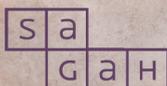


INTERFACE HUMANO- COMPUTADOR

Leonara de Medeiros Braz



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Modelagem de interfaces

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Descrever modelos e técnicas de modelagem de interfaces em IHC.
- Discutir modelos de tarefas e modelos de interação.
- Ilustrar representações e modelos utilizados no *design* de interface com o usuário.

Introdução

A interface é o canal de comunicação entre o usuário e o sistema e, por isso, precisa ser bem estruturada para apresentar de forma clara e consistente o que o usuário pode (e como) realizar dentro do sistema. Porém, a especificação da estrutura da interface do sistema depende de outras características, como a especificação dos modelos de interação e das funcionalidades do sistema.

O processo de *design* é uma atividade intelectual e criativa de desenvolver um sistema computacional a partir de especificações de necessidades dos possíveis usuários do sistema. Assim, é preciso conhecer as técnicas específicas para uma boa coleta e para a análise das informações necessárias a fim de que essas informações apoiem o processo de proposta do sistema computacional.

Além disso, o produto deve ser apresentado ao usuário para que seja avaliado, de modo que erros possam ser corrigidos e o conhecimento do *designer* seja aprofundado. Dessa forma, o processo de *design* deve ser cíclico, passando por essas fases constantemente até que o produto seja aprovado e o processo tenha sido finalizado.

Neste capítulo, portanto, considerando a importância da interface para o usuário dos sistemas, você vai conhecer modelos e técnicas de modelagem que existem e são utilizados para o oferecimento de sistemas de qualidade. Para isso, vai conhecer modelos de interfaces, modelos de tarefas e modelos de interação considerando a relação usuário/sistema.

1 Conceitos básicos

Para que a proposta de interface tenha qualidade e apresente as informações de forma estruturada, outros pontos devem ser considerados no momento do processo de *design*. A especificação do processo de interação apresenta ao *designer* como o usuário irá interagir com o sistema. Entretanto, para que o estilo de interação seja escolhido de forma precisa, é necessário compreender o domínio da aplicação: quem são os usuários, conhecimentos, necessidades e até mesmo o local de utilização do sistema. Dessa forma, fica claro para o *designer* o que o sistema deve disponibilizar para o usuário, qual é a melhor forma de o usuário interagir com o sistema e como as informações devem estar disponibilizadas para facilitar o processo de interação.

Desse modo, o processo de *design* assume três etapas primordiais para que o sistema seja visto com qualidade: análise, síntese (ou intervenção) e avaliação. A **análise** consiste em compreender a situação atual, quem são as pessoas impactadas pelo problema, os processos atuais realizados, os artefatos utilizados e outras características que sejam importantes para a compreensão do problema e do contexto que será trabalhado. A **síntese** é uma atividade mais criativa; com base na análise do domínio e na situação atual, o *designer* propõe soluções que atendam as necessidades observadas. Como última fase, o resultado da síntese deve ser avaliado pelos usuários. A fase de **avaliação** é de extrema importância para que o *designer* compreenda se a proposta realmente atende as especificações do usuário e se existem problemas a serem corrigidos.

Recomenda-se que o processo seja cíclico e que o produto apresentado na fase de síntese configure partes do produto (a concepção da ideia, um protótipo do sistema, uma funcionalidade), de modo que o *designer* não perca muito tempo construindo um sistema robusto que pode ser descartado posteriormente. A Figura 1 apresenta um modelo genérico do processo de *design* em IHC. Alguns processos de *design* apresentam a sequência que deve ser seguida durante a realização das atividades, mas, para Lawson (2006), o que realmente importa é partir de um problema, realizar o processo de *design* e, no final, chegar a uma solução.

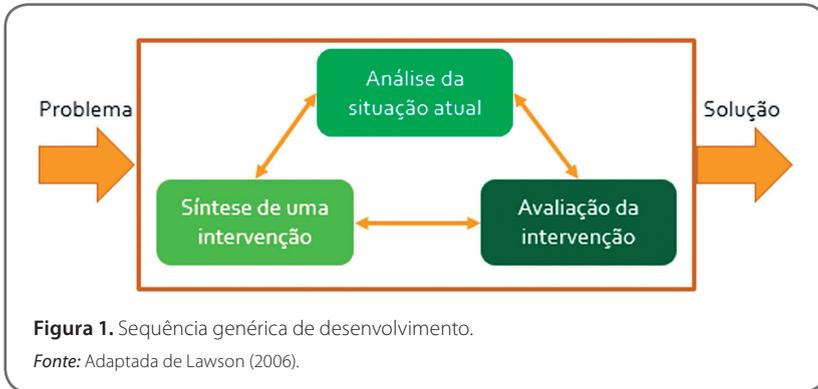


Figura 1. Sequência genérica de desenvolvimento.

Fonte: Adaptada de Lawson (2006).

Técnicas para a modelagem de interface

Vale ressaltar que modelar a interação do sistema não significa modelar a interface, mas, para que se tenha uma boa estruturação da interface, é importante ter modelado o processo de interação. “Modelando o comportamento do sistema pode-se compreender a interação deste com os atores externos e daí refletir nos requisitos de interação, que acabarão por auxiliar na formação da arquitetura da interface do usuário” (GONÇALEZ; SANTORO; MONTE, 2010, p. 5).

Dessa forma, para ter uma boa interface (etapa de síntese, na qual o *design* usa sua criatividade para propor uma solução ao problema enfrentado pelo usuário), é necessário passar pela fase de análise.

A fase de análise, em processo de *design* de sistemas computacionais, investiga três pontos cruciais: usuários; atividades realizadas e objetivos pretendidos; e contexto de uso (físico, cultural e social). Considerando esses pontos, as informações apresentadas e a estrutura da interface tendem a estar de acordo com as expectativas dos usuários, sendo mais efetivas e aceitas.

O usuário deve ser o “centro” de interesse do *designer*, que, portanto, deve seguir o processo de *design* centrado no usuário, buscando garantir que as expectativas dos usuários sejam atendidas por meio do sistema desenvolvido e considerando as características e necessidades do seu público-alvo (NORMAN, 2013).



Saiba mais

O *design* centrado no usuário (do inglês *User-Centred Design*) é um termo cunhado por Norman e Draper em 1986 e que se popularizou principalmente após a publicação do livro *User-Centered System Design: New Perspectives on Human-Computer Interaction*. (NORMAN; DRAPER, 1986) Essa teoria coloca as necessidades dos usuários como o centro da tomada de decisão no processo de *design*.

Com a utilização do *design* centrado nos usuários, algumas técnicas para modelagem de interface podem ser utilizadas, como: modelagem de usuários do sistema e modelagem de cenário de interação.

Com a modelagem do usuário, busca-se entender para quem estamos construindo o sistema, respondendo perguntas como: quem são meus usuários? Quais são seus objetivos? O primeiro passo para que se faça um registro das características coletadas dos usuários é traçar o seu perfil. Segundo Barbosa e Silva (2010, p. 175) “em geral, um *designer* começa seu trabalho com uma ideia inicial de quem são seus usuários, mas essa ideia não costuma ser suficientemente detalhada e pode até ser apenas uma impressão equivocada.” Dessa forma, esse processo deve ser iterativo, sendo aperfeiçoado com o tempo. Veja, a seguir, alguns dados que podem ser analisados nessa etapa do projeto.

- **Papel:** qual é o papel que o usuário vai desempenhar no sistema? Quais tarefas, primárias e secundárias, podem ser desempenhadas por ele? Dados pessoais também podem ser colhidos a fim de detalhar os diferentes tipos de usuários (idade, sexo, etc.).
- **Conhecimento do domínio:** qual é o nível de conhecimento dos usuários com a utilização da aplicação? Existem diferentes níveis de usuários (especialistas e novatos)?
- **Familiaridade com tecnologia:** qual é o grau de familiaridade dos usuários com tecnologia (leigos, especialistas)?
- **Frequência de uso:** com qual frequência os usuários utilizam a aplicação?

Essas informações auxiliam o *designer* a compreender quem são seus usuários e as atividades realizadas. Com essas informações, o *designer* pode categorizar seus usuários, separando-os por grupo, de modo a “generalizar” as informações a respeito desses usuários. A essa generalização, damos o nome

de persona. Uma persona é um personagem, uma pessoa fictícia que representa as características de um grupo de usuário. Geralmente, esses personagens são utilizados desde o início do processo de *design*, ajudando a equipe nas decisões a serem tomadas (BARBOSA; SILVA, 2010).

A utilização das personas durante o processo de desenho do sistema torna claros os objetivos dos usuários, ajudando o *designer* a compreender como estruturar as tarefas de modo a facilitar o entendimento sobre o que o sistema deve fazer. Cooper acredita que uma boa prática é tentar projetar o sistema para atender uma única persona em vez de tentar ampliar as funcionalidades de modo a acomodar a maioria das pessoas (COOPER; REIMANN; CRONIN, 2007; COOPER, 1999).

É importante ressaltar que uma persona não deve corresponder a uma pessoa real, apenas as características que elas recebem devem ser reais. Dessa forma, nome e detalhes pessoais são informações criadas, “dando vida” a esse personagem; os demais detalhes (habilidades, objetivos, tarefas, relacionamentos, etc.) devem ser derivados dos dados das coletas realizadas. Quanto maior o nível de detalhamento dessas informações, mais a persona auxiliará o *designer*. A Figura 2 apresenta um exemplo de persona; note que, além do nome e das características pessoais, a persona ganha uma imagem que a representa, tornando-se uma pessoa “concreta” e que faz parte do processo de *design*.

Paulo Correa, técnico de suporte – “comandos para máxima eficiência”

Paulo Correa, de 43 anos, trabalhou durante muitos anos consertando e configurando computadores. Atualmente, trabalha na universidade AprendaMais, configurando PCs e as contas dos alunos de cada turma. Ele fez um curso de administração de rede, mas prefere aprender fazendo do que assistindo a cursos ou lendo manuais. Quando tem alguma dúvida, ele faz uma busca na Internet por informações que lhe ajudem a resolver os seus problemas. Usuário “das antigas”, Paulo prefere utilizar linguagem de comando do que assistentes em interface gráfica, pois acredita que assim seja mais eficiente. Sempre que uma tarefa se repete com frequência, ele tenta elaborar um script ou fazer alguma configuração que acelere o seu trabalho.



Todo início de período, Paulo precisa configurar dezenas de contas para cada turma, com diferentes perfis, fornecendo acesso diferenciado para alunos regulares, monitores, instrutores e coordenadores de cada disciplina. Precisa atender aos pedidos dos professores sobre o que deve estar disponível na intranet de cada disciplina (e.g., publicação de material didático; fórum de discussão; recebimento de trabalhos dos alunos; cadastramento de notas; pedidos de revisão). Seu maior objetivo é atender aos professores com a maior eficiência possível. Para isso, é importante ele poder acessar o sistema onde quer que esteja, no horário que for, para realizar qualquer tarefa remotamente.

Figura 2. Exemplo de persona para sistema acadêmico.

Fonte: Barbosa e Silva (2010, p. 178).

Além das personas, outro método utilizado para modelar a interface é a criação de cenários de utilização, isto é, histórias contadas com a apresentação de detalhes e situações de uso e usuários envolvidos.

A criação de cenário conta com um enredo, apresentando: o que os usuários fazem, o que acontece com eles e que mudanças ocorrem no ambiente (BARBOSA; SILVA, 2010). Segundo Rosson e Carroll (2002), os elementos necessários para a criação de um cenário são ambientes, atores, objetivos, planejamento, ações, eventos e avaliação. Os *designers* podem fazer perguntas mais específicas de cada elemento para aprofundar seu entendimento sobre as informações presentes no cenário.

Cooper (1999) ressalta que o cenário deve abordar os diferentes usuários, contemplando seus objetivos e habilidades; adicionalmente, os cenários devem abranger ao máximo o conjunto de funcionalidades que o sistema oferece. Porém, o *designer* deve ficar atento, pois a elaboração do cenário não deve consumir muito tempo no processo de *design*; para isso, deve-se elencar os principais pontos do sistema para a elaboração de cenários. A Figura 3 apresenta um exemplo de cenário de sistema.

Cadastro de projetos finais com coorientador externo não cadastrado

Atores: Joana Marinho (secretária), Fernando Couto (aluno)

Na primeira semana de aula, Joana Marinho, secretária do curso de Engenharia Ambiental, precisa cadastrar entre vinte e trinta projetos finais dos alunos no período atual. Um projeto final é um trabalho individual de um aluno sob a orientação de um ou dois professores. Cada aluno preenche um formulário impresso e o entrega na secretaria. Em vez de cadastrar os projetos finais à medida que são entregues, Joana prefere juntar vários para cadastrá-los de uma vez, pois acha que assim perde menos tempo. Joana confere o formulário, verificando se o aluno definiu seu(s) orientador(es) e o título e formato de entrega do seu trabalho (e.g., relatório, software), para então cadastrar os dados no sistema. No caso do aluno Fernando Couto, após informar o título do trabalho e o orientador principal, Joana descobre que o seu coorientador, que não é professor regular do curso, não está cadastrado no sistema. Ela interrompe o cadastramento, pega o e-mail de Fernando da sua ficha cadastral (impressa) e lhe envia uma mensagem solicitando os dados do seu coorientador externo: nome completo, CPF e e-mail para contato. No dia seguinte, Joana recebe a mensagem de resposta de Fernando com os dados solicitados. Ela então reinicia o cadastro do projeto final de Fernando, sem poder aproveitar o que havia feito na véspera. Ao terminar o cadastro, Joana entra no seu sistema de correio eletrônico e envia uma mensagem para todos os envolvidos (aluno e coorientadores), para que eles confirmem os dados cadastrados e confirmem sua participação no projeto.

Figura 3. Exemplo de cenário para sistema acadêmico.

Fonte: Barbosa e Silva (2010, p. 190).

2 Análise e modelagem de tarefa

A análise de tarefas é realizada para que o *designer* compreenda quais são os objetivos e tarefas dos usuários.

Em IHC, a análise de tarefas pode ser utilizada nas três atividades habituais: para análise da situação atual (apoiada ou não por um sistema computacional), para o (re)design de um sistema computacional ou para a avaliação do resultado de uma intervenção que inclua a introdução de um (novo) sistema computacional (BARBOSA; SILVA, 2010, p. 191).

Diversos métodos podem ser utilizados para realizar a análise de tarefas dos usuários. Um desses métodos é o *Hierarchical Task Analysis* (HTA, ou análise hierárquica de tarefas), que quebra a tarefa do usuário em subtarefas menores — e essas, por sua vez, são quebradas em sub-subtarefas). Essas subtarefas são agrupadas em uma forma hierárquica, especificando como a tarefa é executada. As ações do usuário (relacionadas ao *software* ou não) são incluídas nessa lista de tarefas. Veja, a seguir, um exemplo de tarefas utilizando o método HTA. Com esse método, gráficos podem ser feitos para estruturar melhor os planos de ações para a realização das tarefas.



Exemplo

Pegar livro emprestado na biblioteca.

1. Ir até a biblioteca.
2. Encontrar o livro:
 - buscar o acervo da biblioteca;
 - adicionar os critérios de busca;
 - identificar se o livro apresentado é o correto;
 - anotar a localização do livro.
3. Ir até a estante em que o livro se encontra.
4. Autorizar a retirada do livro no balcão.

Outro método utilizado para a realização de análise de tarefa é o GOMS, que busca descrever uma tarefa analisando seus objetivos (*Goals*), operadores (*Operators*), métodos (*Methods*) e regras de seleção (*Selection rules*).

Existe um conjunto de modelos que segue o método GOMS conhecido como Família GOMS e que inclui, por exemplo: *Keystroke-Level Model* (KLM), CMN-GOMS e COM-GOMS (CARD; MORAN; NEWELL, 1983; JOHN; GRAY, 1995).



Link

Saiba mais sobre modelagem e tarefas acessando o *link* a seguir.

<https://qrgo.page.link/Tw6mz>

Modelagem de tarefas

A partir das informações adquiridas nas fases de coletas de informações e análise de tarefas, o *designer* pode estruturar as tarefas que o usuário desempenha, traçando caminhos que devem ser percorridos para que o objetivo final seja alcançado. Segundo Barbosa e Silva (2010), um *design* pautado na engenharia semiótica não leva em consideração apenas a sequência hierárquica da tarefa, mas também as interações realizadas e as tarefas alternativas.

A nomenclatura utilizada para representar a tarefa é guiada por objetivo de alto nível, tarefas que são realizadas e operadores, que são operações atômicas (ações que são realizadas). A Figura 4 apresenta a estrutura de representação de cada elemento.



(a) Objetivo geral



(b) Tarefa

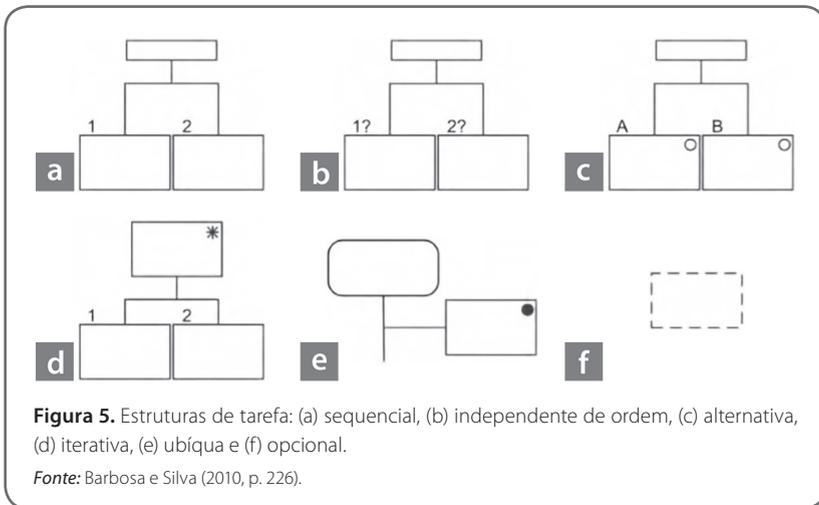


(c) Operadores

Figura 4. Representação gráfica dos elementos utilizados para modelagem de tarefa.

Fonte: Adaptada de Barbosa e Silva (2010).

As tarefas são modeladas em forma de estrutura hierárquica; desse modo, toda ação do usuário sobre o sistema ficará no último nível da “árvore” (representada por operadores). A tarefa não é uma ação em si, mas realizada de forma indireta por meio dos operadores associados a ela. Essas tarefas podem ser organizadas de diferentes formas; na Figura 5, Barbosa e Silva (2010) apresentam alguns dos vários modos de representação.



- A estrutura em sequencial (a) é representada por uma numeração, mostrando a sequência de realização das ações (por exemplo, ir para o *site*, escolher o produto, realizar a compra).
- A estrutura independente de ordem (b) é representada por números com sinal de interrogação, simbolizando que qualquer tarefa representada usando essa estrutura pode ser realizada a qualquer momento, sem necessariamente seguir uma ordem lógica (p. ex., o usuário pode realizar a compra e depois se cadastrar no *site* ou pode se cadastrar no *site* e depois realizar a compra).
- A estrutura alternativa (c) é representada por letras e o usuário deve escolher realizar uma tarefa ou outra (p. ex., filtrar um busca por preço ou por relevância).

- A estrutura iterativa (d) é representada por um asterisco e significa que a realização da sequência de atividades se dará repetidas vezes, enquanto for necessária a sua realização (p. ex., buscar produto e adicioná-lo no carrinho — o usuário faz isso quantas vezes quiser, até comprar todo o material que deseja).
- A estrutura ubíqua (e) é representada por um ponto preto e informa que aquela atividade pode ser realizada a qualquer momento, independentemente de ordem; por isso, ela geralmente fica abaixo do objetivo geral, representando que é independente das outras tarefas (p. ex., desistir de realizar a compra e sair do *site*).
- A representação opcional (f) representa uma tarefa que pode ser realizada pelo usuário, mas que não é obrigatória; assim, em um momento da interação, o usuário realiza essa tarefa e, em outro, pode escolher não realizá-la (p. ex., filtrar busca por faixa de preço até 1000 reais).

Cabe destacar, como afirmam Barbosa e Silva (2010), que, “Em geral, não modelamos as tarefas relacionadas a todos os objetivos do sistema. A modelagem de tarefas se destina principalmente a explorar tarefas com estruturas mais complexas” (BARBOSA; SILVA, 2010, p. 228).

3 Modelagem de interação

A modelagem da interação apresenta uma estrutura da conversa entre o sistema e o usuário. Nessa modelagem, elementos e ações de interface não devem ser representadas, contendo apenas ações (e troca de informações) entre o usuário e o sistema.

Paula (2003) desenvolveu uma linguagem para auxiliar no processo de modelagem da interação. A linguagem, denominada MoLIC (*Modeling Language for Interaction as Conversation*), representa a interação como uma conversa entre o *designer* (a partir do preposto do *designer*) e o usuário. As informações sobre o que o sistema faz, o que ele permite e como ele age devem ser apresentadas de forma clara a partir dessa modelagem. “A MoLIC foi projetada de modo a ser não apenas uma notação para especificar a interação, mas também como uma ferramenta epistêmica, ou

seja, para aumentar a compreensão dos designers sobre o problema sendo resolvido e o artefato sendo projetado” (BARBOSA; SILVA, 2010, p. 229).

Os elementos básicos para a criação de diagramas utilizando a linguagem MoLIC são cenas, transições e processos do sistema. A Figura 6 apresenta a representação desses elementos.

- **Cenas:** indica uma conversa (dentro da interface) sobre algum ponto que está sendo utilizado pelo usuário.
- **Transições:** falas que mudam a conversa, transitando entre uma cena e outra.
- **Processos do sistema:** momentos nos quais o sistema determina o próximo tópico da conversa.

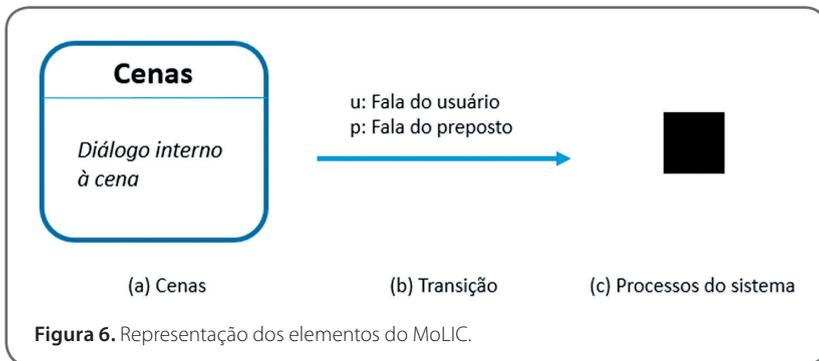
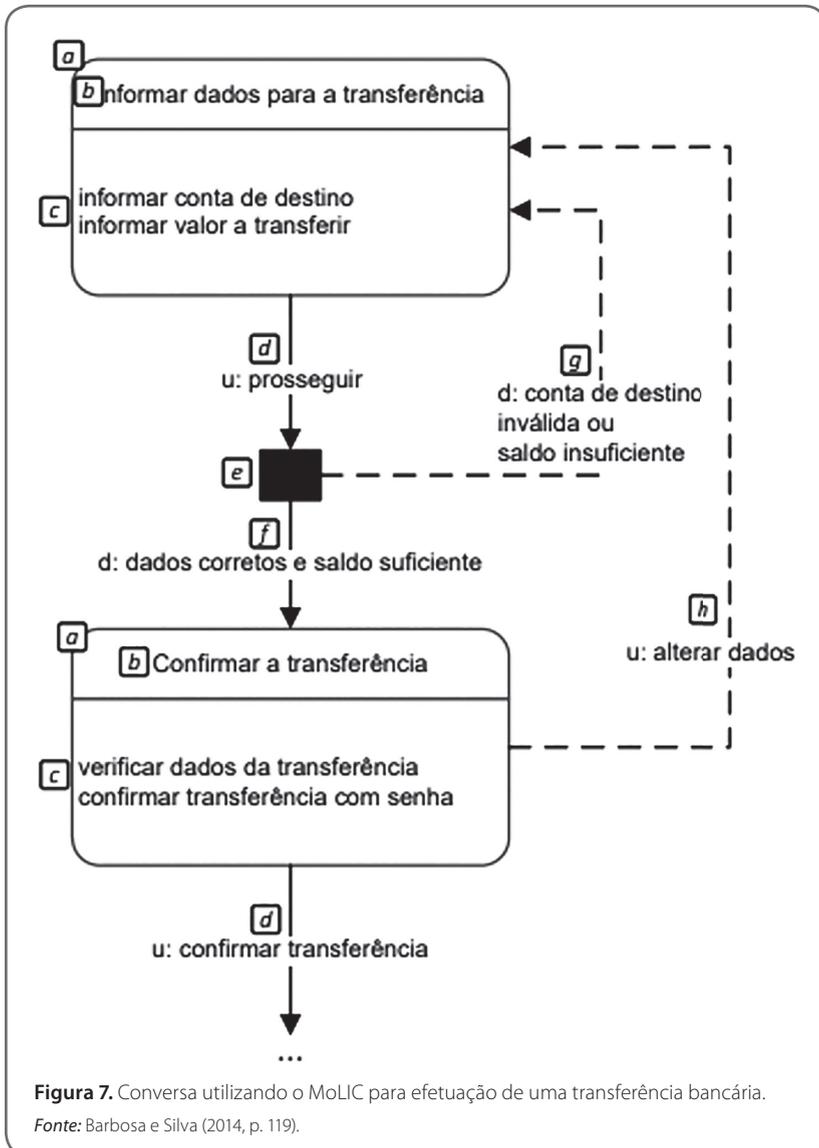


Figura 6. Representação dos elementos do MoLIC.

A Figura 7 apresenta uma representação da utilização do MoLIC para modelar a interação do sistema e representa as ações que podem existir durante a realização de uma transação bancária. Cada cena (retângulo) possui diálogos internos, que são informações que serão apresentadas aos usuários por meio da interface; esses diálogos podem ser solicitações de informações (“por favor, informe a conta de destino”) ou apresentação de informação (“Confira aqui os dados da conta. Eles estão corretos?”). As setas mostram a transição de uma cena para outra, representando a ação do usuário sobre o sistema (“Prosseguir: já inseri os dados necessários”). O quadrado preto representa o processamento, pelo sistema, dos dados enviados pelo usuário (“Opa! O usuário me enviou aqui os dados, deixa eu verificar se está tudo

correto”). As bifurcações existentes representam alguns dos possíveis cenários que podem acontecer: (1) conta inválida, (2) saldo insuficiente, (3) usuário percebeu um erro e deseja corrigir. Quanto maior a complexidade do sistema computacional, mais a representação vai aumentando e apresentando mais informações da “conversa” do sistema com o usuário.



4 Exemplo de modelagem

Considerando todos os modelos para representação da tarefa e da interação, você verá, a seguir, uma forma de utilizar as técnicas para o entendimento do problema.

Persona

José Fernandes tem 22 anos e é estudante de computação na universidade AprendaMais. José teve que se deslocar de sua cidade natal e mora com outros estudantes. Apesar de esforçado, José tem dificuldades para aprender certos conteúdos das disciplinas cursadas e, por isso, sempre que possível, aluga livros na biblioteca de sua faculdade. José também gosta de participar de grupos de estudo. A Figura 8 apresenta um infográfico com as informações de José.



Cenário

O professor da disciplina de Interação Humano-Computador apresentou em sala um novo assunto: modelagem de interface. José, para não ficar atrasado no conteúdo, resolveu buscar o livro recomendado pelo professor na biblioteca, para estudar e resolver a atividade solicitada. Porém, José não sabe a localização do livro na biblioteca. Dessa forma, busca no acervo digital, inserindo as informações necessárias para encontrar o livro desejado.

Modelagem da tarefa

Existem duas possibilidades de ação, de modo que o plano de ação traçado pode variar. Por exemplo, no cenário apresentado, podemos ter dois planos de ação. A Figura 9 apresenta a estrutura hierárquica da tarefa.

- Plano 1
 - Fazer:
 - 1. ir até a biblioteca;
 - 2. ir até a estante e pegar o livro;
 - 3. levar o livro para o balcão.
 - Se o livro não estiver na prateleira desejada:
 - 1. encontrar o livro;
 - 2. ir até a estante e pegar o livro;
 - 3. levar o livro para o balcão.
- Plano 2
 - Se não souber onde está o livro:
 - 1. acessar o acervo;
 - 2. identificar o livro;
 - 3. anotar a localização.
 - Se o livro não tiver identificação:
 - 1. acessar tela de busca;
 - 2. entrar com critérios de busca;
 - 3. identificar o livro desejado.

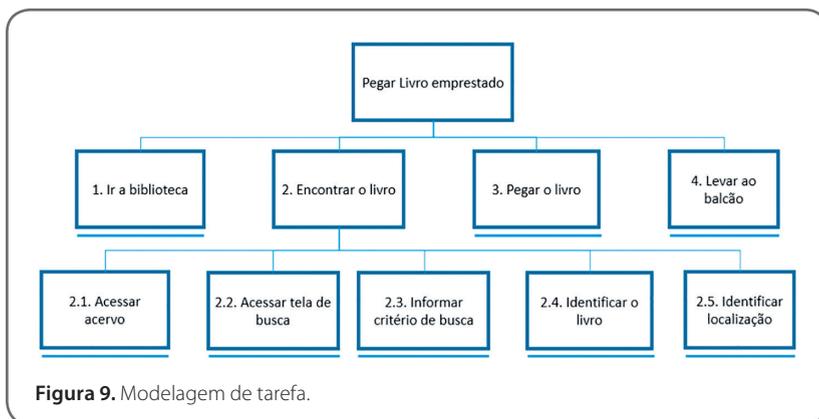
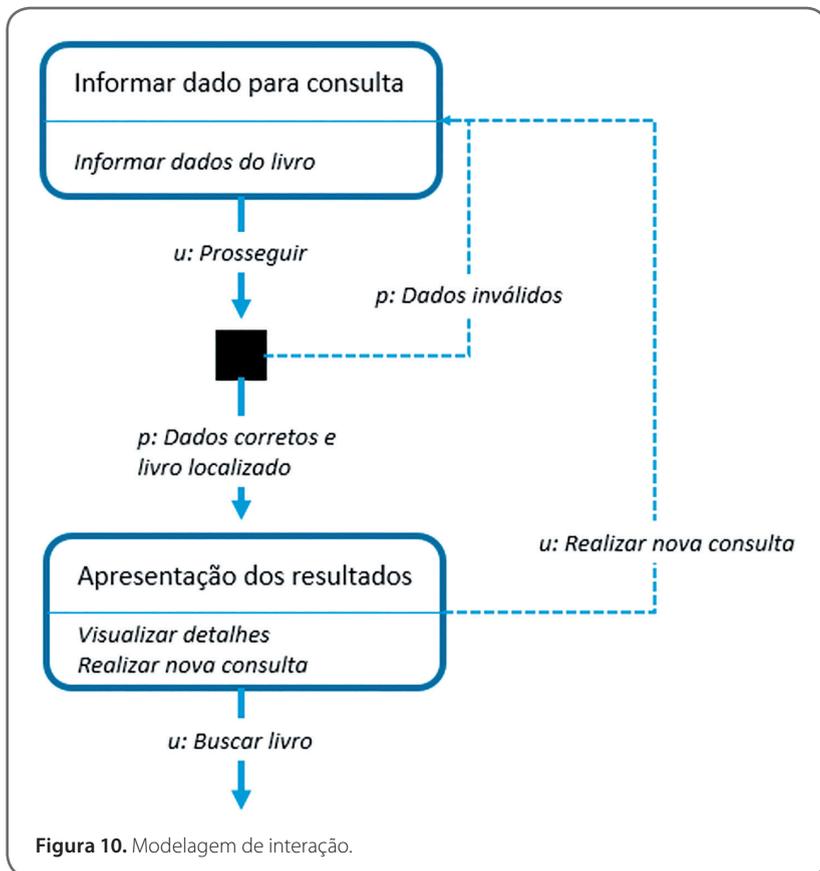


Figura 9. Modelagem de tarefa.

Modelagem de interação

A modelagem da interação apresenta a conversa entre usuário e sistema para a realização da atividade pretendida. A linguagem MoLIC apresenta uma estrutura para modelar essa conversa por meio de ações e transições entre essas ações no decorrer da interação. Na Figura 10, a representação da interação da tarefa de buscar um livro no acervo da biblioteca.



Para que haja a consulta do livro, o usuário precise informar os dados necessários exigidos pelo sistema (“Por favor, insira as informações do livro que você deseja buscar.”). Ao inserir as informações, o usuário prossegue

com a realização da busca. O sistema computa as informações fornecidas e verifica se os dados são válidos ou não: se forem inválidos, uma mensagem de erro é retornada; se os dados forem válidos, o sistema redireciona o usuário para uma nova cena e um novo diálogo acontece.

A utilização dessas diferentes técnicas auxilia a equipe de *design* a compreender de forma mais detalhada quem são seus usuários, suas necessidades e as tarefas envolvidas. Com esse entendimento, a modelagem de interface fica mais efetiva, pois produzirá um material adequado às necessidades dos usuários.



Referências

- BARBOSA, S. D. J.; SILVA, B. S. Design da interação humano-computador com MoLIC. In: IHC '14: Companion Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems, 14., Foz do Iguaçu, Brazil, 2014. *Proceedings* [...]. Porto Alegre: Sociedade Brasileira de Computação, 2014. p. 79–80.
- BARBOSA, S. D. J.; SILVA, B. S. *Interação humano-computador*. Rio de Janeiro: Elsevier, 2010.
- CARD, S. K.; MORAN, T. P.; NEWELL, A. *The psychology of human-computer interaction*. New Jersey: Lawrence Erlbaum Associates, 1983.
- COOPER, A. *The inmates are running the asylum: why high-tech products drive us crazy and how to restore the sanity*. Indianapolis: Sams, 1999.
- COOPER, A.; REIMANN, R.; CRONIN, D. *About Face 3: the essentials of interaction design*. New York: John Wiley & Sons, 2007.
- GONÇALEZ, M. A. D.; SANTORO, F. M.; MONTE, L. C. M. Modelando interfaces de usuários utilizando um processo ágil. Relatórios Técnicos do Departamento de Informática Aplicada da UNIRIO nº 0003/2010. Rio de Janeiro: Universidade Federal do Estado do Rio de Janeiro, 2010.
- JOHN, B. E.; GRAY, W. D. CPM-GOMS: an analysis method for tasks with parallel activities. In: CHI '95 CONFERENCE COMPANION: MOSAIC OF CREATIVITY, 1995. Denver, Colorado, USA, May 7-11, 1995. *Proceedings* [...]. New York: Association for Computing Machinery, 1995. p. 393–394
- LAWSON, B. *How designers think: the design process demystified*. 4. ed. Oxford: Architectural Press, 2006.

NORMAN, D. A.; DRAPER, S. W. *User centered system design: new perspectives on human-computer interaction*. New Jersey: Lawrence Erlbaum Associates, 1986.

NORMAN, D. *The design of everyday things: revised and expanded edition*. Philadelphia: Basic Books, 2013.

PAULA, M. G. *Projeto da interação humano-computador baseado em modelos fundamentados na engenharia semiótica: construção de um modelo de interação*. 2003. 87 f. Dissertação (Mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, 2003.

ROSSON, M. B.; CARROLL, J. M. *Usability engineering: scenario-based development of human-computer interaction*. San Francisco: Morgan Kaufmann, 2002.



Fique atento

Os *links* para *sites* da Web fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integridade das informações referidas em tais *links*.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

