais as informações viajam a velocidades inimagináveis na forma de impulsos eletrônicos que podem ser atualizados de forma imediata.” [Manta, 2000:cap03.htm]. Ou seja, usamos a expressão jornal *online* na falta de um nome próprio e porque é mais confortável criar uma denominação a partir de algo que já conhecemos bem. Mas o jornal digital é praticamente um novo gênero noticioso. E deve se aproveitar disso, utilizando recursos multimídia e atualizando as informações à velocidade das conexões. A utilização de uma metáfora como suporte, porém, tem um aspecto positivo. “É muito mais fácil para o usuário interagir com um produto que lhe pareça familiar de alguma forma. Não é à toa que a maioria dos serviços de notícias na rede siga uma lógica parecida com a das publicações impressas.” [idem, ibidem] Nada mais natural, portanto, do que jornais digitais usarem uma diagramação similar a que se vê no jornal convencional. O que acontece, de modo geral, é que os próprios desenhistas gráficos do meio impresso são chamados a desenhar o seu similar eletrônico. A transposição é portanto automática.

Neste mesma linha, é possível encontrar títulos em CD-ROM com “cara” de projetor de *slides* ou de filmes, tal o caráter linear do projeto ou páginas WWW cuja interface é apresentada dentro da moldura de um livro. Nela o usuário simplesmente clica em um botão para virar as páginas. Há ainda títulos em CD-ROM ou páginas Web que usam a

⁷⁰ T.A.: “A lógica e as linguagens, os símbolos os sistemas semânticos, são as ferramentas do pensamento para construir ferramentas do pensamento que colocam a evolução humana em quinta marcha.”

metáfora de livros infantis, dos *cartoons* ou das histórias em quadrinhos. O ambiente é lúdico, colorido e animado. Os exemplos não param por aqui.

Há casos, no entanto, em que o computador não aparece de modo tão evidente somente como suporte para meios tradicionais. A hipermídia pode ser considerada um gênero em si. Há diversas manifestações de caráter hipermídia que não se enquadram em nenhum modelo conhecido. Mas apesar das características diferenciadas da hipermídia, que a tornam um fenômeno comunicativo especial e um formato inovador, em muitos casos ela ainda insiste em herdar a experiência de outros meios como auxílio na criação de suas peças. Isto é possível porque na interface gráfica do computador é possível construir estruturas específicas de outros meios. “A analogia entre suportes pode ser feita a partir de uma aproximação estrutural, isto é, a partir das analogias e dos modos de exibição.” [Plaza, 1986:36].

Não é preciso condenar este tipo de experiência, porque ela já ocorreu com o surgimento de outros fenômenos de caráter artístico ou comunicativo. Já citamos o fato de que nos primórdios do cinema a câmera mantinha-se fixa, e o filme parecia, aos olhos de hoje, muito mais um teatro filmado do que uma peça cinematográfica. Isto é bastante normal na germinação de um novo formato e, na verdade, até bastante saudável no processo de desenvolvimento de sua própria linguagem. De mais a mais, com o tempo, o novo gênero vai adquirindo naturalmente as suas especificidades, mesmo porque algumas apropriações tornam-se obsoletas ou porque percebe-se que não são adequadas. Plaza acrescenta ainda que:

“se partirmos da premissa de que todo meio inclui os anteriores como conteúdo, impõe-se uma forma de organização eminentemente tradutora das linguagens. Em segundo lugar, toda mensagem, como organização física de sinais, deve adequar-se às condições impostas pelo suporte, procurando isoformia na adequação ao meio.” [idem, 37]

Na hipermídia, a apropriação de outros meios pode ser considerada algo ainda mais natural, pois, no seu caso especial, ela possui a característica técnica e conceitual de agregar dentro de si todos os outros meios. É comum que um deles acabe funcionando como suporte metafórico para a obra. Além do mais, por ser um formato novo, é um pouco difícil estabelecer quais são as especificidades da hipermídia. Estaríamos mesmo amputando toda

uma liberdade de criação que existe hoje, se disséssemos “como” deve ser feito. No atual estágio, para os autores de hipermídia, uma estratégia é se apropriar da linguagem de outros suportes, como metáfora, para se fazer entender. É uma questão de alfabetização do usuário. Como afirma Plaza, “a aprendizagem de um novo meio sempre se faz pela referência e comparação a outros já conhecidos.” [idem, 34]

Por outro lado, o movimento de apropriação de metáforas a partir de outros meios ou gêneros começa a adquirir duas mãos. A TV, o jornal, o cinema etc também começam a sofrer influência da hipermídia, num caminho inverso. A interface destes meios começam a se apropriar de algumas características que foram se formando na hipermídia: simulação de efeitos de cliques em eventos televisivos, o desenho das telas de TV diretamente inspiradas em *sites*, páginas de jornais diagramadas de modo a criar articulações hipertextuais (linhas ou fios que puxam uma palavra para uma foto, por exemplo), filmes que tentam desvincular-se de orientações lineares⁷¹ etc. Parece que já estamos tendo uma amostra do que será o futuro quando todos os meios estarão conjugados – televisão, computador, telefone etc.

Na atualidade, em que os suportes ainda estão claramente individualizados, o computador leva vantagem. Não é um aparelho fechado como uma televisão, um videocassete ou um aparelho de som. Pois se a televisão transmite programas de TV, se o videocassete reproduz fitas de vídeo, se o aparelho de som toca CDs de música, hoje o computador pode fazer tudo isto e muito mais. É um aparelho multifuncional, flexível e versátil; ao mesmo tempo meio de produção e de veiculação. Ao ligar o aparelho-computador você poderá operar dentro dele outros aparelhos. O computador é uma máquina que simula outras tantas. É uma mesa de trabalho (*desk*) que possui máquina de escrever, calculadora, prancheta de desenhista, mesa de cartado, ilha de edição, toca-CD, videocassete, rádio, telefone, fax, televisão, retroprojetor (se conectado a um *data-show*) e ainda juntou tudo isto para criar o novo formato da hipermídia

⁷¹ Exemplos típicos: o filme alemão “Corra Lola Corra” (Tom Tykwer, 1998) ou o brasileiro “Amores Possíveis” (Sandra Werneck, 2000). Em ambos os diretores criam três versões possíveis numa mesma narrativa. É como se o espectador pudesse escolher qual é o melhor final a partir das três opções assistidas.

Sob este olhar, é injusto dizer que o computador é uma máquina difícil de operar. Na realidade é muito simples. Basta apertar o botão Ligar (*Power*). Difícil mesmo é operar os diversos aparelhos que são simulados dentro dele. Cada um destes aparelhos virtuais têm o seu *modus operandi* particular. Portanto, nada melhor do que criar metáforas destes aparelhos em programas que são analogias daquilo que conhecemos no nosso mundo real. O exemplo dado pelos programas tocadores de mídia (*RealPlayer*, *QuickTime*, *Media Player*) são bastante ilustrativos. Por simular os controles de um aparelho de som ou de um videocassete, eles são fáceis de usar. Suas interfaces não são percebidas como interfaces, portanto são naturais para o usuário. Ele não precisa entender como usá-las. São intuitivas. Portanto, se o projetista quiser que a sua ferramenta seja compreendida de imediato que construa esta ferramenta segundo o que certos autores chamam de “princípio da transparência”.

3.5.1 Interfaces Transparentes

“... Colocar ênfase na interface é um erro. Uma interface é um obstáculo: ela está localizada entre a pessoa e o sistema usado pelo usuário”. Em seu artigo intitulado “*Why Interfaces Don’t Work*” [Laurel, 1992:120] Donald A. Norman, diretor *do Institute of Cognitive Science da University of California*, quis afirmar que ao desenvolver uma interface, os projetistas deveriam, segundo suas palavras, privilegiar as necessidades do usuário e sua tarefa. Nunca a interface em si, nunca o aspecto gráfico dos elementos interfaciais.

“The real problem with the interface is that it is an interface. Interfaces get in the way. I don’t want to focus my energies on an interface. I want to focus on the job. My tool should be just something that aids, something that does not get in the way, and above all, something that does not attract attention and energy to itself.” [idem, ibidem]⁷²

Norman queria enfatizar que as tarefas deveriam ser o centro das atenções, não as ferramentas. Para o autor o foco estava sendo colocado no computador em si, em detrimento dos usuários. Era necessário auxiliar a tarefa, não a interface da tarefa.

⁷² T.A.: “O verdadeiro problema da interfaces é que ela é uma interface. A interfaces ficam no caminho. Eu não quero cocentrar minhas energias na interface. Eu quero me concentrar na tarefa. Minhas ferramentas deveriam ser somente algo que me auxilie, alguma coisa que não fique no meu caminho, e sobretudo, algo que não atraia a atenção e a energia para ela mesma.”

Muita coisa aconteceu desde a época em que Norman publicou o seu artigo, no início da década de 90, mas muitos usuários ainda reclama de uma certa dificuldade de compreensão operacional do computador, o que explica a existência de um número tão grande de cursos de computação. Estes cursos não pretendem tornar ninguém um *expert* em *hardware* ou um exímio programador. A intenção é menos ousada: é a de ensinar como operar a máquina – ou, em outras palavras, como interagir com a interface desta máquina.

Um mundo ideal faria do computador um aparelho tão fácil de ser operado quanto uma televisão, uma geladeira ou um aparelho de som. Donald A. Norman afirma que o computador deveria tornar-se como um jogo da Nintendo: o usuário liga e joga [idem, 216]. Ou seja, no futuro, a interface do computador deveria ser invisível. Um outro pesquisador, Ben Shneiderman, da Universidade de Maryland, já dizia mais ou menos o mesmo em tempos mais remotos: “*when an interactive system is well designed, it almost disappears, enabling the users to concentrate on their work or pleasure.*”⁷³ [Shneidermanm, 1987:9] Chris Rutkowski, no seu artigo *An Introduction to the Human Applications Standart Computer Interface*⁷⁴, apresentou um conceito similar, que chamou de ‘princípio da transparência’: o usuário deve interagir diretamente com a tarefa; as ferramentas em si têm que desaparecer.

Por trás destes conceitos, há uma constatação: as interfaces de sistemas computacionais devem sumir, a ponto do usuário não perceber a sua existência. Devem ser simples, discretas como as interfaces de outros sistemas – os jogos da Nintendo, por exemplo ou de outros aparelhos domésticos, como a televisão (devemos lembrar, no entanto, que as gerações mais antigas têm muito mais dificuldades em manejar, por exemplo, um videocassete do que as crianças, que nasceram vendo e mexendo no aparelho. Sua interface, portanto, não é tão invisível assim – a prova disso é que estes aparelhos, em geral, são acompanhados pelo seu manual de operação).

⁷³ T.A.: “Quando uma interface é bem projetada, ela sempre desaparece, fazendo o usuário se concentrar no seu trabalho ou lazer.”

⁷⁴ Chris Rutkowski. “An Introduction to the Human Applications Standart Computer Interface”, Part 1: Theory and Principles, *BYTE* (7, 11), Outubro 1982, 291-310.

É natural, portanto, que um dos modos mais simples de se obter a transparência de uma interface é a partir da utilização de uma metáfora na interface que represente algo que já é conhecido pelo usuário em todas suas dimensões: interação, navegação, caminhos, modos de seleção etc. Carl Steadman, uma das cinquenta pessoas que mais conhecem de Internet no mundo, segundo a revista Newsweek, afirmou que *“the key to a truly new media is not interactivity, but transparency, achieved through designing an interface that appears immediately obvious to its users.”*⁷⁵ [Steadman, 2000:125] Para se tornar mais óbvia ou intuitiva a interface só precisaria de mecanismos de interação que fossem mais naturais ou que ao menos tivessem algo a ver com a interface representada. Mas tudo o que o usuário tem é o *mouse*, é o teclado e mais algum outro dispositivo de entrada, que estão muito mais relacionados ao modelo clássico de computador do que o que o usuário poderia esperar duma interação com a interface. A história do computador tem demonstrado que o usuário está à mercê da máquina e não o contrário. O usuário tornou-se escravo destes dispositivos (como *mouse* e teclado), quando deveria haver algo de uso mais natural. A interface até tenta preencher este vácuo criando metáforas para estes dispositivos de entrada, mas não é suficiente para deixar o usuário completamente à vontade.

Norman diz que foi um acidente histórico que nos adaptou a certas tecnologias: *‘keyboards, simple pointing devices, and video displays are not the most appropriate ways to interact with many of the tasks we do today, but they are all the computer provides.’*⁷⁶ [Laurel, 1992:128]

Talvez uma saída seja o computador buscar soluções naquilo que podemos considerar um dos eixos na evolução das interfaces computacionais: a realidade virtual. Além de atuar de modo mais natural com os preceitos definidos pelo princípio da transparência, a realidade virtual disponibiliza outros dispositivos de interação que, de certo modo, colocam o usuário dentro da interface.

⁷⁵ T.A.: “A chave para o novo meio não é a interatividade, mas a transparência, que pode ser atingida ao se projetar uma interface que se mostra imediatamente óbvia para o usuário.”

⁷⁶ T.A.: “O teclado, os dispositivos de apontamento, como o *mouse*, e o monitor não são os caminhos mais apropriados para interagir com muitas das tarefas que fazemos diariamente.”

3.6 A Realidade Virtual

Apresentada com certo estardalhaço no fim dos anos 80, a realidade virtual⁷⁷ voltou a se recolher no discreto ambiente dos departamentos de pesquisa de universidades ou de algumas empresas interessadas. Mas suas pesquisas não estacionaram. Apenas a mídia deixou de lado algo que, no início, parecia ser um sucesso mercadológico maior.

A expressão “realidade virtual” sempre foi muito criticada por não enunciar de maneira contundente o seu significado. Muitos a criticam também por colocar numa mesma expressão um adjetivo (virtual) que é a antítese de seu substantivo (realidade). Mas realidade não contradiz necessariamente virtualidade. Como afirma Lévy: “A palavra virtual vem do latim *virtualis*, derivado por sua vez de *virtus*, força, potência. Na filosofia escolástica, é virtual o que existe em potência e não em ato. O virtual tende a atualizar-se, sem ter passado no entanto à concretização efetiva ou formal.” [Lévy, 1996:15]. Portanto “o virtual não se opõe ao real, mas sim ao atual.” [idem, 16] Ou seja, “A virtualização pode ser definida como o movimento inverso da atualização.” [idem, 17]

O que incomoda tanto na expressão “realidade virtual” pode ser o fato de que o virtual indique, segundo afirmação do próprio Lévy, o “não estar presente” [idem, 19] e a realidade está sempre presente de algum modo. Mas o virtual escapa da noção lugar-comum que temos do conceito do que é “real”, que é o “presente” no espaço-tempo. A realidade, neste caso, existe, mas está virtualmente guardada em algoritmos ou em memórias. O virtual é um real “quase presente” ou um real que está potencialmente presente. Mas o virtual tem o seu real, que se instaura nos suportes físicos de armazenamento para que possa se atualizar. Quando isto acontece, ele se manifesta na realidade lugar-comum que sentimos. O “real” do virtual existe como signo, que quando se atualiza torna-se um outro signo, o signo do real. O signo do virtual é um metáfora do signo do real. Ambos representam o mesmo objeto, mas são representados de modos diferentes. “A virtualização não é uma desrealização, mas uma mutação de identidade, um deslocamento do centro de gravidade

⁷⁷ Embora as primeiras demonstrações públicas de experiências da realidade virtual tenham sido realizadas na Europa no final da década de 80, já por volta de 1966, Ivan Sutherland começou a fazer experiências com dispositivos de imersão, como o capacete de visão, no laboratório Lincoln, do MIT. Mas o primeiro a

ontológico do objeto considerado.” [idem, 18] A semente é virtualmente uma árvore, a planta arquitetônica é a virtualização de uma casa, um pensamento é atualizado no discurso.

O virtual é, portanto, algo que está inserido em nossa sociedade muito antes do advento do computador. Mas a expressão “realidade virtual” é comumente definida para descrever o universo de pesquisas e aparatos tecnológicos realizados no sentido de virtualizar ambientes construídos por algoritmos de computador e que quando atualizados colocam o receptor imerso num mundo imaginário, embora ele ache que esteja interagindo no real. Segundo Cadoz, “o homem substitui a relação com seu mundo em evolução pela relação com uma representação desse mundo, que ele procura dotar de propriedades e atributos do meio ambiente ‘natural’ ao qual está acostumado.” [Cadoz, 1997:68]

Cadoz lembra de outras tentativas de definir esta manifestação, como “realidade artificial, mundo virtual, ambiente multissensorial interativo ou *cyberspace*” e propõe, ele mesmo, uma expressão, a meu ver, mais completa, que é a de “representação integral”, pois “o representante se manifesta à totalidade de nossos sentidos coordenados (visual, auditivo, tátil etc) e que se pode estabelecer uma interação material efetiva entre ele e nós, por meio de nossos gestos em particular.” [idem, 108]. “Representação Integral” estampa de

modo mais competente a sensação percebida pelo usuário de imersão no ambiente virtual. É o voar sem estar voando, é o sentir-se como um peixe nas profundezas do mar sem estar molhado, é o pisar na Lua sem estar nela, é o voltar para o passado ou o avançar para um futuro hipotético. Na realidade virtual, o usuário está imerso, mergulhado, afundado no ambiente representado. Ou seja, o usuário está dentro da interface. Isto é interessante porque sugere uma contradição. Se a interface é algo que está entre o meio e o receptor, como o receptor pode “estar” na interface? Talvez fosse mais adequado usar a velha metáfora de navegação, pois esta sugere que o receptor está na superfície e não dentro dela.

funcionar, ainda de forma rudimentar, ocorreu quando o pesquisador já estave na Universidade de Utah em 1970. Fonte: [Cadoz, 1997:11].

Evidentemente “imersão” é também apenas mais uma metáfora. O usuário de realidade virtual se sente imerso, mas não está imerso. Há uma ilusão de imersão, um mágica que faz com que o usuário se sinta dentro da interface, embora não esteja. A sensação de ilusão tanto maior será quanto menos perceptível for a noção de que a interface é de fato uma interface. O extremo seria uma espécie de fim da mediação. É neste ponto que os autores citados acima enfatizam a importância de tornar as interfaces transparentes. Alan Kay, um pesquisador e colaborador da Apple, cunhou o termo “*user ilusion*” para descrever este fenômeno. Ao contrário do senso comum, ele condena o uso da palavra metáfora para explicar a representação para aquilo que o usuário vê na interface. Para Kay, “a metáfora é uma pobre metáfora” [Laurel, 1992:199] para o que de fato acontece. Por outro lado, *user ilusion* tem claras conotações no mundo do teatro, do cenário e da mágica – “todos eles apontam mais fortemente a direção a ser seguida” [idem, ibidem].

De qualquer modo continua havendo “algo” no meio, a interface, e ela está entre o usuário e a representação; a diferença é que na realidade virtual o usuário não a percebe mais. Isto pode significar a solução para um problema contra o qual Donald Norman tanto clamava: *‘an interface is an obstacle: it stands between a person and the a system being used. Aha, ‘stands between’ – that is the difficulty. How can anything be optimal if it is in the away, if stands between the person and what needs to be done?’*⁷⁸ [Laurel, 1992:216] A realidade virtual tenta levar às últimas conseqüências o desejo de uma interface transparente.

Mas há ainda alguns obstáculos. O problema da representação é na verdade o menos complexo. As interfaces computacionais têm uma vocação para serem representadas por imagens esquematizadas, diagramas, ícones simplificados, e com o aumento da resolução da representação nas interfaces computacionais é possível “desenhar” também imagens complexas. Isto na maioria das vezes não afeta o grau de reconhecimento do leitor. Mas na realidade virtual, o caráter realístico da imagem é o ideal. Ela deve dar a ilusão, através de suas qualidades cromáticas, de tons e texturas, de que vemos o objeto real e não a imagem representada. “Graças a programas sofisticados, sabemos hoje fabricar por síntese imagens

⁷⁸ “Uma interface é um obstáculo: ela fica entre uma pessoa e o sistema que está sendo usado. Aha, ‘fica entre’ – esta é a dificuldade. Como algo pode ser ótimo se ele está no caminho, se ele fica entre uma pessoa e o que ela precisa fazer?” (tradução do autor)

extremamente ricas, em que os objetos evocados têm formas e volumes, cores, sombras e luzes, reflexos e texturas.” [Cadoz, 1997:15] Mas a sensação de realidade é multiplicada pelo elemento interação. Ao agirmos sobre ela, a imagem representada nos responde com a sua atualização.

‘Como acontece nos ‘objetos’ manipulados pela computação gráfica, a imagem é uma realidade fantasmática: ela está ali para todos os efeitos práticos, mas, a rigor, não passa de uma equação matemática à qual se forma plástica, através de um algoritmo de visualização. Basta mudar-lhe os parâmetros numéricos para que ela se transfigure em outra coisa e basta cancelar o algoritmo de visualização para que ela retorne ao seu estado de pura possibilidade matemática, em repouso nas memórias digitais.’ [Machado, 1993:18]

Mas um dos obstáculos é que a realidade virtual ainda não conseguiu construir de modo completamente eficiente todo o aparato tecnológico necessário para que o usuário se sinta, por interação, de fato imerso na representação. Nem estamos falando do *mouse* ou do teclado, tão criticados por Norman. [Laurel, 1992:128] Estes não conseguem captar com realismo necessário todo o gestual que nos dê a sensação de imersão. Os dispositivos de entrada do computador são vistos e são percebidos como ferramentas. Mas para que haja a convicção de uma realidade, as ferramentas também não deveriam ser percebidas; elas têm que ser quase transparentes. O importante, na realidade virtual, é priorizar aquilo que se “faz” na interface, não a interface em si, não suas ferramentas : *‘I emphasized the fact that it is the tasks that are central, not the tools.’*⁷⁹ [idem, ibidem].

O alfabetismo de um software se faz muito mais simples se os procedimentos se espelharem nas ações de nosso dia-a-dia. Esta tentativa já vem sendo feita há algum tempo, quando o computador, só para ficar num exemplo, introduziu eventos como arrastar (o chamado *drag-and-drop*⁸⁰), que é muito mais natural para algumas ações do que o clique do *mouse*. Por exemplo, na maioria dos sistemas operacionais de hoje, quando queremos apagar um arquivo, devemos arrastar o seu ícone até a lixeira. Isto é

⁷⁹ T.A.: ‘Eu enfatizei o fato de que são as tarefas que são centrais, não as ferramentas.’

mais realista do que seria se fizessemos o mesmo clicando no ícone do arquivo e depois clicando no ícone da lixeira. Na realidade virtual isto pode ir ainda mais longe. Ao virar nossa cabeça, o ambiente gira de modo correspondente. Os dispositivos de entrada “enxergam” o nosso movimento e um algoritmo traduz naquilo que vemos na representação. “A imagem e o sujeito têm agora a capacidade de interagir – ou de dialogar – quase instantaneamente, em tempo real. A existência da imagem, as suas formas, o seu movimento tornam-se em grande medida dependentes deste diálogo com o sujeito.” [Couchot, 25]

Mas embora a realidade virtual tenha dispositivos como o capacete de visão⁸¹, as luvas de dados⁸², o macacão de dados⁸³, os fones de áudio etc, tudo isto ainda não foi suficiente para colocar o usuário completamente imerso na representação. É um começo. Uma avanço considerável em relação aos dispositivos de entrada que usamos hoje nos computadores.

A realidade virtual, com o aparato tecnológico que possui hoje, ainda esbarra nos limites lógicos entre o que é visto e o que deveria ser sentido. Se o usuário que está “imerso” na realidade virtual tenta socar um muro “representado”, ele não sente dor. Ele sente como se sua mão passasse pela parede. Como afirma Cadoz, “queremos que a máquina se mostre a nós como nosso meio ambiente natural. Em outros termos, queremos que ela o represente de forma que possa interagir com ele” [Cadoz, 1997:69]; “deve haver uma correspondência

⁸⁰ Literalmente significa “arrastar e largar”. Na prática significa pressionar o botão do *mouse* em algum ícone da interface e arrastar sem soltar até um outro ponto da interface, quando liberamos o botão do *mouse* para que algo ocorra.

⁸¹ “O capacete de visão é um dispositivo de visualização estereoscópica das imagens em síntese, dotado de um captador dos movimentos da cabeça que permite simular as posições relativas realistas do operador com relação à cena tridimensional visual.” [Cadoz, 1997:102]

⁸² “Luva de dados é uma luva dotada de captadores que permitem enviar a um computador informações sobre os movimentos dos dedos da mão.” [idem, 105]

⁸³ “Macacão de dados é o macacão dotado de captadores vestido pelo operador. Permite ao computador receber dados sobre os movimentos efetuados por aquele.” [idem, 106]

natural sobre todos os aspectos acessíveis à experiência sensorial, assim como sobre os próprios fenômenos de interação.” [idem,71] Mas ainda não é isto o que acontece.

Talvez a imersão total, a ilusão completa, a representação perfeita, a interação integral sobre todos os nossos sentidos seja somente um ideal. Conforme o postulado mcluhaniano⁸⁴, os dispositivos de imersão são mera extensões de nosso corpo. Podemos perder a noção de que eles estão lá, como ferramentas, mas eles nunca deixarão de ser ferramentas. O mesmo podemos dizer do ambiente virtual. Como afirma Cadoz, “a representação consiste em substituir pura e simplesmente o objeto que se quer representar por um objeto que se considera idêntico a ele.” [idem, 73] Mas estes dois objetos “têm que ter pelo menos uma diferença: a que nos permite considerar que são dois (...) Dois objetos só podem ser parcialmente idênticos.” [idem, 74] Neste sentido, o objeto representado vai ser sempre um signo, um representante de um objeto real, e de acordo com nossa interpretação, uma metáfora que substitui o ícone algorítmico pelo signo visual.

Mas a realidade virtual é a vida como metáfora. Construída segundo a estrutura permutativa de nossa vida, as coisas acontecem por associação. Se esticamos o braço, devemos sentir a resposta imediata na representação. Se movimentamos as pernas, damos um passo virtual. Todos os movimentos realizados quando imergimos na realidade virtual são metáforas: o andar, o nadar, o voar. Como afirma Plaza:

“A animação de imagens em tempo real torna -se uma atividade lúdica e estética. A evolução, a transformação e a metamorfose de imagens transformam em metáforas da vida as suas representações. O crescimento, a expansão, a mudança e a metamorfose simulam um tipo de pensamento icônico. O prazer encontra-se na comparação entre aquilo que se imaginou e o que vemos nascer.” [Plaza, 1991:68]

Talvez seja esta a maior das vocações da realidade virtual: a simulação de movimentos, de ações, de sensações. Graças à capacidade de cálculo do computador, representa-se um ambiente a partir de um modelo e podemos vivenciá-lo como se fosse real. “A realidade que a imagem numérica dá a ver é uma outra realidade: uma realidade sintetizada, artificial, sem substrato material (...) A imagem-matriz digital não apresenta mais nenhuma aderência

ao real: libera-se dele. Faz entrar a lógica da figuração na era da Simulação.” [Couchot, 1993:42] Num ambiente simulado, podemos fazer qualquer coisa sem correr os riscos que haveria se tivéssemos interagindo no real. Ao imergir no ambiente virtual, colocamos lá dentro a nossa representação. Um signo de nós mesmos. Tornamo-nos metáforas.

3.6.1 O Ciberespaço⁸⁵: o mundo como metáfora

O ambiente virtual quando interligado tem um nome: ciberespaço. Atualmente ele se manifesta, por exemplo, na forma de uma rede de computadores interconectados mundialmente, mas isto é só um começo. Com a velocidade de intercomunicação e processamento que esta estrutura vêm adquirindo de modo crescente nos últimos anos, em breve poderemos nos colocar em tempo real em qualquer ponto do planeta sem sair de casa. Algumas experiências atuais evidenciam este processo, como as videoconferências, compras virtuais e os fóruns sociais. Num futuro próximo poderemos considerar como usuais cirurgias à distância ou visitas monitoradas ao interior do corpo humano. Conceitos como presença, distância, identidade, deslocamento já são colocados em cheque.

Entrando no ciberespaço, um novo mundo é atualizado em nossos sentidos. Este mundo pode ser construído a partir do transporte de algo que já existe no ambiente real, levado a cabo por câmeras de vídeo, microfones e *scanners* de imagens ou a partir de algoritmos e modulações. Este jogo intrigante pode resultar nas mais diferentes caracterizações, que vão desde a representação por telepresença⁸⁶, passando pelos *sites* típicos da Internet, até uma realidade virtual interligada⁸⁷. Se por um lado podemos “estar” telepresentes em qualquer lugar do mundo sem de fato estar fisicamente no local, por outro lado, podemos imergir

⁸⁴ “The wheel is an extension of the foot, the book is an extension of the eye, clothing, an extension of the skin, electric circuitry an extension of the central nervous system.” In: Marshall McLuhan and Quentin Fiore, *The medium is the Message*, 1967.

⁸⁵ O termo ciberespaço ou *cyberspace* foi criado em 1984 pelo escritor de ficção científica William Gibson no seu livro *Neuromancer*. Para o autor, o ciberespaço é um espaço que não se manifesta fisicamente, mas que se sustenta num conjunto de redes de computadores por onde circulam dados das mais variadas formas de representação. Hoje o conceito de ciberespaço pode ser ampliado pela perspectiva da realidade virtual, pois as redes vão ser interligadas entre si num mundo virtual de três dimensões.

⁸⁶ “Situação de interação entre um operador e um ambiente distante na qual, através de dispositivos de transmissão bidirecionais das informações e das ações, damos ao operador a sensação de estar presente no ambiente.” [Cadoz, 1997:109]

num mundo completamente construído por computadores, como o demonstrado no filme “O passageiro do Futuro”⁸⁸. O que há a acrescentar ao caso característico da realidade virtual é o fato de mais de um operador poder imergir ao mesmo tempo em determinado ambiente, colocando-os em interação. Isto já existe, de alguma forma, nos joguinhos de computador que permitem a oponentes disputarem remotamente.

O ciberespaço é portanto um ambiente virtual que pode ser compartilhado por qualquer pessoa que queira interagir com outra(s) remotamente a partir de qualquer lugar do mundo. Podemos estar “ciberpresentes” ao mesmo tempo em diversos pontos do planeta, em cada lugar cumprindo um papel diferente: participando de uma videoconferência em Nova York, jogando uma partida de xadrez contra um oponente em Moscou, controlando uma máquina instalada em São Paulo e até mesmo imerso numa montanha-russa virtual. Numa situação como esta, o *eu* de cada pessoa se multiplicará, incluindo nesta lista o *eu* real, que poderá estar comendo um pastel enquanto controla tudo. A própria Internet já nos dá uma pequena amostra disto. “Com as redes, estamos vivendo a experiência de habitar dois mundos (o real e o ciberespaço) ao mesmo tempo.” [Leão, 1999:109]

O caso dos joguinhos remotos que citamos acima, aliás, é bem ilustrativo neste sentido: neles temos dois *eu's*, um é o que entendemos como o nosso *eu* real, aquele que está segurando o *mouse* ou o *joystick* e controlando o jogo, e o outro é o *eu* virtual, que está inserido no ambiente do joguinho. O *eu* é ao mesmo tempo dissolvido (porque, em princípio, nem um dos dois têm exclusividade nas preocupações e processos mentais da pessoa) e multiplicado (pois são dois, mas poderiam ser ainda mais).

⁸⁷ Quero dizer neste caso a construção de um ambiente virtual, conforme o apregoado pelo que conhecemos como realidade virtual, mas que pode ser manipulado e imerso por mais de um operador ao mesmo tempo.

⁸⁸ Dirigido por Brett Leonard, *O Passageiro do Futuro* (1992) foi o primeiro filme a explorar a realidade virtual como elemento dramático, a partir da utilização de efeitos de animação computadorizada. No filme, o personagem principal vivido por Jeff Fahey é usado em experiências científicas relacionadas à realidade virtual. A imersão do personagem no mundo imaginário faz com que ele adquira, numa proporção sem precedentes, conhecimentos que o permitem controlar o mundo. Mas o aspecto a considerar para esta pesquisa é que o filme demonstra de modo competente a caracterização do que seria um mundo de imersão virtual.

Na telepresença, esta impressão de dissolução do *eu* se manifesta de modo ainda mais contundente, uma vez que, de certo modo, todo um ambiente real é “transportado” aos nossos sentidos. “Operando uma dissolução do espaço, a telepresença oferece ao homem um grande número de novas maneiras de ser: ‘estar’ num lugar sem nele permanecer fisicamente; num lugar perigoso (radioativo, tóxico, cheio de minas, sujeito a agressões diversas...) ou inacessível porque lá não sobreviveríamos (fundo do mar, atmosferas não respiráveis, espaço...)” [Cadoz, 1997:87] Se não podemos estar em um local colocamos um “representante” de nosso *eu* no lugar. Uma espécie de procurador. A diferença é que este representante não é real, é apenas um signo de nosso *eu*.

Mas podemos estar presentes em diversos lugares sob toda forma significativa, ora com a nossa representação icônica, ora com uma representação simbólica (como por exemplo alguém interagindo com o nosso nome, estampado através de um texto na tela), ora com uma representação indicial (nossa fotografia) etc. Ao imergir no ciberespaço seremos metáforas de nós mesmos. Nossas ações serão metáforas. Transformaremos-nos em signos.

A dissolução do conceito de identidade, no entanto, não se manifesta somente na representação do nosso *eu* por signos. Deve ser colocado em discussão também quem controla tal signo. Enquanto eles forem controlados pela própria pessoa que é representada, a legitimidade estará assegurada. Mas no momento em que outro “cibernauta”⁸⁹ estiver controlando de algum modo o nosso *eu* virtual, haverá uma invasão de identidade.

Esta invasão, evidentemente, pode se manifestar em diversos níveis. Desde um nível mais inofensivo, quando alguém no ciberespaço se faz passar por nós mesmos para conseguir algo para si; até numa invasão total, como a que foi demonstrada no filme “Matrix”⁹⁰. Nesta ficção científica, o ciberespaço praticamente aniquila o espaço real, num nível tal,

⁸⁹ Termo tipicamente utilizado para designar o usuário que interage ou “havega” no ciberespaço.

⁹⁰ Filme de 1999, dirigido e escrito por Andy Wachowski e Larry Wachowski, a partir do livro “Neuromancer” de William Gibson. No filme, Neo (vivido pelo ator Keanu Reeves) é um hacker que sente que sua vida é dominada por uma força estranha, mas que ele não consegue compreender (o Matrix). A partir de suas investigações pessoais, ele é descoberto e levado por uma rebelde chamada Trinity (Carrie-Anne Moss) para o esconderijo de um líder contra-revolucionário, conhecido como Morpheus (Laurence Fishburne). Morpheus acredita que Neo é um predestinado, e que poderá ajudá-lo a vencer a ditadura cibernética, conhecida como Matrix. Matrix é uma representação simbólica para uma entidade dominadora que estabelece o seu controle pela ocupação virtual das pessoas no ciberespaço.

que as pessoas não percebem mais a sua existência física. É o fim da mediação. As pessoas não sentem mais o mundo físico, vivem quase completamente imersas no seu signo, que só se manifesta no mundo da fantasia do ciberespaço. Enquanto isso, o *eu* real de cada ser vivo fica hibernado em cápsulas de segurança do anulado mundo real. Quem controla tudo é Matrix, um ser imaterial, tecnológico, digital, que domina o *eu* de todo ser real. Este deixa de ser sujeito de seus atos para transformar-se em objeto de uso cibernético. Numa realidade como esta, a interface torna-se num mundo como metáfora.

Por outro lado, numa perspectiva contrária ao controle apocalíptico do Matrix, podemos vislumbrar um ciberespaço interpessoal, construído concomitantemente por bilhões de mentes, em constante e rizomática expansão. Um espaço cibercultural, uma rede de inteligência coletiva, mágica, mas um local conectado na realidade. Como afirma Thimoty Leary: *“the information ... are going to link amplified individual minds into a global groupmind. Interactivity is interpersonal. (...) The personal computer is in the process of becoming the interpersonal computer.”*⁹¹ [Laurel, 1992:230] Este ciberespaço, orgânico e global, seria a metáfora de um cérebro social, onde estaria instalada toda a inteligência mundial.

3.7 A inversão do postulado: da representação ao objeto

Finalmente há que se considerar a inversão do postulado da representação. Na concepção tradicional, temos um objeto a partir do qual é formado um signo de representação. Agora há também o sentido contrário: um objeto pode ser desenvolvido a partir de um modelo representado em computador. Arlindo Machado diz que “vivemos uma época em que os objetos com os quais nos relacionamos no dia-a-dia são, cada vez mais, ‘cópias’, ‘reflexos’, ‘duplos’ das imagens sintetizadas no computador.” [Machado, 1993:121]

Com sua capacidade de síntese e simulação, o computador permite que um objeto possa ser construído virtualmente, com todas as vantagens que isto implica em termos de custo e

segurança. Experiências deste tipo já existem na indústria automobilística e de aviação, no *design* de objetos utilitários, no desenvolvimento de embalagens ou mesmo na área de serviços, como no treinamento de pessoal e na simulação de interferências no ecossistema. Como afirma Couchot, “hão se trata mais de figurar o que é visível (nota: muitas vezes com a intenção de ir além do visível, de exprimir o invisível, o inteligível): trata-se de figurar aquilo que é modelizável.” [Couchot, 1993:43]

O caráter mais ou menos inofensivo desta inversão talvez esconda conseqüências esquizofrênicas quando mergulharmos na imensidão ciberespacial. Num mundo onde tudo é representação, o objeto físico perde *status*. Seria um mundo composto por metaobjetos, que alterariam a sua forma ao gosto da representação. Este novo paradigma que postula uma inversão ontológica do objeto talvez nos obrigue a buscar uma resposta na metafísica. Agora, quem é o objeto e quem é a representação? Teríamos um mundo em que os objetos torna-se-iam metáforas de sua representação. Seria o fim ou a vitória da metáfora? Ou ambas, já que sua vitória poderia significar o seu próprio aniquilamento.

⁹¹ “A infomação está vindo para ligar para ampli ar as mentes individuais em uma mente agrupada grobal. A interatividade é interpessoal (...) O computador pessoal está num processo de tornar-se o computador interpessoal.”

Conclusão

Voltemos ao pressuposto de Lévy: “A breve história do diálogo entre o homem e o computador pode ser analisada como um esforço para sustentar o abismo entre linguagens formais e línguas naturais. Através de uma série cada vez mais longa de tradutores intermediários, através da invenção de novos órgãos de entrada e saída, visuais, táteis e acústicos, através de sutis efeitos cênicos, flexibiliza-se a rigidez mecânica da comunicação formal.” [Lévy, 1998:30]

Embora tantos tradutores já tenham sido criados, parece que o homem não se vê por satisfeito. Primeiro foram as linguagens formais de alto nível, como Pascal, Fortran, Cobol etc. Com aparecimento da interface gráfica vieram as linguagens visuais e os aplicativos, depois a manipulação direta dos sistemas hipermídia e a realidade virtual. Como afirma Gassée: *“if you you want to think about the future of the human-computer interface, think about the history and future of symbolic systems. The true antecedents of tomorrow’s computers were not calculating machines or electronic circuits, but alphabets, natural and format languages, and the symbolic language known as science.”*⁹² [Laurel, 1992:226] Embora esta seja uma defesa muito clara do papel das linguagens na evolução das interfaces, não podemos descartar a evolução do próprio computador e de todos os seus dispositivos. O que seria do computador hoje sem o *mouse*? E quanto às placas de vídeo, cada vez mais poderosas e velozes? Através delas, as interfaces gráficas são reconstruídas em tempo real. Portanto, a flexibilização da comunicação homem-máquina só foi possível graças à evolução tecnológica do próprio computador. Por outro lado, este processo tem se pautado por um movimento dialético de evolução entre linguagens e tecnologia. As tecnologias têm evoluído e para suprir este avanço são criadas novas linguagens. Estas trazem consigo um novo leque de possibilidades criativas, que clamam por novos avanços tecnológicos. E assim por diante.

⁹² T.A.: “Se você quiser pensar sobre o futuro das interfaces computacionais, pense na história e no futuro dos sistemas simbólicos. A verdadeiro antecedente do computadores do futuro não eram as máquinas de calcular ou os circuitos eletrônicos, mas os alfabetos, e as línguas formais e naturais, e a linguagem simbólica conhecida como ciência.”

Nesta linha evolutória, a metáfora teve participação ativa. Primeiro no processo de formação de palavras para comandos e dispositivos. A denominação para *mouse* é um exemplo. Depois na composição de interfaces gráficas que são interagidas por manipulação direta, cujo exemplo clássico é o da metáfora do tampo de escrivaninha (*desktop*), presente na maioria dos sistemas operacionais atuais. Este modelo de construção, a partir das metáforas, foi utilizado também nos aplicativos, como os processadores de texto, as planilhas eletrônicas, os tocadores de mídia, os softwares de autoria etc. Mais tarde, o tropo entrou como um recurso de inspiração para as interfaces hipermídia de CD-ROMs multimídia e de páginas para a Internet gráfica. Paralelamente a tudo isto, a metáfora continuou a formar palavras para a nova terminologia da informática, especialmente para a Internet, e ajudou o público leigo a entender um pouco mais dos processos e ações relacionadas ao computador, através de comparações. Por exemplo, é recorrente dizer que a Internet é uma “biblioteca” ou uma “rede” mundial e que “havegamos por um mar de informações” etc. Enfim, a participação da metáfora nesse processo de transformação cultural é de grande importância.

Mas por que a metáfora? Antes de mais nada, porque ela é um recurso da linguagem. Se as linguagens evoluem, a metáfora participa do processo. Depois porque a metáfora é um tradutor, e como tal tem uma função que é denotativa, cognoscitiva, retórica, conforme apregoado por Aristóteles; uma função de facilitar a comunicação. Por outro lado, a metáfora é um recurso tradutor especial, porque além da função retórica, tem também uma função poética, que é conotativa, emotiva, catártica. Este caráter da metáfora “aproxima” o computador do usuário, pois é aberto, informal ou até mesmo apelativo, conforme Jakobson. De um lado temos o mundo formal, técnico, hostil, representado pela tecnologia; de outro, há o usuário, que é leigo a este mundo e só quer utilizar o computador como uma ferramenta. A metáfora diminui o abismo entre estes dois mundos de modo muito eficiente, pois a sua tradução se opera de modo bem particular: além de ser cognitiva, também é poética.

Mas nada disso seria possível se a metáfora não se adequasse ao novo paradigma de apresentação de informações no computador, que é ditada por uma interface gráfica. Sua contribuição estaria resumida ao período em que o computador se apresentava numa

interface por linhas de comando, típica de uma linguagem de carácter lingüística. Com as interfaces gráficas, as metáforas se manifestam nos diversos ícones que caracterizam esta forma de apresentação. A tradução se estabelece na substituição do ícone algorítmico pelo signo visual. Do objeto virtual para o objeto visual, que é atualizado e vemos na tela do computador. Este processo de virtualização pode chegar ao ponto da representação total que caracteriza a realidade virtual, em que cada elemento componente da interface seria a metáfora de seu correspondente objeto real, com o efeito adicional da interação.

Voltemos novamente a Ted Nelson. O autor percebeu o prazer do usuário quando uma interface é construída com o que ele denomina princípio da virtualidade – “uma representação da realidade que pode ser manipulada”. Quanto mais realística for a representação, melhor para ele. Ele quer imergir na interface, e cada nova linguagem computacional que é criada possibilita isto de forma mais contundente.

Por outro lado, um novo movimento começa a emergir. O usuário também quer, cada vez mais, que o computador fale, escute, interaja. As primeiras aplicações de reconhecimento de fala e de voz são provas disto. Assim, se de um lado já experienciamos a realidade virtual, em que a representação é uma espécie de metáfora de nossa realidade palpável, por outro lado, poderemos vir a fazer do computador algo tão manifesto quanto o homem: ele fala, escuta, cheira, “sente”. O limite seria transformar o computador num *clone* humano, numa metáfora do próprio homem. Quando este dia chegar, teremos dois movimentos. Pois se entramos na interface do computador, também queremos que ele “entre” na nossa.

Referências Bibliográficas

- [Cadoz, 1997]: CADOZ, Claude. *Realidade Virtual*. São Paulo, Ática, 1997.
- [Cohen, 1974]: COHEN, Jean. *Estrutura da Linguagem Poética*. São Paulo, Cultrix, 1974
- [Couchot, 1993]: COUCHOT Edmond. ‘Da Representação à Simulação: Evolução das Técnicas e das Artes da Figuração’. In: Parente, André (org.). *Imagem Máquina: A era das tecnologias do Virtual*. Rio de Janeiro, Ed. 34, 1993
- [Couchot, 1997]: COUCHOT Edmond. *Tecnologias da simulação: Um sujeito aparelhado*. Paris, CNRS, Université de Paris, 1997.
- [Dondis, 1991]: DONDIS, Donis A. *Sintaxe da Linguagem Visual*. São Paulo, Martins Fontes, 1991.
- [Dubois, 1991]: DUBOIS, Jean. *Dicionário de Lingüística*. São Paulo, Cultrix, 1991.
- [Eco, 1971]: ECO, Umberto. ‘Os códigos visuais’. In: *A Estrutura Ausente: Introdução à Pesquisa Semiológica*. Perspectiva, 1971. P.97-121
- [Fidalgo, 1999]: FIDALGO, António. ‘A Biblioteca Universal na Sociedade de Informação’. In: *Revista da Comunicação e Linguagem*. Departamento de Ciências da Comunicação da Faculdade de Ciências Sociais da Universidade Nova de Lisboa. Editora Cosmos, 282-87: , 1999.
- [Filipak, 1983]: FILIPAK, Franciso. *Teoria da Metáfora*. Curitiba, HDV, 1983.
- [Gombrich, 1999]: GOMBRICH, E. H. *A História da Arte*. Rio de Janeiro, LTC, 1999.
- [Heckel, 1993]: HECKEL, Paul. *Software Amigável: técnicas de projeto de software para uma melhor interface com o usuário*. Rio de Janeiro, Editora Campus, 1993.
- [Jakobson, 1969]: JAKOBSON, Roman. *Lingüística e Comunicação*. São Paulo, Cultrix, 1969.
- [Laurel, 1992]: LAUREL, Brenda (ed.). *The art of human-computer interface design*. Apple Computer & Addison-Wesley Publishing Company, 1992.
- [Leão, 1999]: LEÃO, Lucia. *O Labirinto da Hiperídia*. São Paulo, Iluminuras, 1999.
- [Lévy, 1996]: LÉVY, Pierre. *O que é o virtual?* Rio de Janeiro, Editora 34, 1996.
- [Lévy, 1998]: LÉVY, Pierre. *A Máquina Universo: Criação, Cognição e Cultura Informática*. Porto Alegre, Artmed, 1998.
- [Lynch, 1999]: LYNCH, Patrick J. & HORTON, Sarah. *Web Style Guide: Basic Design Principles for Creating Web Sites*. New Haven e Londres, Yale University Press, 1999.
- [Machado, 1993]: MACHADO, Arlindo. *Máquina e Imaginário: O Desafio das Poéticas Tecnológicas*. São Paulo. EDUSP, 1993
- [Machado, 1997]: MACHADO, Arlindo. ‘Hiperídia: O Labirinto como Metáfora’. In Domingues, Diana (org.). *A Arte do Século XXI*. São Paulo. UNESP, 1997.
- [Mandel, 1997]: MANDEL, Theo. *The Elements of User Interface Design*. Nova York, Wiley Computer Publishing, 1997.
- [Manta, 2000]: MANTA, André. ‘O Jornal como Metáfora’.
<http://facom.ufba.com.br/pesq/cyber/manta/index.html>
- [Meirelles, 1994]: MEIRELLES, Fernando de Souza. *Informática: Novas Aplicações com Microcomputadores*. São Paulo, Makron, 1994.
- [Monteiro, 1996]: MONTEIRO, Mário A. *Introdução à Organização de Computadores*. Rio de Janeiro, LTC, 1996.
- [Munari, 1968]: MUNARI, Bruno. *Design e Comunicação Visual*. São Paulo, Martins Fontes, 1968.
- [Peirce, 1975]: PEIRCE, Charles S. *Semiótica e Filosofia*. São Paulo, Cultrix & Edusp, 1975
- [Plaza, 1986]: PLAZA, Julio Gonzalez. *Videografia em Videotexto*. São Paulo, Hicitec, 1986.
- [Plaza, 1987]: PLAZA, Julio Gonzalez. *Tradução Intersemiótica*. São Paulo, Perspectiva, 1987.
- [Plaza, 1991]: PLAZA, Julio Gonzalez. *A imagem Digital: Crise dos Sistemas de Representação*. Tese de Livre Docência. Escola de Comunicações e Artes da Universidade de São Paulo, 1991.
- [Pouillon, 1966]: POUILLON, Jean (e outros). *Problemas do Estruturalismo*. Rio de Janeiro, Zahar, 1966.
- [Ricoeur, 1975]: RICOEUR, Paul. *A Metáfora Viva*. Porto, Editora RÉES, 1975
- [Sacks, 1992]: SACKS, Sheldon (org.). *Da Metáfora*. São Paulo: Educ-Pontes, 1992
- [Santaella, 1983]: SANTAELLA, Lúcia. *O que é Semiótica*. São Paulo, Ed. Brasiliense, 1983
- [Shneiderman, 1987]: SHNEIDERMAN, Ben. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Publishing Company, 1987.

- [**Smith, 1982**]: SMITH, D. & IRBY, C. & KIMBALL, R & VERPLANK, W. & HARSLEM, E. "The Star User Interface: An Overview". AFIPS National Computer Conference, jun. 1982. pp. 512-528.
- [**Steadman, 2000**]: STEADMAN, Carl. "Faceless Interface: Transparent Design and The Evolution of Internet Radio". In: *Websights: The Future of Business and Design of The Internet*. New York, RC Publications, 2000, p.124-133.
- [**Takahachi, 1988**]: TAKAHACHI, Tadao. *Introdução à Programação Orientada a Objetos*. Curitiba, EBAI, 1988.
- [**Tanenbaum, 1992**]: TANENBAUM, Andrew S. *Organização Estruturada de Computadores*. Rio de Janeiro, PHC, 1992.
- [**Tanenbaum, 1995**]: TANENBAUM, Andrew S. *Sistemas Operacionais Modernos*. Rio de Janeiro, PHC, 1995.
- [**Thüring, 1995**]: THÜRING, Manfred & HANNEMANN, Jörg & HAAKE, Jörg M. "Hype rmedia and Cognition: Designing for Comprehension". In: *Communication of the ACM*, vol.38:nº 8:57-65, agosto-1995.
- [**Tori, 1996**]: TORI, Romero. "Mapas e Metáforas Espaciais em Sistemas de Hipermídia". In: *Graf & Tec*, São Paulo, 0:0, jul. 1996.

Bibliografia

- BABO, Maria Augusta. "O Hiperlivro: Ainda um Livro?": In: *Revista da Comunicação e Linguagem*. Departamento de Ciências da Comunicação da Faculdade de Ciências Sociais da Universidade Nova de Lisboa. Editora Cosmos, 25-26: , 1999.
- BARBOSA, João Alexandre. *A Metáfora Crítica*. São Paulo, Editora Perspectiva, 1974.
- BARBOSA, Simone D. J. & LUCENA, Carlos José Pereira de & SOUZA, Clarisse Sieckenius. "Supporting Metaphorical Mappings to Facilitate End -User Programming". In: *Monografias em Ciência da Computação*. PUC, Rio de Janeiro, nº 30/97, set. 1997.
- BOLTER, Jay David. "Topographic writing: hypertext and the eletronic writing space". In: Landow, George P. & Delany, Paul. *Hypermedia and literary studies*. Cambridge, MIT press, 1990.
- BUSH, Vannevar. "As We May Thi nk". In: Druckrey, Timothy (ed.). *Eletronic Culture: Technology and Visual Representation*. New York, Aperture, 1996.
- CADOZ, Claude. *Realidade Virtual*. São Paulo, Ática, 1997.
- CANDIDO, Celso. "As transformações da subjetividade no contexto de emergência da hipermídia". <http://www.terravista.ciclone.com.br/praiabrava/1433/artigo.htm>.
- CHAVES, Eduardo O. C. *Multimídia: Conceituação, Aplicações e Tecnologia*. Campinas, People, 1991.
- _____. "Virtualização da Realidade". In: *Revista Comunicação & Educação*. São Paulo, 16:26-28, set/dez 1999.
- CHERUBIM, Sebastião. *Dicionário de Figuras de Linguagem*. São Paulo, Pioneira, 1989.
- COHEN, Jean. *Estrutura da Linguagem Poética*. São Paulo, Cultrix, 1974
- COHEN, Ted. "A Metáfora e o Cultivo de Intimidade". In: Sacks, Sheldo n (org.). *Dá Metáfora*. São Paulo, EDUC, 1992.
- CORREIA, Carlos. "Conceito de Representação em Multimedia". In: *Revista da Comunicação e Linguagem*. Departamento de Ciências da Comunicação da Faculdade de Ciências Sociais da Universidade Nova de Lisboa. Editora Cosmos, 25-26: , 1999.
- COUCHOT Edmond. "Da Representação à Simulação: Evolução das Técnicas e das Artes da Figuração". In: Parente, André (org.). *Imagem Máquina: A era das tecnologias do Virtual*. Rio de Janeiro, Ed. 34, 1993
- _____. *Tecnologias da simulação: Um sujeito aparelhado*. Paris, CNRS, Université de Paris, 1997.
- DAVIDSON, Donald. "O que as metáforas significam". In: Sacks, Sheldon (org.). *Dá Metáfora*. São Paulo, EDUC, 1992.
- DONDIS, Donis A. *Sintaxe da Linguagem Visual*. São Paulo, Martins Fontes, 1991.
- DUBOIS, Jean. *Dicionário de Lingüística*. São Paulo, Cultrix, 1991.
- ECO, Umberto. "Os códigos visuais". In: *A Estrutura Ausente: Introdução à Pesquisa Semiológica*. Perspectiva, 1971. P.97-121
- ERICKSON, Thomas D. "Working with Interface Metaphors". In: Laurel, Brenda. *The Art of Human Computer Interface Design*. Apple Computer, 1992.
- FIDALGO, António. "A Biblioteca Universal na Sociedade de Informação". In: *Revista da Comunicação e Linguagem*. Departamento de Ciências da Comunicação da Faculdade de Ciências Sociais da Universidade Nova de Lisboa. Editora Cosmos, 282-87: , 1999.
- FILIPAK, Franciso. *Teoria da Metáfora*. Curitiba, HDV, 1983.

- GASSÉE, Jean-Louis. "The Evolution of Thinking Tools". In: Laurel, Brenda. *The Art of Human Computer Interface Design*. Apple Computer, 1992.
- GOMBRICH, E. H. *A História da Arte*. Rio de Janeiro, LTC, 1999.
- HECKEL, Paul. *Software Amigável: técnicas de projeto de software para uma melhor interface com o usuário*. Rio de Janeiro, Editora Campus, 1993.
- JAKOBSON, Roman. *Linguística e Comunicação*. São Paulo, Cultrix, 1969.
- KAHN, Paul. "Visual Cues for Local and Global Coherence in the WWW". In: *Communication of the ACM*, vol.38:nº 8:67-70, agosto-1995.
- KAY, Alan. "User Interface: A Personal View". In: Laurel, Brenda. *The Art of Human Computer Interface Design*. Apple Computer, 1992.
- LANDOW, George. *Hypertext: the convergence of contemporary critical theory and technology*. Baltimore, John Hopkins Press. 1993.
- LAUREL, Brenda (ed.). *The art of human-computer interface design*. Apple Computer & Addison-Wesley Publishing Company, 1992.
- LEÃO, Lucia. *O Labirinto da Hipermídia*. São Paulo, Iluminuras, 1999.
- LEARY, Timothy . "The Interpersonal, Interactive, Interdimensional Interface". In: Laurel, Brenda. *The Art of Human Computer Interface Design*. Apple Computer, 1992.
- LÉVY, Pierre. *A Máquina Universo: Criação, Cognição e Cultura Informática*. Porto Alegre, Artmed, 1998.
- _____. *O que é o virtual?* Rio de Janeiro, Editora 34, 1996.
- _____. "The Art of Cyberspace". In: Druckrey, Timothy (ed.). *Electronic Culture: Technology and Visual Representation*. New York, Aperture, 1996.
- LYNCH, Patrick J. & HORTON, Sarah. *Web Style Guide: Basic Design Principles for Creating Web Sites*. New Haven e Londres, Yale University Press, 1999.
- MACHADO, Arlindo. "Hipermídia: O Labirinto como Metáfora". In Domingues, Diana (org.). *A Arte do Século XXI*. São Paulo. UNESP, 1997.
- _____. *Máquina e Imaginário: O Desafio das Poéticas Tecnológicas*. São Paulo. EDUSP, 1993
- _____. "Repensando Flusser e as imagens técnicas". In: *Revista da Comunicação e Linguagem*. Departamento de Ciências da Comunicação da Faculdade de Ciências Sociais da Universidade Nova de Lisboa. Editora Cosmos, 25-26: , 1999.
- _____. "Tendencias Recientes del Media Art". In: Giannetti, C. (org.). *Arte en la era electronica*. Barcelona, L'angelot, 1997, p.27 -27.
- MANDEL, Theo. *The Elements of User Interface Design*. Nova York, Wiley Computer Publishing, 1997.
- MANTA, André. "O Jornal como Metáfora".
<http://facom.ufba.com.br/pesq/cyber/manta/index.html>
- MCLUHAN, Marshall. *A Galáxia de Gutemberg*. São Paulo, Companhia Editora Nacional, 1977.
- MIRANDA, José A. Bragança de. "Fim da Mediação: De uma agitação na metafísica contemporânea". In: *Revista da Comunicação e Linguagem*. Departamento de Ciências da Comunicação da Faculdade de Ciências Sociais da Universidade Nova de Lisboa. Editora Cosmos, 25-26: , 1999.
- MANDEL, Theo. *The Elements of User Interface*. New York, Wiley, 1997.
- MEIRELLES, Fernando de Souza. *Informática: Novas Aplicações com Microcomputadores*. São Paulo, Makron, 1994.

- MOLES, Abraham. *Teoria da Informação e Percepção Estética*. Brasília. Editora Universidade de Brasília, 1978.
- MONTEIRO, Mário A. *Introdução à Organização de Computadores*. Rio de Janeiro, LTC, 1996.
- MOUNTFORD, S. Joy. "Tools and Techniques for Creative Design". In: Laurel, Brenda. *The Art of Human Computer Interface Design*. Apple Computer, 1992.
- MUNARI, Bruno. *Design e Comunicação Visual*. São Paulo, Martins Fontes, 1968.
- NELSON, Theodor Holm. "The Right Way to Think about Software Design". In: Laurel, Brenda. *The Art of Human Computer Interface Design*. Apple Computer, 1992.
- NORMAN, Donald A. "Why Interfaces Don't Work". In: Laurel, Brenda. *The Art of Human Computer Interface Design*. Apple Computer, 1992.
- PAULA Fº, Wilson de Pádua. *Multimídia: Conceitos e Aplicações*. Rio de Janeiro, LTC, 2000.
- PEIRCE, Charles S. *Semiótica e Filosofia*. São Paulo, Cultrix & Edusp, 1975
- PINTO, Ricardo Jorge & SOUSA, Jorge Pedro. "O futuro da Internet". In: *Revista Comunicação e Sociedade*. São Paulo, 31, p. 99-116.
- PLAZA, Julio Gonzalez. *A imagem Digital: Crise dos Sistemas de Representação*. Tese de Livre Docência. Escola de Comunicações e Artes da Universidade de São Paulo, 1991.
- _____. *Tradução Intersemiótica*. São Paulo, Perspectiva, 1987.
- _____. *Videografia em Videotexto*. São Paulo, Hicitec, 1986.
- POUILLON, Jean (e outros). *Problemas do Estruturalismo*. Rio de Janeiro, Zahar, 1966.
- PRESS, Larry. "McLuhan Meets the Net". In: *Communication of the ACM*, vol.38:nº 6:15-19, jun. 1995.
- "Reading from the Map: Metonymy and Metaphor in the Fiction of Forking Paths". In: Delany, Paul & Landow, George P. *Hypermedia and Literary Studies*. Cambridge, MIT Press, 1991.
- RICOEUR, Paul. *A Metáfora Viva*. Porto, Editora RÉ S, 1975.
- _____. "O Processo Metafórico como Cognição, Imaginação e Sentimento". In: Sacks, Sheldon (org.). *Dá Metáfora*. São Paulo, Educ-Pontes, 1992.
- RUTKOWSKI Chris. "An Introduction to the Human Applications Standard Computer Interface", Part 1: Theory and Principles. *BYTE* (7, 11), Outubro 1982, 291-310.
- SACKS, Sheldon (org.). *Da Metáfora*. São Paulo: Educ-Pontes, 1992
- SANTAELLA, Lúcia. *O que é Semiótica*. São Paulo, Ed. Brasiliense, 1983
- SCHMANDT, Chris. "Ilusion in the Interface". In: Laurel, Brenda. *The Art of Human Computer Interface Design*. Apple Computer, 1992.
- SHNEIDERMAN, Ben. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Publishing Company, 1987.
- SMITH, D. & IRBY, C. & KIMBALL, R & VERPLANK, W. & HARSLEM, E. "The Star User Interface: An Overview". *AFIPS National Computer Conference*, jun. 1982. pp. 512-528.
- STEADMAN, Carl. "Faceless Interface: Transparent Design and The Evolution of Internet Radio". In: *Websights: The Future of Business and Design of The Internet*. New York, RC Publications, 2000, p.124-133.
- TAKAHACHI, Tadao. *Introdução à Programação Orientada a Objetos*. Curitiba, EBAI, 1988.
- TANEMBAUM, Andrew S. *Organização Estruturada de Computadores*. Rio de Janeiro, PHC, 1992.

- _____. *Sistemas Operacionais Modernos*. Rio de Janeiro, PHC, 1995.
- THÜRING, Manfred & HANNEMANN, Jörg & HAAKE, Jörg M. ‘Hypermedia and Cognition: Designing for Comprehension’. In: *Communication of the ACM*, vol.38:nº 8:57-65, agosto-1995.
- TORI, Romero. *Hipermídia e Multimídia: Introdução e Terminologia Básica*. Apostila do Curso PCS-722 (Escola Politécnica – USP). São Paulo, 1995.
- _____. ‘Mapas e Metáforas Espaciais em Sistemas de Hipermídia’. In: *Graf & Tec*, São Paulo, 0:0, jul. 1996.
- TURKLE, Sherry. ‘Constructions and Reconstructions of the self in virtual reality’. In: Druckrey, Timothy (ed.). *Electronic Culture: Technology and Virtual Representation*. New York, Aperture, 1996.