

Conceitos cognitivos e modelos mentais no uso de sistemas computacionais

Cognição

A cognição se refere aos processos por meio dos quais nos relacionamos com as coisas ou, em outras palavras, a como ganhamos conhecimento.

O principal objetivo na cognição associada à IHC tem sido compreender e representar a forma como as pessoas interagem com computadores em termos de como o conhecimento é transmitido entre os dois.

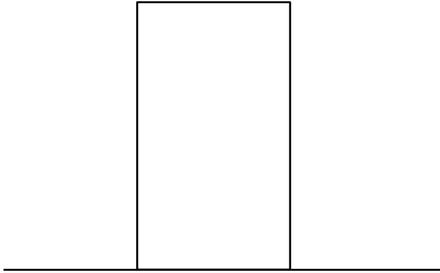
A Psicopatologia das Coisas do Dia-a-dia (Norman 88)

“Kenneth Olsen, the engineer who founded and still runs Digital Equipments Corp., confessed at the annual meeting that he can’t figure out how to heat a cup of coffee in the company’s microwave oven.”

“Você necessitaria um grau de engenheiro para resolver o problema.”

- Objetos de *design* pobre podem ser difíceis e frustrantes de usar;
- Eles proporcionam falsas dicas, ou não proporcionam dica alguma;
- Eles atrapalham o usuário no seu processo de interpretação e compreensão.

- Objetos bem desenhados são fáceis de interpretar e compreender;
- Eles contêm dicas visíveis para a sua operação.



Uma porta sem visibilidade

Exemplo 1: uma porta

Não há muito que se fazer com uma porta: **abrir** ou **fechar**;

Você se defronta com uma porta:

- Como ela abre? Ela é de **correr** ou de **girar**?
- Se ela for de girar, tem que **puxar** ou **empurrar**? De que **lado**?
- Se ela for de correr, ela abre para a **esquerda** ou para a **direita**? De que **lado**?

A resposta deveria ser dada através do *design*, sem necessidade de ensaio e erro!

O exemplo ilustra um dos princípios fundamentais do *design*: a **visibilidade**.

As partes corretas devem ser visíveis, e devem passar a mensagem correta, através de **sinais naturais**, que são interpretados sem necessidade de se ter consciência dos mesmos. ("**design natural**")

Outro problema diz respeito ao **mapeamento entre a intenção e o que parece possível**.

Exemplo 2: um projetor de slides

Considere um projetor de *slides* com apenas **1 botão**;

Você se defronta com ele:

- Como correr para a **frente**? Como correr para **trás**?
- Um único botão para fazer as 2 coisas?

A resposta foi conhecida através de pessoas que já tinham usado o projetor (e passado pelo mesmo impasse!):

- **Toque curto:** move para a frente;
- **Toque longo:** move para atrás.

Decididamente, um único controle não seria o ideal para representar a funcionalidade disponível.

Nesse caso, seria preciso, então, recorrer a noção de congruência, revelando, no controle único, a semântica associada às funções disponíveis.

Exemplo 3: uma máquina de lavar

Imagine um casal inglês - um engenheiro e uma física - que acabava de comprar uma máquina de lavar italiana último modelo que fazia tudo o que se pudesse imaginar;

Indagados acerca de como conseguiam operar a máquina, responderam:

- engenheiro simplesmente se recusava e chegar perto;
- A física tinha decorado um procedimento que resolvia o seu problema comum: lavar roupa.

Se o *design* era tão ruim ao ponto de inviabilizar o uso das funções realmente distintas do aparelho, por que eles o adquiriram?

Os dispositivos no dia-a-dia têm problemas associados: funções e mais funções, controles e mais controles.

O usuário necessita de ajuda. As coisas corretas precisam estar **visíveis**: precisam indicar **quais** partes operam o artefato e **como**; indicar de que forma o usuário deve interagir com ele.

Exemplo 4: o controle de um refrigerador

Resposta tardia leva ao estrago dos produtos nele guardados.

A **visibilidade da resposta** (ou **feedback**) deve ser imediata. De nada adianta, por exemplo, a gente se dar conta, no outro dia, de que a programação feita estava incorreta.

Exemplo 5: a função de *redial* automático dos aparelhos telefônicos

Linhas congestionadas podem causar a efetivação da ligação horas depois da primeira tentativa e da chamada da função;

Você recebe a ligação:

“Alô, quem fala?” “Como ‘quem fala’? Você me ligou!”

Instruções complicadas não são lembradas facilmente quando necessário.

Exemplo 6: redirecionamento de ligações em centrais telefônicas

O horário de trabalho acabou, mas você está trabalhando normalmente;

O telefone toca na sala ao lado, e você não sabe como pegar a ligação;

Nem mesmo a leitura do manual de instruções (se estiver à mão), vai resolver, pois o telefone vai parar de tocar antes de você ler o necessário! (problema da circularidade do acesso à informação de ajuda!)

Exemplo 7 (de bom projeto): controle de automóveis

As coisas são visíveis; os mapeamentos entre os controles e as coisas controladas são naturais;

Em geral, cada função tem o seu controle associado (1:1);

Há bom *feedback*; o sistema é compreensível;

Os relacionamentos entre as intenções do usuário (o motorista), as ações requeridas e os resultados são sensatos, não-arbitrários e com significado.

Conceitos cognitivos

Existem alguns conceitos cognitivos, isto é, associados às atividades mentais realizadas pelo usuário quando do uso de um artefato tecnológico, que servem como princípios para o *design* de interfaces, aplicando-se principalmente ao *design* de interfaces gráficas ou de manipulação direta. Os principais princípios são descritos a seguir.

Em resumo, a idéia consiste em que os controles e as suas limitações têm que ser visíveis, com um bom mapeamento entre as operações sobre eles e os seus efeitos, e o *design* deve também sugerir (*afford* - “bancar”) sua funcionalidade.

Nos **carros**, em geral, as coisas são visíveis.

- Há bons mapeamentos entre os controles e o seu efeito, entre as metas e as necessidades do motorista e as funções disponíveis no painel de controle;
- Um controle normalmente tem uma única função;
- Há bom *feedback* e o sistema é compreensível;
- Em geral, os relacionamentos entre as metas do usuário, as ações requeridas e os resultados são sensatos, têm significado e são não-arbitrários.

Já em **aparelhos eletrônicos**, como aparelhos de vídeo, por exemplo, isto não ocorre.

- Não há estrutura visível;
- Os mapeamentos entre os controles e os seus efeitos são arbitrários, e não há correspondência entre as metas do usuário e os botões e/ou *displays* que compõem a interface (o painel de controle);
- Muitos botões têm múltiplas funções;
- Há um *feedback* pobre, e o usuário normalmente fica em dúvida de se o resultado desejado será atingido.

Affordances

Affordance é um termo técnico que se refere à propriedade de os objetos deixarem explícita a informação das operações e das manipulações que podem ser feitas sobre os mesmos.

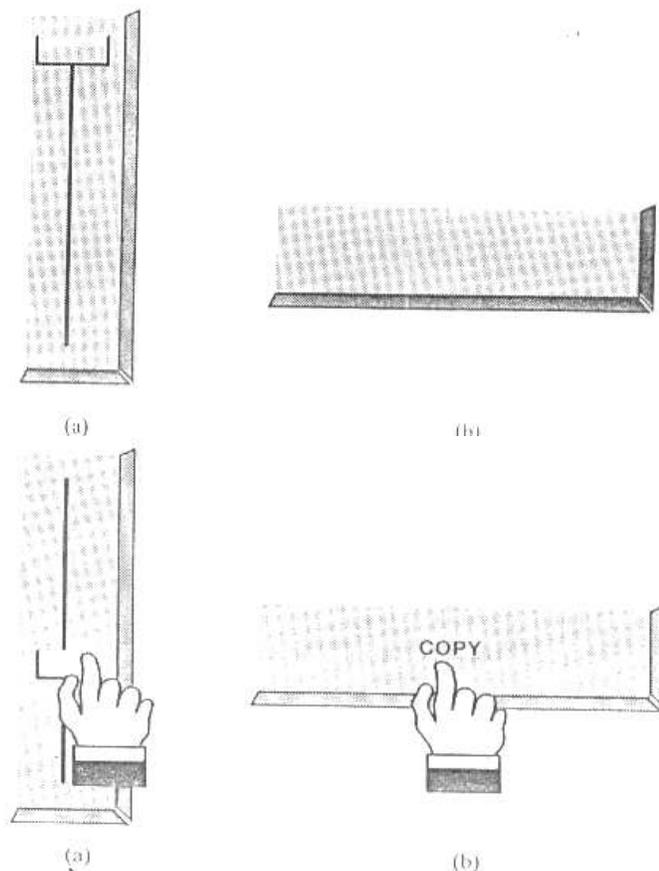
As portas, por exemplo, sugerem as possibilidades de abertura, enquanto uma cadeira sugere suporte. *Affordances* jogam um papel importante no *design* de objetos, mas o mais importante é a sua percepção pelo usuário.

O que se enxerga do comportamento de um objeto ou sistema é aquilo que pode ser *afforded* ou permitido pelo mesmo. Quando as *affordances* de um objeto ou sistema são reveladas, perceptivamente óbvias, é fácil o usuário interagir com ele. Contrariamente, quando as *affordances* são menos óbvias ou ambíguas, é fácil o usuário ter falsas expectativas e realizar erros ao interagir com o objeto ou sistema.

Por exemplo, o *design* de uma determinada porta deixa evidente se ela é de correr ou de puxar? O *design* seria bom, do ponto de vista da *affordance*, se a porta tivesse uma barra (para porta de correr) ou um puxador (para porta de puxar).

As setinhas na barra de *scroll* são um bom exemplo de *affordance* bem-resolvida.

Infelizmente, aspectos estéticos às vezes entram em conflito e se sobrepõem à *affordance*, comprometendo-a no resultado final do processo de *design*.



Affordance e duas maneiras de resolvê-la

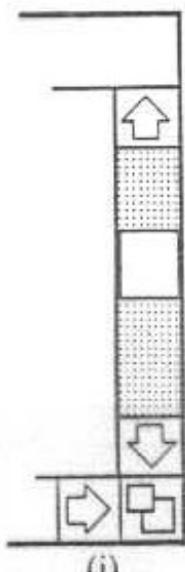
Restrições

Enquanto a *affordance* sugere o escopo de um objeto em termos do que pode ser feito com ele, as restrições (*constraints*) limitam as possibilidades do que pode ser feito com ele.

Norman (1992) sugere que há 4 tipos de restrições: restrições **físicas**; restrições **semânticas**; restrições **culturais**; restrições **lógicas**.

As **restrições físicas** restringem as possíveis operações sobre um objeto. Numa casinha didática para aprender as formas pelas crianças de 1 a 2 anos, há orifícios em diferentes formas e objetos que devem ser inseridos nos mesmos. A forma do orifício é uma restrição física que leva a criança a perceber qual o objeto que cabe ali.

Na barra de *scroll*, as linhas da esquerda e da direita, juntamente com as setas, restringem os movimentos possíveis ao de-baixo-para-cima e de-cima-para-baixo.



Uma boa *affordance* sugerindo, além das possibilidades, as restrições.

As **restrições semânticas** dependem da semântica da situação de uso que o usuário conhece. Dependem do conhecimento do mundo e da situação específica em que o sistema computacional ou artefato tecnológico está inserido.

O cesto de lixo, por exemplo, só faz sentido se dentro de uma metáfora de escritório. Ele é representado em pé, pois se fosse de-cabeça-para-baixo contrariaria nosso conhecimento prévio de como as coisas são usadas.

No entanto, a posição espacial do cesto relativamente à mesa de trabalho é semanticamente inconsistente na interface do Mac. Ninguém tem um cesto sobre a sua mesa. Os cestos normalmente ficam debaixo da mesa ou fora do prédio. Por que, então, ele é posto sobre a mesa na interface? Porque, se fosse colocado debaixo, ficaria escondido e não seria visível.

As **restrições culturais** consistem na informação e nas regras que permitem que nos comportemos em ambientes sociais. De forma análoga, muitos ícones podem ser criados de forma a capitalizar convenções culturais.

Por exemplo, um sinal de interrogação é associado culturalmente com um sistema de perguntas e respostas, um ambiente de auxílio ou *help*.

As **restrições lógicas** limitam a ordem, a posição e/ou a localização do objeto. A exibição de itens de um menu é logicamente restrito na medida em que a lista de itens é exibida dentro de uma linha condutora, horizontal ou vertical. Seria ilógico se as diversas opções estivessem dispostas aleatoriamente na tela em diversas direções.

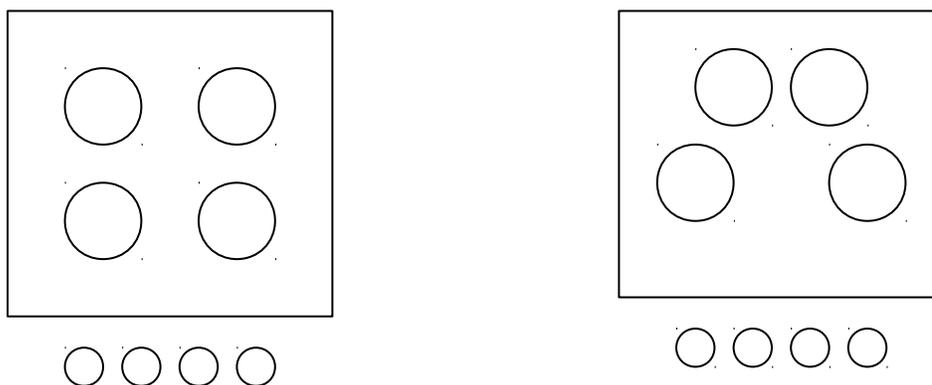
As interfaces gráficas e na Web abrem a possibilidade de que menus adquiram formatos alternativos, mas a importância de se restringir a sua apresentação a formas lógicas continua vigente.

Mapeamentos

Os mapeamentos entre as metas e intenções do usuário e os controles disponíveis na interface são bons se eles são naturais.

Mapeamentos são ruins quando os relacionamentos entre metas e controles são inconsistentes, ou incompatíveis.

Um exemplo em que normalmente ocorre inconsistência entre as intenções do usuário e as operações disponíveis é no painel nos fogões. É raro quando o *design* deixa claro qual botão acende qual boca do fogão.



Mapeamentos tradicionalmente problemáticos

Outro tipo comum de mapeamento pobre ocorre quando há vários botões ou itens de menu com rótulos de significado parecido na língua-mãe do usuário, dificultando a interpretação controle-ação.

As opções Recortar e Excluir na planilha EXCEL apresentam este problema.

Visibilidade e *Feedback*

A **visibilidade** indica o mapeamento entre as **ações** pretendidas pelo usuário e as **operações** disponíveis para acionar o aparelho.

A **visibilidade** indica **distinções** cruciais, que permitem, por exemplo, distinguir entre sal e pimenta.

A **visibilidade da resposta** do sistema permite verificar se o **resultado** esperado foi ou não alcançado.

A visibilidade tem estreita relação com o **mapeamento entre os controles e as funções associadas**.

Controles com mais de uma função associada são não-naturais, difíceis de lembrar e de usar;

Quando o número de controles coincide com o número de funções, cada controle pode ser especializado, rotulado; as funções disponíveis são visíveis e, se o usuário esquece alguma função, o controle associado serve como lembrete.

O **feedback** pode ser definido como o “envio de informação ao usuário sobre a ação que ele efetivamente executou e qual o resultado associado”.

Em interfaces de manipulação direta ou gráficas, o *feedback* pode ser textual (como nos outros estilos de interação), visual ou auditivo.

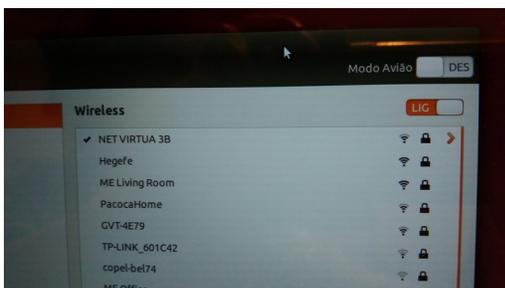
Contra-exemplos



ON / OFF

Utilização de **1 único botão** para indicar **2 estados**

O estado do sistema não é aparente.



Modo Avião ?!

Wireless ?!

Utilização de **1 único controle** (embora com 2 posições diferentes):

- **para a Entrada** (ativar um modo específico) **E**
- **para a Saída** (exibir estado do sistema)

Deixa ambíguo o caráter do botão!

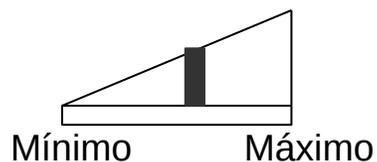
O Modo Avião está DESligado?

(O objeto tem por função exibir o estado do sistema.)

Eu preciso mover o botão para a posição DES para desligar?

(O objeto é um botão de controle propriamente dito.)

A congruência minimizando limitações físicas:



Download from
Dreamstime.com
This watermark-free image is for previewing purposes only.

11667044
Planctonvideo | Dreamstime.com

Para trás (antihorário)

Para a frente (horário)

O feedback é instantâneo.

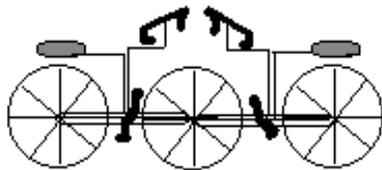
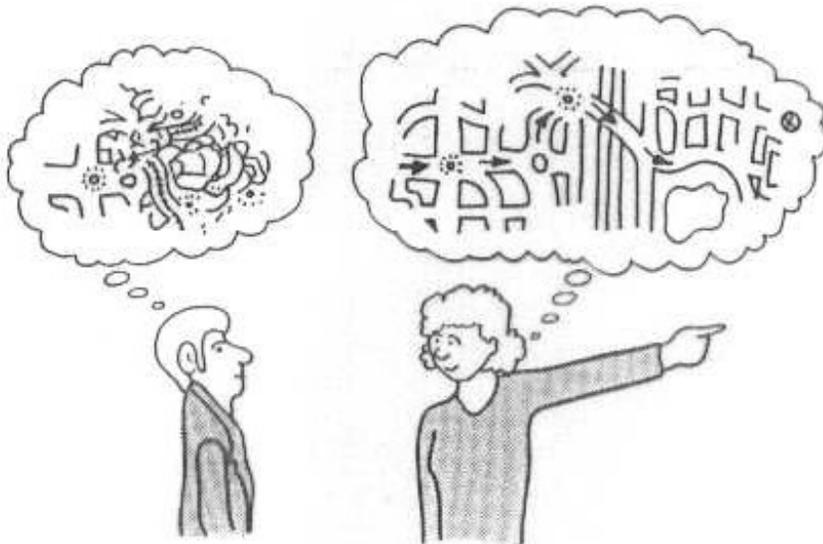


Os botões são simbólicos* e indicam a ação (movimento e sentido). (* ou resemblance?)

Questões para reflexão: potencial e problema do Controle Universal



Modelos conceituais



Uma bicicleta diferente

Como a gente sabe que a bicicleta da figura não vai funcionar?

A gente tem um **modelo conceitual** ou **modelo mental** do funcionamento de uma bicicleta.

Modelos mentais são os modelos que as pessoas têm de si mesmas, dos outros, do ambiente e das coisas com as quais elas interagem.

Modelos mentais e acomodação entre modelos

As pessoas podem construir os modelos mentais através de experiência, de treinamento ou de instrução. Por isso, é comum encontrar artefatos ou programas (assim como o projetor de slides do exemplo) que tem uma interface pobre mas que é dominada por quem já usou o artefato.

Um exemplo claro da existência e do acomodamento de modelos mentais é o uso de uma nova aplicação (tal como um processador de textos) diferente daquela com a qual a gente está acostumada. O fato de termos um modelo mental de como um processador funciona, das funções que ele tem, permite que a gente faça uma série de previsões em relação à nova aplicação, transformando a tarefa de aprendizagem numa acomodação de estruturas (se os modelos forem muito diferentes) e uma tradução entre linguagens (“A função **save** aqui é **gravar**.”).

O modelo mental de um dispositivo ou artefato é formado através da interpretação das ações percebidas e de sua estrutura visível.

Um bom modelo conceitual permite antecipar o resultado das ações e verificar se os efeitos são de fato os esperados.

O modelo conceitual das coisas do dia-a-dia não precisam ser muito complicados.

Não é necessário entender as propriedades físicas ou químicas subjacentes a cada artefato, mas, sim, tornar explícito o relacionamento (**mapeamento**) entre os **controles** e os **resultados das ações**.

Da mesma forma, o modelo conceitual das interfaces homem-máquina não precisa - nem deve - incorporar conceitos de eletrônica, mas, sim, tornar explícito o relacionamento entre as funções oferecidas e o resultado de sua seleção pelo usuário.

Prescrições para o processo de “*User-centered design*” (Norman 1988)

1. Facilite a determinação das ações possíveis a cada instante (faça uso de restrições);
2. Torne as coisas visíveis, incluindo o modelo conceitual do sistema, as ações alternativas e os resultados das ações;
3. Facilite a avaliação o estado corrente do sistema;
4. Siga mapeamentos naturais entre intenções do usuário e ações do sistema; entre ações e efeitos resultantes; e entre a informação visível e a interpretação do estado do sistema.

Interfaces “*user-centered*” são inteligíveis, compreensíveis, e suavizam o cruzamento do espaço entre o usuário e o sistema.

Modelo do sistema, modelo do usuário e imagem do sistema

É muito comum existir uma distância relevante entre o ambiente disponível para executar uma tarefa (ambiente de interface inadequado) e o modelo que o usuário tem dessa tarefa.

Essa distância é transposta por meio de treinamento.

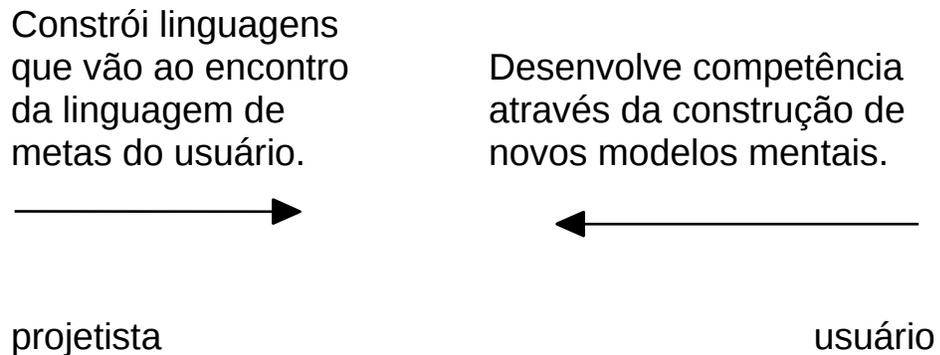


Distância entre o usuário e o sistema, e sentidos de transcurso possíveis

O modelo mental do sistema, ou “**modelo do usuário**” (que este tem do sistema) proporciona poder de previsão e explicação para a compreensão das tarefas.

Ele evolui naturalmente no uso de sistemas bem projetados.

O modelo do usuário é dependente do tipo da interação, do conhecimento prévio e da experiência anterior do usuário.



Processos assíncronos executados no percurso da distância entre o usuário e o sistema

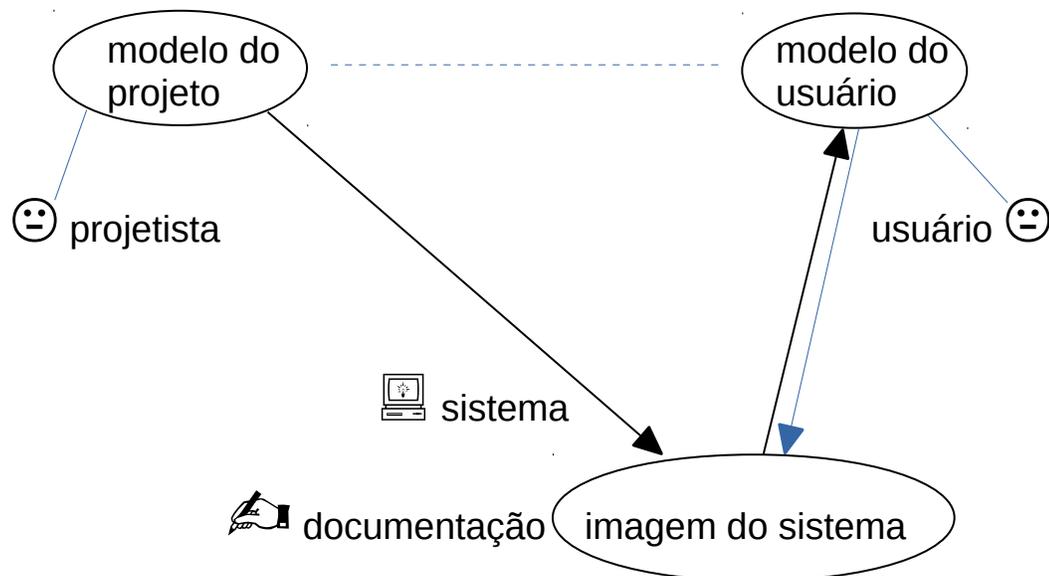
Existem três modelos envolvidos na interação entre um ser humano e um ambiente computacional.

Modelo do sistema: Conceitualização do sistema pelo projetista (modelo do projeto ou do *design*)

Modelo do usuário: Modelo conceitual da tarefa em questão desenvolvido pelo usuário (*user model*)

Imagem do sistema: Imagem física a partir da qual o usuário constrói (ou modifica) o modelo do usuário

Não existe comunicação direta entre o usuário e o projetista em tempo de uso!



Modelos mentais envolvidos no uso de ambientes computacionais

Na melhor das hipóteses, quando o projetista constrói um modelo da tarefa (modelo do sistema) próximo ao modelo do usuário, eles, projetista e usuário, se comunicam via imagem do sistema (interface e documentação), isto é, não há, na maioria das vezes, comunicação direta projetista-usuário final.

O modelo do usuário é resultante da **interpretação** que o usuário faz da imagem do sistema e não do modelo do projeto!

O projetista tem que tornar a imagem do sistema compatível com o modelo de projeto.

A imagem do sistema deve ser **explícita, inteligível e consistente**.

O projetista precisa, então acertar em duas etapas: a elaboração do modelo de projeto próxima ao modelo do usuário, e a “tradução” deste modelo, de forma consistente, para a imagem do sistema.

Nos sistemas que apresentam uma imagem clara e consistente em geral esta imagem vai ao encontro do modelo do usuário. Ou seja, o ajuste a ser feito entre os dois modelos é pequeno.

Em geral, quanto mais especializada for a ferramenta, mais alto o nível da linguagem e mais facilitada a tarefa, pois a linguagem da ferramenta fica mais próxima da linguagem da tarefa.

Sistemas de propósito geral e linguagens de programação não são facilitados pela interface.

Para sistemas muito abrangentes, é necessário definir **perfis de usuário** para viabilizar a sua compreensão e o seu uso.

Os dois **princípios fundamentais** do *design* são:

1. Proporcione um bom **modelo conceitual**;
2. Torne as coisas **visíveis**.

Eles **se traduzem em projetar um sistema que**:

1. obedeça uma conceitualização **consistente**;
2. **permita que o usuário desenvolva o modelo do usuário consistente com o modelo do projeto**.

Críticas à abordagem cognitiva

A abordagem cognitiva tradicional tem deixado de lado a importância de como as pessoas trabalham no mundo real ao usar sistemas computacionais. Mais ainda, foram negligenciados outros aspectos do comportamento exceto a forma como as pessoas interagem com o sistema: como as pessoas interagem entre si e com outros objetos ao redor do sistema, o ambiente no qual eles estão inseridos.

Novas perspectivas consistem em esquemas interdisciplinares que auxiliam ao *design* de sistemas reais para pessoas reais para levar adiante atividades de trabalho reais em ambientes de organizações reais. (Preece, 1994)

A idéia principal é substituir o termo tradicional “fatores humanos” (que coloca o usuário como ser passivo), por “atores humanos” (que implicando um ser ativo no controle). (Bannon, 1991)

Cognição distribuída

A cognição distribuída consiste em prover explicações que vão além do indivíduo, conceitualizar atividades cognitivas como encorpadas e situadas em um contexto de trabalho no qual elas ocorrem (Hutchins, 1990; Hutchins e Klausen, 1992).

Três preocupações da cognição distribuída são:

- a análise de como os diferentes componentes do sistema são coordenados. Isto envolve a análise de como a informação é propagada através do sistema funcional em termos de aspectos tecnológicos, cognitivos, sociais e organizacionais;
- consciência da situação (Norman, 1993) envolve a comunicação entre o grupo que utiliza o sistema;
- análise e explicação dos cortes (*breakdowns*) na coordenação que emergem nos ambientes de trabalho (Rogers, 1993) O termo *breakdowns* é utilizado para descrever os vários incidentes, problemas, ineficiências, mal-entendidos e acidentes que ocorrem no ambiente de trabalho.