

4

O Discurso do Preposto do Designer

Como o designer não pode estar presente fisicamente na interface, ele é representado por seu preposto. Conforme descrito anteriormente, o preposto tem uma capacidade comunicativa que lhe permite fazer um discurso completo e exclusivamente referente à conclusão final do designer sobre as necessidades, preferências, capacidades e oportunidades que ele entende que o usuário tem (interpretante congelado do designer). E a melhor alternativa que o preposto tem de realizar esta metacomunicação (comunicação sobre a comunicação designer→usuário) é o sistema de ajuda.

Melhor por quê? É a melhor alternativa em termos de: (a) variedade de códigos e meios de comunicação – o discurso do preposto pode incluir linguagem natural, imagens, diagramas, réplicas, reprodução de interações gravadas, e assim por diante; (b) variedade de níveis de profundidade do discurso – ele pode falar apenas sobre a interface, mas também pode falar sobre o porquê e para quê da interface, sobre os “por que não?”, sobre os “e se”, e toda a gama de explicações, argumentações e ilustrações que pareça necessária e conveniente para passar aos usuários a inteireza da lógica do design; (c) variedade de metas comunicativas tratáveis – o discurso tanto pode ser tutorial como minimalista e breve, tanto pode ser metafórico quanto técnico, tanto pode ser monologado quanto dialogado – isto sem mencionar a possibilidade de se atingirem várias metas simultâneas, ou de se associar certo tipo de meta a certo tipo de grupo de usuários. Poderia ser melhor? Sim, poderia ser melhor se o designer estivesse em pessoa, ou se ele tivesse um preposto humano especializado em comunicação, mas então já não estaríamos tratando da comunicação mediada por artefato computacional, que é o território em que esta tese se circunscreve.

Para a construção do discurso do preposto são usadas as expressões de acesso à ajuda. O uso destas expressões (expressões de comunicabilidade e

expressões mais utilizadas por usuários em caso de dúvida) auxiliam o designer a pensar sobre possíveis rupturas de comunicação que possam ocorrer durante a interação, as quais irão guiar a construção do conteúdo da ajuda. Este conteúdo está intimamente ligado à lógica de design, a qual é a base para a construção das instruções operacionais, táticas e estratégicas (sobre como, para que e por que utilizar a aplicação) que serão fornecidas aos usuários em resposta a estas rupturas.

Esta visão do designer – ponto de vista do designer sobre a aplicação por ele construída (questões vistas na seção 3.1) – e as instruções operacionais, táticas e estratégicas fundamentadas em sua lógica de design compõem o teor do discurso do preposto, conforme a abordagem aqui apresentada.

A forma como esta visão será transmitida ao usuário pode variar, dependendo da arquitetura que for utilizada para a construção do sistema de ajuda, mas o que deve ser preservado em qualquer arquitetura a ser construída (requisitos desejáveis) sob esta abordagem são:

- o teor desta mensagem: a visão de design;
- a possibilidade de aprofundamento da “conversa” preposto↔usuário; e,
- a disponibilidade da ajuda a qualquer momento de dúvida do usuário.

Este capítulo irá detalhar a concretização do discurso do preposto a partir de uma proposta de captura das informações necessárias para sua construção e de uma arquitetura de sistema de ajuda sob esta perspectiva, além de sua exemplificação através de um estudo de caso.

4.1

Captura dos Elementos Necessários à Construção do Discurso do Preposto

Muitas das informações necessárias para a composição da abordagem de ajuda aqui proposta advêm da lógica de design, a qual não é fácil de ser capturada (Moran e Carroll, 1996). Na proposta original (Silveira et al., 2000), estas informações deveriam ser capturadas do designer durante o processo de design da

aplicação, através das perguntas vistas na seção 3.1. Estas perguntas eram, então, analisadas sob quatro perspectivas diferentes: o ponto de vista do Domínio, da Tarefa, do Agente e da Aplicação, as quais definiam o modelo então proposto (figura 4.1).

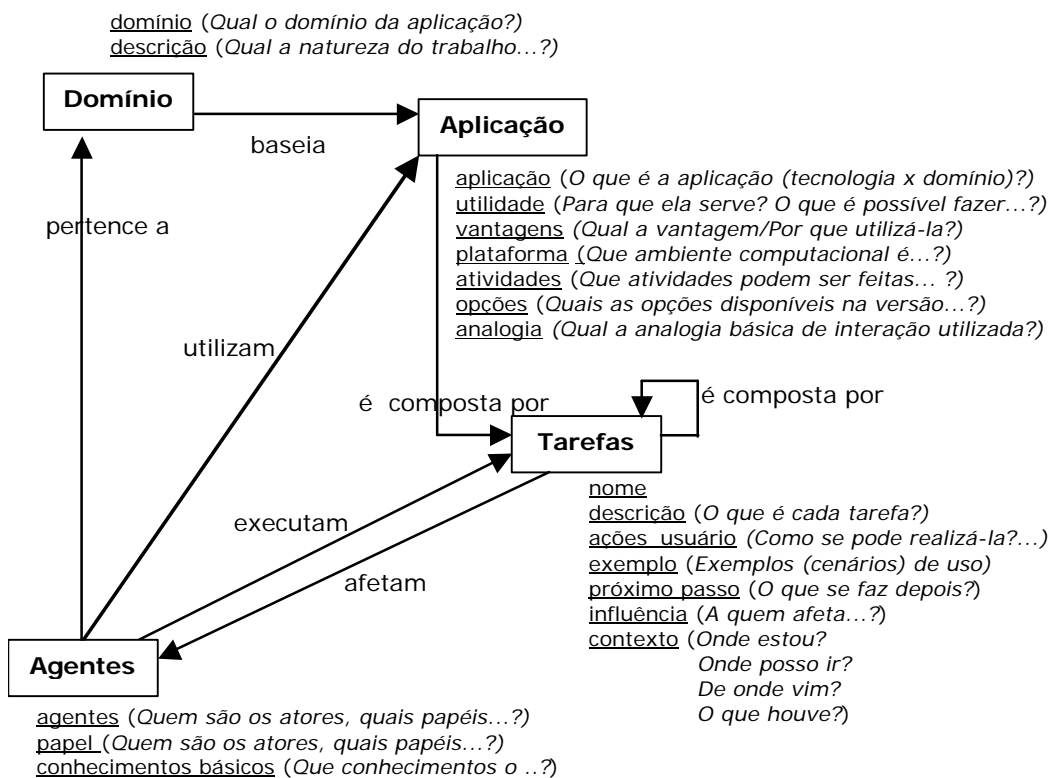


Figura 4.1: Modelo de ajuda.

Este modelo (figura 4.1) foi refinado posteriormente, para uma melhor distribuição das tarefas e ações correspondentes (figura 4.2).

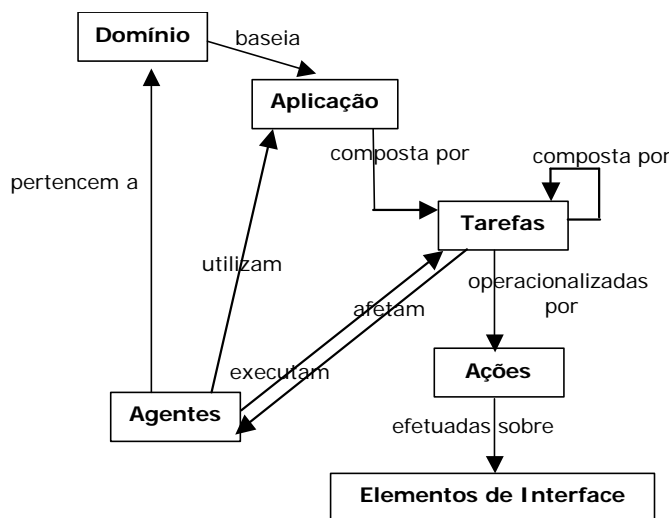


Figura 4.2: Modelo de ajuda estendido.

Dentro do processo de design de IHC, são utilizados diversos modelos que permitem representar soluções de interação de forma a apoiar a reflexão e a tomada de decisões sobre o design. Após contrastar-se o modelo de ajuda proposto com estes diferentes modelos de design de IHC, verificou-se que uma parte das informações necessárias à construção da ajuda já encontrava-se disponível nos mesmos. Assim, no intuito de reduzir o trabalho do designer da ajuda⁶ e de aproveitar o trabalho realizado pelos demais membros da equipe de design, optou-se pela captura destas informações a partir dos demais modelos no lugar de construir-se mais um modelo, específico para a ajuda.

É grande a diversidade de modelos de design de IHC existentes. Os modelos de tarefa são os mais comumente utilizados (Moran, 1981; Card et al., 1983; Johnson et al., 1988; Scapin e Pierret-Golbreich, 1989; Payne e Green, 1989; Hix e Hartson, 1993; van der Veer et al., 1996; Paternò, 2000), mas encontram-se também referências a modelos de usuário, de domínio, de apresentação e de diálogo, dentre outros (Preece et al., 1994; Puerta, 1996 e 1997; Paternò, 2000).

Em (Puerta, 1996) ressalta-se que, através dos modelos de design, deveriam poder ser obtidas informações para responder a uma série de questões, que guiarão o trabalho da equipe durante todo o processo. Dentre estas questões, estão:

- Quem são os usuários da interface?
- Que tarefas os usuários executam usando a interface?
- Que objetos de domínio são necessários na interface?
- Como os componentes da interface são apresentados a cada usuário?
- Que comandos e ações o usuário pode executar na interface?

Estas informações são essenciais para o design da aplicação, mas não suficientes para o design do sistema de ajuda *online* da aplicação, seguindo a abordagem aqui apresentada. Além destas questões mais funcionais, são fundamentais para o conteúdo da ajuda informações que cubram questões como:

⁶ Neste trabalho, sempre que é usado o termo “designer da ajuda” este refere-se ao designer (ou equipe de designers) responsável(is) pelo desenvolvimento do sistema de ajuda.

- Qual a natureza do trabalho no domínio da aplicação?
- Qual a projeção da tecnologia sobre este domínio?
- Por que é necessário realizar esta ou aquela tarefa ou ação? A quem isto irá afetar? De quem isto depende?
- Por que determinado caminho de interação é preferencial a outro?

Tendo em vista a possibilidade de reutilizarem-se as informações que já estivessem disponíveis, bem como de suprir a necessidade das informações específicas à ajuda, foi feito um estudo aprofundado dos modelos de design encontrados. A partir deste estudo foi definido um conjunto de modelos a serem utilizados, os quais foram estendidos a fim de conter as informações específicas para a ajuda. Este conjunto é composto pelos modelos de domínio, de aplicação, de tarefas, de usuário, de interação e de interface, que serão detalhados adiante. O modelo de ajuda discutido foi eliminado e as informações necessárias ao mesmo passaram a ser obtidas a partir deste novo conjunto (figura 4.3).

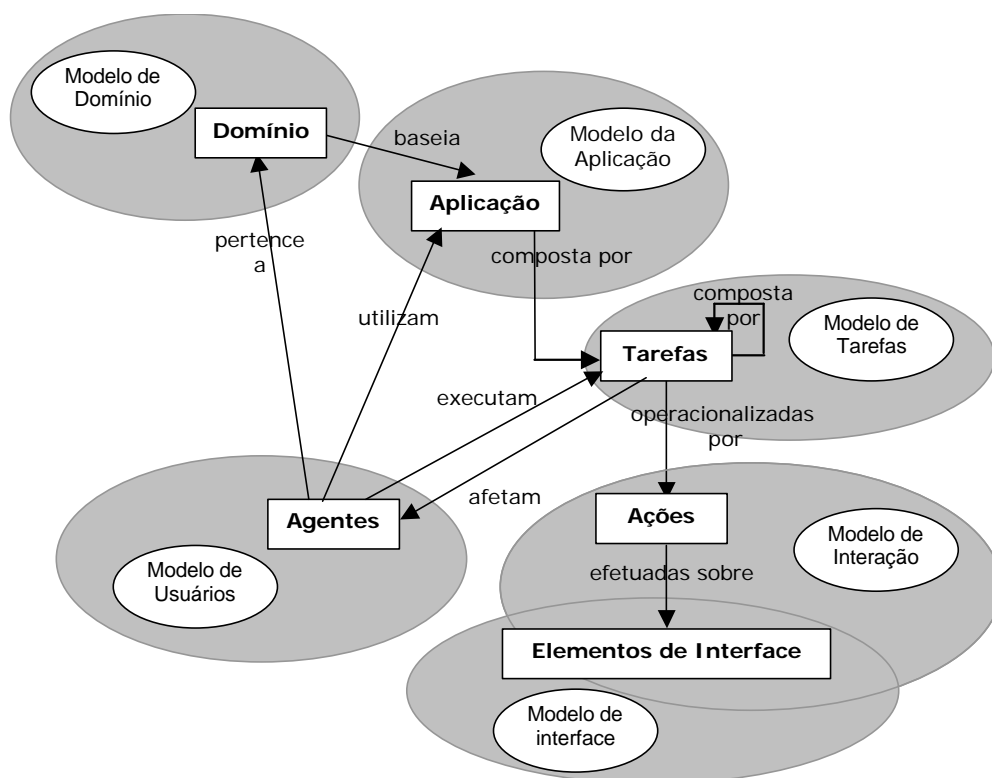


Figura 4.3: Modelos de design propostos e as informações de ajuda.

A criação destes modelos é feita durante períodos distintos do processo de design de IHC. O diagrama da figura 4.4 apresenta um esquema genérico dos períodos em que cada um destes modelos é elaborado.

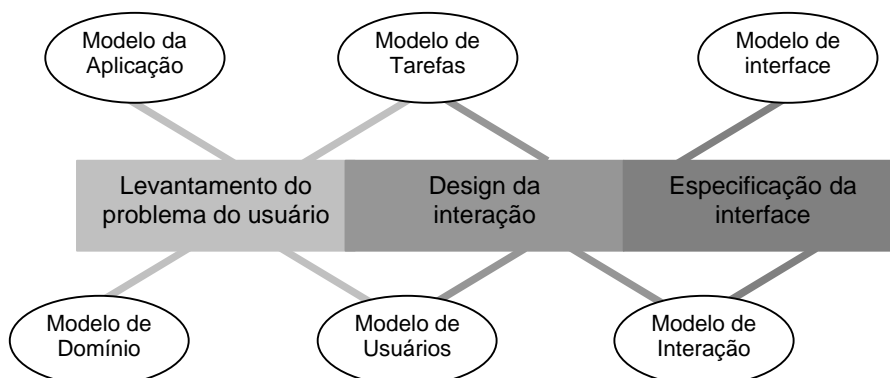


Figura 4.4: Períodos de criação dos modelos de design de IHC.

Defende-se, aqui, que a construção destes modelos seja feita durante o processo de design da aplicação (Silveira et al., 2000), para que as informações necessárias possam ser capturadas no momento em que são geradas, não sendo necessário capturá-las novamente, no período final de desenvolvimento, quando a maioria dos sistemas de ajuda costumam ser elaborados. Com o uso dos modelos propostos, e sua criação nos períodos descritos, estas informações não apenas servem de insumo às discussões/elaborações da própria equipe de design de IHC, mas também aos responsáveis pelo design do sistema de ajuda, que já o podem ir desenvolvendo *pari passu* ao design e desenvolvimento da aplicação. Assim, ao final do desenvolvimento da aplicação já é possível ter, também, boa parte do sistema de ajuda desenvolvido, restando apenas ajustes finais ao mesmo.

Cabe ressaltar que a maior parte destes modelos (modelos de domínio, de aplicação, de tarefas, de usuário e parte do modelo de interação) está desvinculada de qualquer tecnologia específica. Já a porção do modelo de interação que trata da operacionalização das tarefas e o modelo de interface, estes, sim, irão depender de como a aplicação está sendo especificada, ou seja, da tecnologia empregada e do estilo de interface a ser implementado.

A seguir serão listados os modelos utilizados, explicitando-se apenas a parte deles que é relevante para a construção do discurso do preposto. Estes modelos não precisam, necessariamente, ser especificados na notação sugerida a seguir. O

importante é que as informações contidas nos mesmos garantam a composição das respostas às solicitações de ajuda propostas.

Para fins de ilustração, junto à notação de cada modelo será apresentada parte de sua especificação feita para o estudo de caso⁷ realizado, o qual será detalhado na seção 4.3.

Modelo de Domínio

Este modelo contém informações relativas ao domínio da aplicação, com foco na definição do domínio, da natureza do trabalho no mesmo e, também, no detalhamento dos signos (informações apresentadas ou manipuladas) de domínio utilizados.

```
<modelo de domínio> ::=  
  <descrição>  
  <natureza>  
  {<signo de domínio>}  
  
<signo de domínio> ::=  
  <nome>  
  <descrição>  
  <utilidade>
```

```
...  
SIGNO DE DOMÍNIO  
{  
  NOME(Seção)  
  DESCRIÇÃO(Uma seção é, como em um  
  jornal ou revista, o lugar onde os  
  avisos são agrupados de acordo com  
  seu assunto.)  
  UTILIDADE(Para classificar os avisos  
  que têm características comuns,  
  facilitando sua localização. Por  
  exemplo, avisos sobre eventos  
  encontram-se na seção Eventos, sobre  
  voluntários, na seção Voluntários e  
  assim sucessivamente.)  
}  
...
```

Modelo da Aplicação

Este modelo contém informações relativas à aplicação (a ser) implementada. Aqui o foco está na descrição da aplicação, sua utilidade, vantagens, atividades e opções possíveis, e nos signos desta aplicação.

⁷ Este estudo de caso, conforme será detalhado na seção apropriada, trata de um ambiente de apoio ao trabalho de voluntários em uma organização não governamental, que está sendo desenvolvido atualmente (ORÉ, 2002). O módulo deste ambiente que será utilizado como exemplo é o Quadro de Avisos. Em termos gerais, este módulo permite que os voluntários e funcionários da organização possam postar e verificar avisos referentes a seu trabalho e/ou a organização como um todo. Estes avisos estarão disponíveis para toda a comunidade (membros ou não da organização), através da Internet.

```

<modelo de aplicação> ::=
  <descrição>
  <utilidade>
  <vantagens>
  <atividades>
  <opções>
  {<signo de aplicação>}

<signo de aplicação> ::=
  <nome>
  <descrição>
  <utilidade>

```

```

...
SIGNO DE APLICAÇÃO
{
  NOME (Indicador da Seção Atual)
  DESCRIÇÃO(Este é o indicador de qual
  é a seção atual (seção Quadro Geral
  ou seção Como ajudar, por exemplo).
  Uma seção pode ser pública ou
  privativa da comunidade.)
  UTILIDADE(Para indicar qual a seção
  que está sendo apresentada no
  momento.)
}
...

```

Modelo de Tarefas

Este modelo contém informações relativas às tarefas que os usuários podem realizar. Para cada tarefa possível, é relacionada a descrição da mesma, sua utilidade, o motivo pelo qual ela deve ser realizada (ponto de vista do designer), de qual tarefa-mãe ela provém, o operador que a conecta a próxima tarefa, estabelecendo, assim, a forma de execução da mesma, as condições necessárias para sua execução e os signos de domínio e de aplicação relacionados a ela.

A definição dos operadores baseia-se em (Paternò, 2000): seqüência (necessidade de execução de todas as tarefas especificadas, na ordem indicada), alternativa (necessidade de execução de uma das tarefas especificadas) e independente (necessidade de execução de todas as tarefas especificadas, independente da ordem em que sejam executadas).

```

<modelo de tarefas> ::=
  <tarefa>{<tarefa>}

<tarefa> ::=
  <nome>
  <tarefa-mãe>
  [<pré-condição>]
  [<descrição>]
  [<utilidade>]
  [<motivo>]
  <operador>
  <seqüência>
  <signo de domínio>
  <signo de aplicação>

<signo de domínio> ::=
  {<nome>}

<signo de aplicação> ::=
  {<nome>}

```

```

...
TAREFA
{
  NOME(Fornecer os dados necessários)
  TAREFA-MÃE(Criar um aviso)
  OPERADOR(seqüência)
  SEQÜÊNCIA(1)
  SIGNO DE DOMÍNIO(autor título data
  seção chamada aviso prazo)
  SIGNO DE APLICAÇÃO(quadro_geral
  login)
}
TAREFA
{
  NOME(Incluir o aviso)
  TAREFA-MÃE(Criar um aviso)
  OPERADOR(seqüência)
  SEQÜÊNCIA(2)
  SIGNO DE DOMÍNIO()
  SIGNO DE APLICAÇÃO()
}
...

```


Modelo de Usuários

Este modelo contém informações relativas aos usuários da aplicação. Nele encontram-se os papéis e perfis existentes na aplicação. A cada papel é relacionado seu nome, as tarefas que podem ser executadas por este e quais os usuários relacionados. Já o perfil identifica a forma como o usuário prefere ou pode interagir com a aplicação, bem como os usuários relacionados.

```
<modelo de usuários> ::=
  <papel>{<papel>}
  <perfil>{<perfil>}

<papel> ::=
  <nome>
  < tarefa>
  <usuário>

<tarefa> ::=
  <nome>{<nome>}

<usuário> ::=
  <nome>{<nome>}

<perfil> ::=
  <nome>
  <descrição>
  <usuário>{<usuário>}
```

```
...
PAPEL
{
  NOME(Membro)
  TAREFA(Consultar avisos
        Imprimir avisos
        Gerenciar avisos)
  USUÁRIO(Antonio Carlos Melo
          João da Silva
          Maria Aparecida Lisboa
          ...
          )
}
PAPEL
{
  NOME(Visitante)
  TAREFA(Consultar avisos
        Imprimir avisos
        Solicitar inscrição)
  ...
}
```

Modelo de Interação

Este modelo contém informações sobre as formas de interação possíveis na aplicação, ou seja, sobre como efetivamente se dá a *conversa* para realizar uma determinada tarefa na aplicação. Para cada tarefa há as alternativas de operacionalização possíveis para sua execução, e, para cada alternativa, há o motivo pelo qual esta alternativa deve ser executada em relação a outras (do ponto de vista do designer), a condição para sua execução, a indicação se esta é a forma preferencial de execução da tarefa (do ponto de vista do designer) e a seqüência de ações necessárias para sua execução. Para cada ação, há o valor *default* da mesma bem como a forma de desfazê-la, além do operador que a conecta à próxima ação.

```

<modelo de interação> ::=
  { <alternativas de
    operacionalização> }

<alternativas de
operacionalização> ::=
  < tarefa >
  < operacionalização >
  { < operacionalização > }

< tarefa > ::=
  < nome >

< operacionalização > ::=
  < nome >
  [< motivo >]
  [< condição >]
  < preferência do designer >
  < ação > { < ação > }

< ação > ::=
  < nome >
  < operador >
  < seqüência >
  [< valor default >]
  [< forma de desfazer >]

```

```

...
ALTERNATIVAS DE OPERACIONALIZAÇÃO
{
  TAREFA
  {
    NOME(Fornecer os dados necessários)
    {
      OPERACIONALIZAÇÃO
      {
        NOME(Default)
        PREFERÊNCIA DO DESIGNER(sim)
        AÇÃO
        {
          NOME(inserir o autor)
          OPERADOR(independente)
          SEQÜÊNCIA(1)
        }
        AÇÃO
        {
          NOME(inserir o título)
          OPERADOR(independente)
          SEQÜÊNCIA(2)
        }
        AÇÃO
        {
          NOME(inserir a data)
          OPERADOR(independente)
          SEQÜÊNCIA(3)
        }
      }
    }
  }
}
...

```

Modelo de Interface

Este modelo contém informações sobre os elementos de interface da aplicação. Para cada elemento, encontra-se seu tipo, os valores que ele pode assumir, seu valor *default*, sua localização na interface e os signos de domínio e aplicação relacionados.

```

<modelo de interface> ::=
  < elemento > { < elementos > }

< elemento > ::=
  < nome >
  < tipo >
  [< valor possível >]
  [< valor default >]
  < localização >
  < signo de domínio >
  < signo da aplicação >

< signo de domínio > ::=
  { < nome > }

< signo de aplicação > ::=
  { < nome > }

```

```

...
ELEMENTO
{
  NOME(texto do aviso)
  TIPO(caixa de texto)
  LOCALIZAÇÃO(tela 'Criar Aviso')
  SIGNO DE DOMÍNIO (texto)
}
ELEMENTO
{
  NOME(seção)
  TIPO(lista)
  LOCALIZAÇÃO(tela 'Criar Aviso')
  SIGNO DE DOMÍNIO (seção)
}
...

```

4.2

Concretização do Discurso do Preposto via uma Arquitetura de Sistemas de Ajuda

O usuário pode especificar suas dúvidas, no momento da interação, através do uso das expressões de acesso à ajuda e, também, “re”questionar o preposto – a partir das respostas obtidas – a fim de aprofundar as informações recebidas. Esta visão comunicativa do sistema de ajuda pode ser melhor compreendida através do exemplo de uma arquitetura construída (e implementada) sob esta perspectiva.

4.2.1

Uma Arquitetura Proposta para Sistemas de Ajuda *Online*

A arquitetura construída contempla os seguintes componentes:

- ajudas locais: permitem acessar o conteúdo da ajuda sensível ao contexto atual de interação, através do uso das expressões de acesso à ajuda associadas a elementos de interação e da possibilidade de aprofundamento das respostas obtidas;
- módulo de ajuda geral: informações detalhadas a respeito da aplicação, do domínio no qual ela está inserida, de suas funcionalidades e de suas formas de operação, bem como uma visão discursiva da ajuda dialogada (ajuda local);
- instruções diretas: informações e avisos que não são ativados pelo usuário e, sim, pela própria aplicação, dando referências ou dicas sobre como este deve proceder em determinado momento; e,
- mensagens de erro: mensagens também ativadas pela aplicação, informando que determinada ação que o usuário tenha feito está incorreta ou incompleta, e como este deve fazer para efetua-la corretamente.

O foco desta seção está nos dois primeiros componentes da arquitetura: ajuda local e geral. As instruções diretas e mensagens de erro – apesar de fazerem parte do discurso do preposto e de serem uma forma de auxílio ao usuário – não serão detalhadas neste trabalho. Um avanço neste sentido pode ser visto em

(Barbosa et al., 2002b), onde propõe-se a especificação, durante o processo de modelagem de IHC, dos tipos de prevenção e tratamento de erro que podem ocorrer. São propostos seis tipos de prevenção e tratamento de erro: prevenção passiva; prevenção ativa; alerta; captura de erro; tratamento apoiado e tratamento automático.

4.2.1.1

Construindo o Discurso para esta Arquitetura: Ajuda Local

Para cada elemento de interação (signo, tarefa, seqüência de ações e ação) estão relacionadas diferentes expressões, de acordo com a tabela abaixo (tabela 4.1).

Elemento	Expressão de acesso à ajuda relacionada
signo de domínio ou signo de aplicação	O que é isto?
	Para que serve isto?
	Onde está?
tarefa	O que é isto?
	Para que serve isto?
	Como faço isto?
	Epa!
	E agora?
	O que aconteceu?
	A quem isto afeta?
	De quem isto depende?
	Quem pode fazer isto?
Onde eu estava?	
seqüência de ações	Por que devo fazer isto?
	Existe outra maneira de fazer isto?
ação	Como faço isto?
	Epa!
	E agora?
	O que aconteceu?
	Onde eu estava?

Tabela 4.1: Relação entre elementos e expressões.

Para cada uma destas expressões está relacionado um *template* de resposta, o qual é instanciado com informações dos modelos de design de IHC descritos na subseção anterior.

A relação entre as expressões e os modelos pode ser vista nos *templates* de composição de respostas de cada uma delas. Para fins de ilustração, junto a alguns dos *templates* serão apresentados exemplos, também parte do estudo de caso a ser detalhado a seguir.

O que é isto?

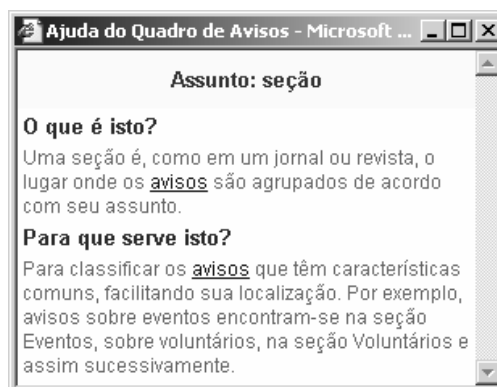
template: descrição(**Signo**)

busca a **descrição**
do Signo de Domínio, no Modelo de Domínio
ou
do Signo de Aplicação, no Modelo de Aplicação,
conforme indicado

```

...
SIGNO DE DOMÍNIO
{
  NOME(Seção)
  DESCRIÇÃO(Uma seção é, como em um
jornal ou revista, o lugar onde os
avisos são agrupados de acordo com
seu assunto.)
  UTILIDADE(Para classificar os avisos
que têm características comuns,
facilitando sua localização. Por
exemplo, avisos sobre eventos
encontram-se na seção Eventos, sobre
voluntários, na seção Voluntários e
assim sucessivamente.)
}
...

```

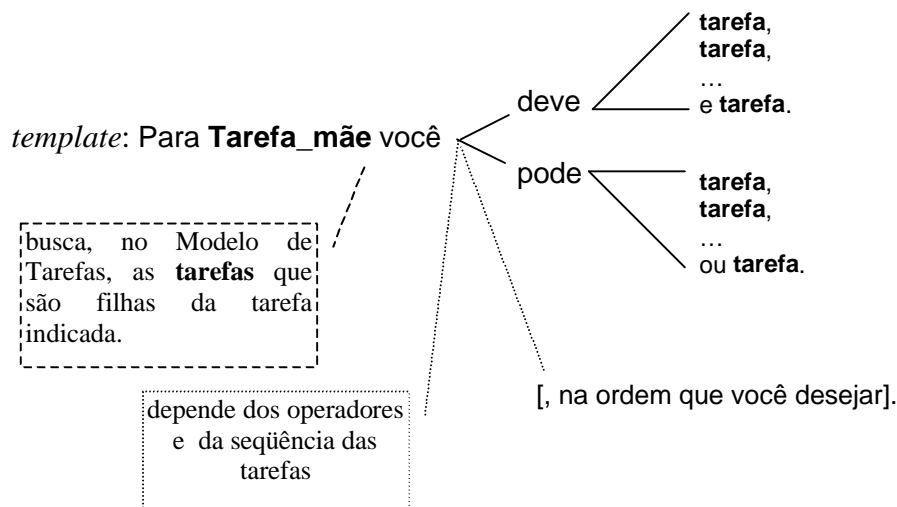


Para que serve isto?

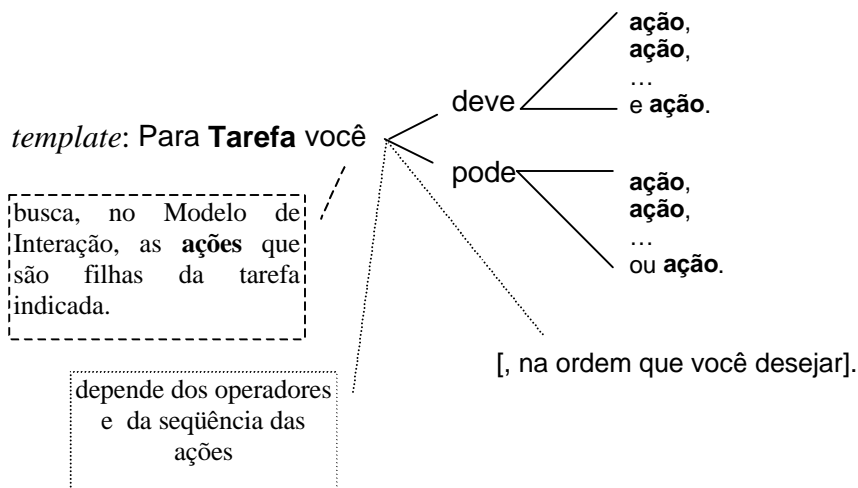
template: utilidade(**Item**)

busca a **utilidade**
do Signo de Domínio, no Modelo de Domínio
ou
do Signo de Aplicação, no Modelo de Aplicação
ou
da Tarefa, no Modelo de Tarefas,
conforme indicado

Como faço isto?

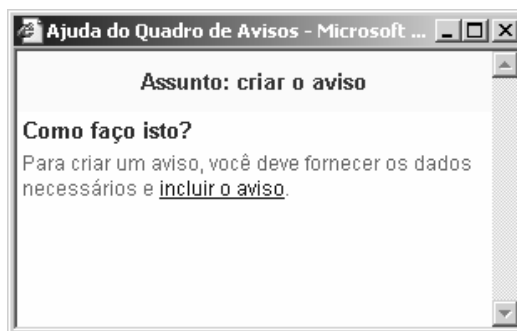


ou



```

...
TAREFA
{
  NOME(Fornecer os dados necessários)
  TAREFA-MÃE(Criar um aviso)
  OPERADOR(seqüência)
  SEQÜÊNCIA(1)
  SIGNO DE DOMÍNIO(autor título data
    seção chamada aviso prazo)
  SIGNO DE APLICAÇÃO(quadro_geral
    login)
}
TAREFA
{
  NOME(Incluir o aviso)
  TAREFA-MÃE(Criar um aviso)
  OPERADOR(seqüência)
  SEQÜÊNCIA(2)
}
...
    
```



Por que devo fazer isto?

template: motivo(**Operacionalização**)

busca o **motivo** da Operacionalização escolhida, no Modelo de Interação

Existe outra maneira de fazer isto?

template: Uma forma alternativa de realizar esta mesma tarefa é
[, na ordem que você desejar].

depende dos operadores e da seqüência das ações

busca as **ações** de uma Operacionalização alternativa, no Modelo de Interação.

ação,
ação,
...
e
ação.

ação,
ação,
...
ou
ação

Epa!

template: Se você quiser desfazer sua última tarefa você pode **forma de desfazer.**

busca a **forma de desfazer** da Tarefa, no Modelo de Tarefas

Onde está?

template: localização(**elemento**)

busca a **localização** do elemento solicitado no Modelo de Interface

E agora?

template: nome (**tarefa**)
ou nome (**ação**)

busca
a próxima **tarefa** no Modelo de Tarefas
ou
a próxima **ação** no Modelo de Interação
conforme o status atual

A quem isto afeta?

template: nome (**papel**)

busca a próxima **tarefa** no Modelo de Tarefas, e o **papel** relacionado à mesma, no Modelo de Aplicação, conforme o status atual

De quem isto depende?

template: nome (**papel**)

busca a **tarefa** anterior no Modelo de Tarefas, e o **papel** relacionado à mesma, no Modelo de Aplicação, conforme o status atual

Quem pode fazer isto?

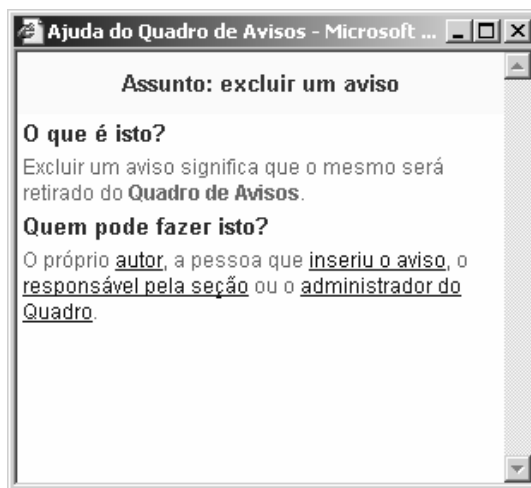
template: nome (**papel**)

busca a **tarefa** atual no Modelo de Tarefas, e o **papel** relacionado à mesma, no Modelo de Aplicação, conforme o status atual

PUC-Rio - Certificação Digital Nº 9916211/CA

```

...
PAPEL
{
  NOME(Membro)
  TAREFA(Consultar avisos
        Imprimir avisos
        Gerenciar avisos)
  USUÁRIO(Antonio Carlos Melo
          João da Silva
          Maria Aparecida Lisboa
          ...
          )
}
PAPEL
{
  NOME(Visitante)
  TAREFA(Consultar avisos
        Imprimir avisos
        Solicitar inscrição)
  ...
}
    
```



Onde eu estava?

template: nome (**tarefa**)
ou nome (**ação**)

busca
a **tarefa** anterior no Modelo de Tarefas
ou
a **ação** anterior no Modelo de Interação
conforme o status atual

Cabe ressaltar que em algumas respostas (as que tratam diretamente de elementos de interface, como as relacionadas às expressões “O que é isto?” ou “Onde está?”, por exemplo), a dúvida do usuário, ao usar a expressão, é mais facilmente detectável porque o usuário está indicando, explicitamente, a que elemento ele está se referindo (ou por apontamento do *mouse*, na expressão “O que é isto?”, ou pela informação direta, no caso do “Onde está?”). Já para uma expressão do tipo “E agora?”, por exemplo, a resposta a ser apresentada ao usuário é dada a partir da inferência do que o designer acredita que é a dúvida/problema do usuário no momento, já que não é explicitamente especificado a que o usuário está se referindo.

Estas respostas geradas automaticamente não são, ainda, as respostas definitivas a serem utilizadas pelo sistema de ajuda. Definidos os elementos que terão acesso direto à ajuda (e as expressões associadas aos mesmos), suas respostas servirão de base para que especialistas em comunicação técnica possam trabalhar, de forma a melhor comunicar a mensagem do designer aos usuários. Este refinamento das mensagens (e as possibilidades de recorrências a partir das mesmas) será detalhado quando o método para construção da ajuda for descrito.

4.2.1.2

Construindo o Discurso para esta Arquitetura: Ajuda Geral

O módulo de ajuda geral contém a ajuda discursiva completa da aplicação. Este é o espaço onde informações relativas ao domínio e à aplicação podem ser detalhadas. A natureza do trabalho no domínio, as atividades que nele podem ser realizadas e a utilidade e vantagens de utilizar-se a aplicação, por exemplo, são informações típicas deste módulo. Este tipo de informação não é disponibilizado via ajuda local, por não estar diretamente associado aos elementos de interação.

Além destas informações, advindas dos modelos de domínio e de aplicação, este módulo também contém as informações obtidas através do uso das expressões, sendo estas apresentadas agora de forma sequencial e com *links* de hipertexto entre elas. Estas informações podem ser descritas em maior

profundidade, utilizando-se cenários e exemplos de execução das tarefas e/ou ações possíveis.

Informações sobre problemas de conexão de rede que podem ocorrer durante a interação também são bastante úteis e podem, aqui, ser disponibilizadas, bem como dúvidas e ou problemas encontrados podem ser enviados, diretamente, à equipe de administração (ou desenvolvimento) da aplicação.

4.2.2 A Arquitetura Proposta e as Arquiteturas Tradicionais

As arquiteturas tradicionais de ajuda apresentam dicas, instruções passo-a-passo ou informações mais detalhadas, dependendo da aplicação e/ou da forma de acesso utilizada, conforme visto no capítulo 2.

Entretanto, o que mais as diferencia da arquitetura proposta é o teor do discurso da mesma e a possibilidade de questionamento por parte do usuário.

A arquitetura aqui apresentada também oferece dicas (respostas típicas à uma expressão “O que é isto?”), instruções passo-a-passo (resposta à uma expressão “Como faço isto?”) e informações mais detalhadas (no caso, o conteúdo da “Ajuda Geral”). No entanto, há a preocupação primordial em manter-se o foco destas informações na lógica de design e nas tarefas a serem realizadas pelo usuário. Além disto, existem informações adicionais relacionadas à utilidade das mesmas (“Para que serve isto?”) e às motivações para realizá-las (“Por que devo fazer isto?” ou “Existe outra maneira de fazer isto?”), por exemplo.

Já em termos de acesso, ferramentas tradicionais como as de busca, através de índices ou palavras-chave, podem ser usadas junto a expressões como “Onde está?” ou “Como faço isto?”, por exemplo, para indicar o objeto de dúvida do usuário.

Além deste diferencial relativo ao teor da mensagem, há, também, a possibilidade de “continuar a conversa”. O usuário pode, a partir de uma resposta recebida, continuar a investigação de suas dúvidas, aprofundando determinados

pontos da informação recebida e, desta forma, aproximando-se, cada vez mais, da mensagem do designer – via preposto – como um todo.

4.3

Detalhando a Concretização do Discurso do Preposto via um Estudo de Caso

A fim de melhor ilustrar a arquitetura proposta e como se dá dada a concretização do discurso do preposto, será apresentado um estudo de caso com sua implementação através de um método de construção de sistemas de ajuda *online* à luz da Engenharia Semiótica.

4.3.1

O Quadro de Avisos

A aplicação que teve seu sistema de ajuda desenvolvido através deste método destina-se a apoiar o trabalho voluntário em uma organização não governamental (ORÉ, 2002). O módulo desta aplicação que será utilizado como exemplo é o Quadro de Avisos (figura 4.5).

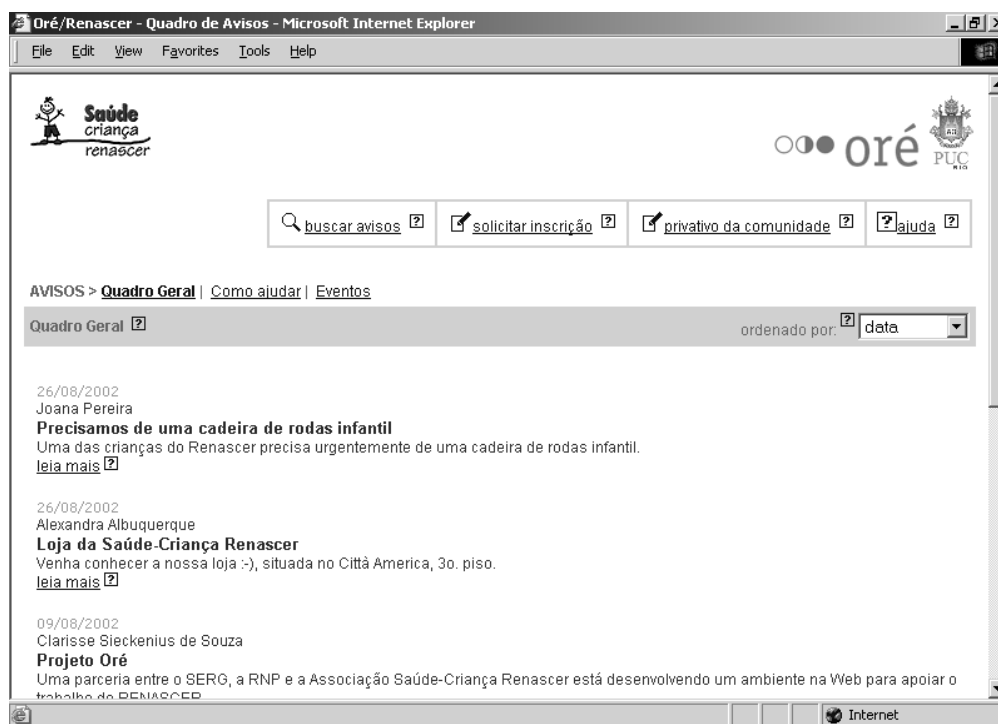


Figura 4.5: Página principal do Quadro de Avisos.

Como o próprio nome indica este módulo é utilizado como quadro de avisos virtual da comunidade de membros desta organização não-governamental. Através do quadro, a comunidade apresenta à sociedade em geral, via Internet, avisos relacionados ao seu trabalho e à organização como um todo, os quais podem ser lidos por qualquer interessado.

Estes avisos podem ser classificados em diferentes seções, de acordo com diferentes tópicos ou divisões da organização. Avisos relacionados a parte administrativa da organização, por exemplo, encontram-se na seção “Administração”; avisos relacionados a pedidos e possibilidades de doações, na seção “Como ajudar”; avisos relacionados a festas e eventos públicos ocorridos (ou a ocorrer), na seção Eventos e, assim, sucessivamente. A pessoa que utilizar o quadro pode navegar pelas diferentes seções em busca dos avisos de seu interesse, buscá-los utilizando palavras relacionadas ao mesmo e, também, ordená-los a fim de facilitar sua busca.

Nem todas as seções do quadro estão disponíveis para toda a sociedade. Determinadas seções são de interesse apenas da comunidade (como as que apresentam avisos relativos a parte administrativa, por exemplo), sendo apresentadas em uma área privativa, visualizada somente pelos membros da organização que se cadastraram, obtendo, assim, permissão de acesso à área. Nesta área, além do acesso às seções privativas, os membros da comunidade podem, também, gerenciar seus avisos (incluir, alterar e excluir os avisos).

4.3.2

Método para Construção de Sistemas de Ajuda *Online* à Luz da Engenharia Semiótica

Os passos a serem seguidos neste método são: construção dos modelos de design de IHC considerando-se questões específicas relacionadas à ajuda; geração de um rascunho do conteúdo da ajuda; refinamento do conteúdo e especificação dos pontos de recorrência; construção do módulo de ajuda geral; conexão dos pontos de acesso à ajuda aos elementos de interface; testes preliminares; análise de uso durante a avaliação de comunicabilidade e refinamento ou re-design do conteúdo da ajuda. Estes passos são ilustrados na figura 4.6.

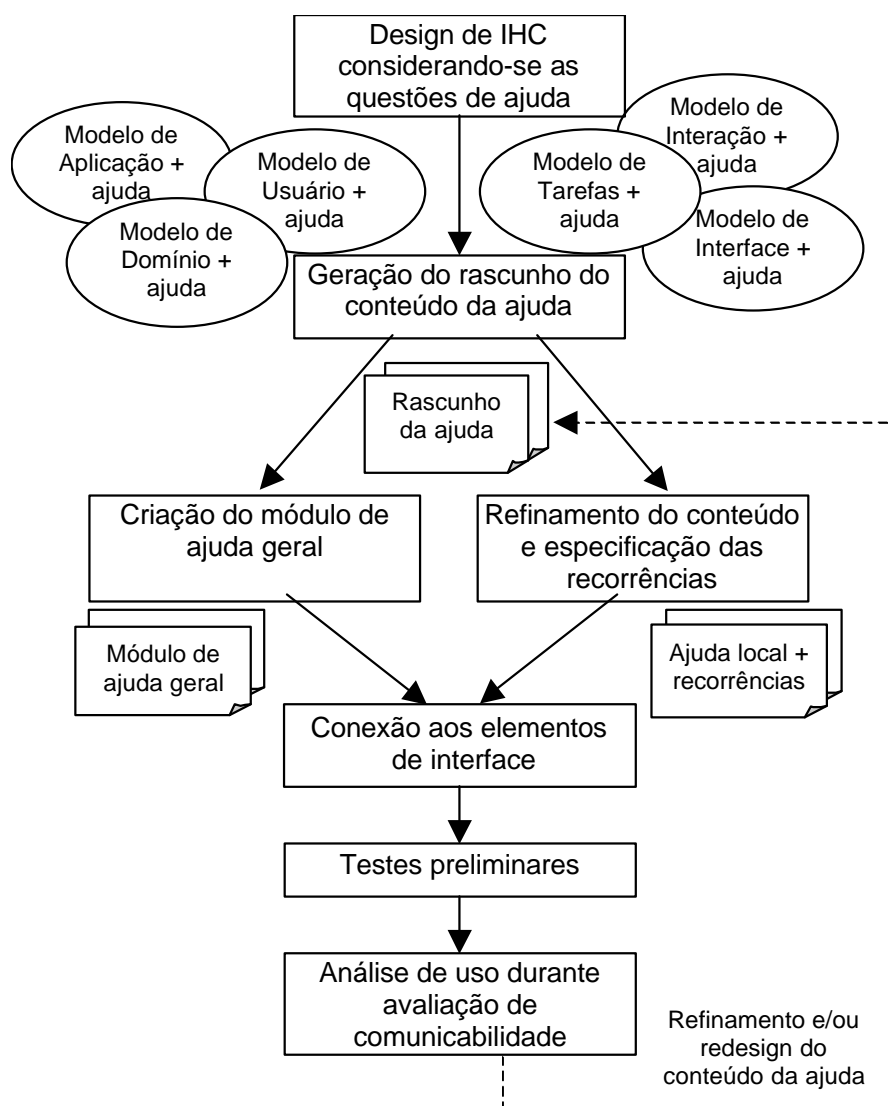


Figura 4.6: O método proposto para construção de sistemas de ajuda *online*.

Como ilustração dos passos seguidos no método, será utilizado o seguinte cenário, ilustrado na figura 4.7:

Joana, uma nova voluntária, foi informada de que agora existe um quadro de avisos virtual onde todos os membros da organização podem colocar avisos relativos a seus grupos. Para ela, usar um quadro deste tipo é fácil pois ela usa muito a Internet como fonte de pesquisa para seus trabalhos de faculdade e como meio de contato com seus amigos que moram fora do estado. Disseram a ela que além de ser muito fácil usar o quadro, se ela tivesse alguma dúvida era só clicar no ponto de interrogação azul, através do qual ela obteria ajuda imediatamente. Ela decide verificar o quadro para ver como ele funciona... Logo na entrada ela vê algumas mensagens sob o título “Destaques”. “Uhm, existe um pequeno ponto de interrogação aqui”, nota Joana. Ela decide ver a explicação e aprende que aquele título (“Destaques”) indica qual é a seção corrente. Como a palavra “seção” estava destacada, ela decide clicar nela para ver o que acontece. “Legal, aqui explica que as seções são utilizadas para podermos

separar os avisos que são relacionados!”. Assim ela vai diretamente a seção eventos para verificar quais serão os próximos eventos da organização...



Figura 4.7: Acessos à ajuda e respostas relacionadas.

Design de IHC

O primeiro passo a ser efetuado no método de construção de ajuda *online* é o uso dos modelos de design propostos para o design da aplicação. Neste momento é enfatizada a necessidade do designer da ajuda participar durante todo o processo de design a fim de explicitar a necessidade destas informações específicas de ajuda e capturá-las em tempo de design, para que as mesmas não sejam perdidas.

Na aplicação usada como ilustração de utilização deste método, o primeiro passo realizado foi a análise de domínio e de usuário, através de entrevistas abertas com membros representativos da organização (Barbosa et al., 2002a). Depois da análise ter sido concluída, foram construídos os cenários, os modelos de tarefa e de interação e os *storyboards*, segundo a abordagem proposta em (Barbosa et al., 2002b). E, em paralelo a isto, o designer da ajuda instanciou as informações coletadas a partir destas análises e modelagem nos modelos descritos na seção 4.1.

O designer da ajuda foi um membro ativo durante todo o processo de design. Sua presença foi essencial para assegurar a captura das informações específicas de ajuda necessárias e armazenar estas informações na base de dados correspondente, que, no final do processo, continha todas as informações necessárias para a geração do rascunho do conteúdo da ajuda. Desta forma, o processo de design da ajuda foi mais eficiente e deu melhores resultados, dado que a informação não refletia somente o produto do design, mas, incluía também a lógica de design por trás de decisões relevantes. Se estas informações não tivessem sido capturadas durante o processo de design, muitas delas poderiam ter sido perdidas ou esquecidas.

Como ilustração deste passo podem ser vistas as porções dos modelos de design exemplificadas na seção 4.1.

Geração do Conteúdo da Ajuda

Conforme visto no capítulo anterior, uma resposta minimalista é projetada para cada combinação elemento-expressão. A partir das informações contidas nos modelos de design de IHC e dos *templates* de resposta para cada expressão (subseção 4.2.1.1), um rascunho das respostas de ajuda é gerado para cada par elemento-expressão.

No estudo de caso em questão, o rascunho das respostas de ajuda foi gerado automaticamente das informações contidas na base de dados derivada dos modelos de design. A partir desta geração, o designer da ajuda selecionou, então, quais respostas deveriam realmente ser incluídas na aplicação.

Considere-se o indicador da seção corrente, “Destaques”. Este indicador é um signo de domínio e as expressões relacionadas a signos (tanto de domínio quanto de aplicação) são: “O que é isto?”, “Para que serve isto?” e “Onde está?”. Tomando como exemplo a expressão “O que é isto?” e usando o *template* mencionado anteriormente, o elemento descrição é buscado do componente de signos de domínio, do modelo de domínio, resultando no seguinte rascunho de resposta:

Este é o indicador de qual é a seção atual (seção Quadro geral, ou seção Como ajudar, por exemplo).

Outro exemplo, este relacionado ao modelo de tarefas, seria a resposta para a expressão “Como faço isto?”, relacionada à tarefa “Criar um aviso”. Utilizando o *template* indicado, e buscando as tarefas que têm “Criar um aviso” como tarefa mãe, o rascunho resultante seria:

Para criar um aviso, você deve fornecer os dados necessários e incluir o aviso.

Refinamento do Conteúdo e Especificação das Recorrências

Tendo sido gerado o rascunho das respostas, o designer da ajuda necessita selecionar os elementos que – de seu ponto de vista (baseado em entrevistas com os usuários, análise de domínio, etc) – devem ter associadas respostas de ajuda locais (sensíveis ao contexto). Para cada elemento, o designer da ajuda deverá, também, selecionar uma ou mais das expressões relacionadas. Em outras palavras, ele selecionará aqueles elementos sobre os quais ele acredita que o usuário poderá ter dúvidas ou que ele gostaria de explicar ou descrever aos usuários, e as expressões que serão utilizadas para acessar este conteúdo de ajuda.

Logo que estes pares de elemento-expressão selecionados, os rascunhos de respostas gerados devem passar por um processo de refinamento, durante o qual especialistas em redação técnica trabalharão o texto, a fim de melhor comunicar a mensagem do designer aos usuários. Tendo feito isto, cada resposta de ajuda é analisada, a fim de verificar os possíveis pontos de recorrência que elas podem conter. Estes pontos indicam os elementos textuais da resposta que terão expressões e conteúdo de ajuda associados. Este conteúdo pode, por sua vez, conter pontos de recorrência adicionais e assim, sucessivamente, permitindo o aprofundamento das informações de ajuda sobre determinado tópico.

Este passo pode ser feito antes da interface ser implementada, baseado nos *storyboards* construídos durante o processo de design de IHC.

No estudo de caso, tendo sido gerado o rascunho das respostas para todas as combinações expressão-elemento, o primeiro passo de refinamento foi selecionar quais destas combinações seriam, de fato, incluídas no sistema.

No exemplo apresentado na figura 4.7, a expressão “Para que serve isto?” está também relacionada a um signo de domínio (no caso, “Indicador de seção atual”). Entretanto, foi decisão do designer não incluir esta expressão, porque, neste caso particular, sua resposta seria redundante com a resposta dada à expressão “O que é isto?”.

O rascunho do texto das respostas selecionadas foi então refinado e analisado mais uma vez, com o propósito de encontrar pontos de recorrência possíveis.

No exemplo trabalhado, na resposta à expressão “O que é isto?”:

Este é o indicador de qual é a seção atual (seção Quadro geral, ou seção Como ajudar, por exemplo).

o designer verificou referência a outro signo de domínio, no caso, “seção”. Ele selecionou esta palavra como um ponto de recorrência na resposta, e associou as expressões “O que é isto?” e “Para que serve isto?” a ele.

O *template* para a expressão “O que é isto?” de um signo de domínio é o seguinte:

descrição(**Signo**)

e o *template* para a expressão “Para que serve isto?” é:

motivo(**Signo**)

Então, a resposta à expressão “O que é isto?” é:

Uma seção é o lugar onde os avisos são agrupados de acordo com seu assunto.

E a resposta para a expressão “Para que serve isto?”:

Para classificar os avisos que têm características comuns, facilitando sua localização. Por exemplo, avisos sobre eventos encontram-se na seção Eventos, sobre voluntários, na seção Voluntários, e assim sucessivamente.

Estas respostas já faziam parte do rascunho inicial de respostas gerado, e foram, então, também refinadas e analisadas para inclusão de possíveis pontos de recorrência.

Construção do Módulo de Ajuda Geral

Basicamente, este módulo contém informações de ajuda sobre o domínio e a aplicação, e, também, explicações sobre como diferentes tipos de ajuda funcionam (geral e local). Além disso, contém informações sobre elementos de interface (e suas expressões correspondentes) e problemas frequentes encontrados fora do contexto da aplicação (por exemplo, problemas de conexão de rede ou falhas de energia durante uma tarefa).

As porções relativas a informações de domínio e de aplicação deste módulo podem ser construídas e refinadas *pari passu* à construção e refinamento do conteúdo da ajuda local. Parte do conteúdo de ajuda local gerado pode também ser incluída no módulo de ajuda geral.

O módulo de ajuda geral da aplicação usada como estudo de caso aparece em uma nova janela (figura 4.8). Assim o usuário pode ver ambas as janelas – aplicação e sistema de ajuda – simultaneamente.



Figura 4.8: Módulo de ajuda geral.

Cinco seções foram projetadas para este módulo:

- informações sobre os tipos de ajuda disponíveis;

- informações sobre o domínio no qual a aplicação está inserida;
- detalhes sobre a aplicação em si;
- uma versão completa das expressões e pontos de recorrência para ajuda local, apresentados seqüencialmente e com *links* de hipertexto;
- informações sobre problemas de conexão de rede que podem ocorrer durante a interação.

Conexão aos Elementos de Interface

Estando o conteúdo da ajuda local (expressões e respostas correspondentes) e também o módulo geral de ajuda bem definidos, estes podem ser disponibilizados na interface.

Para cada elemento escolhido pelo designer da ajuda para ter expressões de ajuda associadas, um *link* ao conteúdo de ajuda correspondente é adicionado. Este *link* oferece as expressões disponíveis para determinado elemento e – através destas – acesso às respostas correspondentes. Estas respostas já foram previamente geradas e refinadas e são embutidas na aplicação a fim de evitar retardos na execução. Mais do que isso, este refinamento prévio possibilita um refinamento manual do rascunho de resposta gerado, a fim de que o formato das respostas seja mais natural e sua comunicação mais eficiente.

Um designer gráfico deve trabalhar junto neste passo a fim de especificar a interface do módulo de ajuda geral, da ajuda local, e dos elementos de interface que representam os pontos de conexão à ajuda.

No estudo de caso, o designer gráfico criou um símbolo a ser disposto junto aos elementos de interface onde deseja-se disponibilizar acesso à ajuda local, para funcionar como um *link* para este acesso. O símbolo escolhido foi um ponto de interrogação com uma borda quadrangular (destacado na figura 4.9).

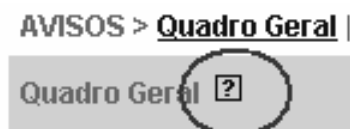


Figura 4.9: Link de acesso à ajuda local.

Tendo sido definidos os elementos que têm ajudas locais associadas, seus *links* foram implementados na interface.

As requisições de ajuda locais que podem estar disponíveis neste formato são aquelas relacionadas a signos e/ou tarefas. Os demais elementos são referenciados através dos pontos de recorrência encontrados dentro das respostas fornecidas.

Testes Preliminares

Estando construídos o protótipo da aplicação, o sistema de ajuda local e o módulo de ajuda geral, a equipe de design deve promover testes preliminares a fim de verificar se as expressões e suas respostas correspondentes e, também, as informações do módulo de ajuda geral, estão consistentes. Toda conexão de ajuda deve ser testada, bem como todos os pontos de recorrência das respostas.

No estudo de caso efetuado, foi feita a verificação de todos os acessos à ajuda – tanto através da interface quanto de recorrências – pelo designer da ajuda e outros membros da equipe, a fim de detectar erros possíveis. Alguns poucos problemas foram detectados, como falta de chamadas de ajuda locais ou inconsistências nos pontos de recorrência, os quais foram rapidamente corrigidos e testados novamente.

Análises mais aprofundadas foram também efetuadas, a fim de verificar se os textos da ajuda estavam ambíguos, levando o usuário a interpretações equivocadas.

Análise de Uso durante Avaliação de Comunicabilidade

Uma vez que a aplicação como um todo esteja implementada, podem ser efetuados testes com usuários reais através da avaliação de comunicabilidade (Prates et al., 2000) para verificar se a interface está transmitindo realmente a mensagem pretendida pelo designer e para investigar problemas que possam ocorrer durante a interação.

Além de analisar a interface como um todo, é possível aproveitar esta avaliação para analisar a interação específica com o sistema de ajuda. A fim de melhor observar a interação durante os testes e de ser capaz de capturar os problemas que possam ocorrer no uso da ajuda, é interessante que o designer da ajuda esteja presente durante as observações, para que ele tenha foco especificamente nestas questões. Ele pode não somente observar problemas no acesso às expressões de ajuda e na compreensão de suas respostas, mas também notar dificuldades dos usuários que ainda não tenham sido antecipadas (e portanto não tenham ajuda associada), ou aquelas respostas que não auxiliam o usuário em seu problema corrente.

Quando o primeiro protótipo da aplicação ficou pronto para teste, algumas sessões de avaliação de comunicabilidade foram efetuadas com seis usuários reais, selecionados especificamente por representarem o perfil geral da maioria da população alvo de usuários, como havia sido indicado pela análise das entrevistas. Dois deles tinham muito pouco conhecimento computacional, três deles eventualmente usavam computadores para propósitos pessoais e um deles tinha bastante experiência.

Do ponto de vista do designer este passo tornou possível determinar problemas na ajuda, os quais podem ser agrupados em três classes: conteúdo da ajuda, o declinar da ajuda e cultura de uso da ajuda.

Conteúdo da Ajuda

Durante os testes, o designer da ajuda observou que os usuários tinham problemas em situações não antecipadas por ele, o que significava que a ajuda correspondente era inadequada ou não tinha sido implementada. Não existiam, por exemplo, explicações sobre as diferenças entre seções públicas e privadas do Quadro de Avisos, ou sobre os critérios disponíveis para busca.

O Declinar da Ajuda

Uma questão interessante observada durante os testes ocorreu quando um usuário, depois de muitas tentativas infrutíferas em usar os mecanismos de busca da aplicação, fez uso da ajuda local. Ele leu a explicação e espontaneamente disse que entendeu o que deveria ser feito, mas mesmo assim decidiu fazer algo

diferente do que está detalhado na ajuda, o que acabou fazendo com que não conseguisse executar a tarefa definida no cenário de teste.

Cultura de Uso da Ajuda

O designer da ajuda também notou que, independente do grau de conhecimento que os usuários tivessem sobre o domínio da aplicação ou sobre o uso de computadores em geral, a maioria deles não acessou a ajuda. Não existe uma cultura de buscar por ajuda *online* quando acontece algum problema. Poucos foram aqueles que buscaram por ajuda e, quando o fizeram, parecia que tinham feito isso de forma aleatória, porque logo fechavam a janela de ajuda sem que tivesse havido tempo suficiente para ler as informações apresentadas. Ou, como no caso anterior, liam a informação mas não procediam como indicado. Um único usuário leu o texto da ajuda e seguiu a explicação sugerida.

Refinamento e/ou Redesign do Conteúdo de Ajuda

A partir dos resultados de teste, a interface pode ser refinada ou re-projetada. Isto pode causar a necessidade de re-projetar o conteúdo da ajuda. Além disto, testes podem indicar a necessidade de mudanças tanto no conteúdo da ajuda quanto nas conexões de interface. Depois destes ajustes, a aplicação deve passar por sessões de testes adicionais com usuários, e esta iteração continuará até que maiores problemas não sejam encontrados.

No estudo de caso, a partir das observações feitas durante os testes, as respostas de ajuda necessárias foram refinadas, o que resultou, também, no refinamento dos modelos de design envolvidos. Por exemplo, a resposta de ajuda para o signo de domínio “seção” foi ajustada, e uma nova resposta de ajuda foi criada para responder aos problemas descritos em **Conteúdo da Ajuda**, sobre seções públicas e privadas (figura 4.10).

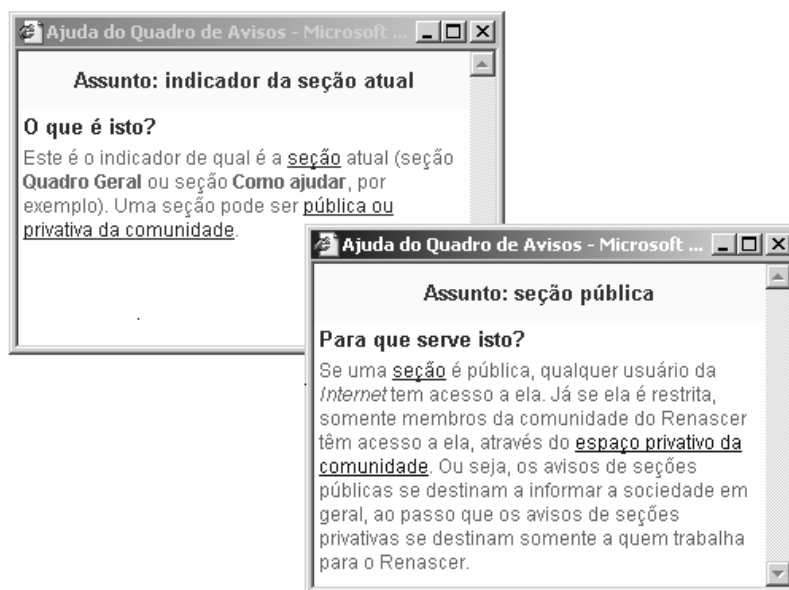


Figura 4.10: Respostas de ajuda revisadas e novas.

4.4

Considerações Gerais: capítulo atual e sua relação com os demais

Este capítulo apresentou a concretização do discurso do preposto do designer através da captura dos elementos necessários à sua construção, a concretização do mesmo via uma proposta de arquitetura para sistemas de ajuda *online*, a comparação desta com as arquiteturas de ajuda tradicionais e seu detalhamento através de um estudo de caso, durante o qual foi desenvolvido e utilizado um método específico de apoio à construção de sistemas de ajuda.

Demais questões relativas às visões de designers (de software e de ajuda) e usuários finais sobre esta visão de ajuda proposta serão discutidas no próximo capítulo.