

6. ANÁLISE E MODELAGEM DE USUÁRIOS E DE TAREFAS

No design de sistemas as atividades de análise devem prover informações a respeito de todos os componentes do domínio da aplicação: os usuários, as suas tarefas e rotinas de trabalho, e as informações. Esta atividade é indispensável, pois é a partir das informações que são adquiridas que derivam-se os modelos do domínio do problema e a definição do sistema.

No desenvolvimento tradicional - centrado-no-sistema - a interface é projetada a partir de informações obtidas na análise de dados e funções do domínio. Desses modelos consegue-se obter a funcionalidade do sistema. Entretanto, pouca importância é dada a conhecer quem são os futuros usuários do sistema e que tarefa eles necessitam realizar. Sem estas informações a interface é construída apenas a partir das funções que o sistema implementa, independente delas serem necessárias ou não aos usuários.

No desenvolvimento centrado-no-usuário a interface é projetada com o objetivo de satisfazer as necessidades do usuário através da construção de um sistema que amplie as suas capacidades. Para isso é necessário conhecer quem são os usuários, quais as suas tarefas e quais são os seus problemas.

Neste capítulo apresentaremos uma introdução às técnicas de análise e modelagem de usuários e tarefas como forma de complementar as abordagens tradicionais da engenharia de software com informações que são essenciais para o design de interfaces de usuário no paradigma de sistemas centrados no usuário.

6.1 Técnicas de análise e modelagem

A literatura de engenharia de software [R. Pressman 91] apresenta diversos métodos, técnicas e ferramentas para análise. Esses métodos podem pertencer a três paradigmas: métodos estruturados, orientado-a-dados e orientado-a-objetos. Entretanto, essas metodologias são voltadas para o processo de construção do software e muitas vezes não consideram projeto de interface, não é necessário métodos tão elaborados. Nos limitaremos a apresentar técnicas necessárias à eliciação do conhecimento necessário ao projeto da interação usuário-sistema.

6.1.1 Técnicas de análise

Para conduzir o processo de análise, o analista deve utilizar algumas técnicas de comunicação para aquisição das informações necessárias. As três técnicas de comunicação mais utilizadas são a *observação*, a *entrevista* e os *encontros*.

Na *observação*, o analista obtém as informações no próprio ambiente onde a aplicação será utilizada. Ele deve observar o fluxo de trabalho, conversar com usuários, examinar documentos, etc.

Na *entrevista*, o analista tenta adquirir informações sobre o domínio através de sessões de questionamento e discussões com especialistas do domínio. Esses especialistas passa a ser o provedor de todas as informações sobre o domínio. Os especialistas podem ser futuros usuários do sistema, gerentes ou diretores. Nesse caso, técnicas mais rigorosas de aquisição de conhecimento são necessárias, uma vez que o usuário pode não conseguir expressar adequadamente todas as informações relevantes.

Os *encontros* consistem de jornadas onde um grupo de analistas, desenvolvedores e especialistas se reúnem para exposições e discussões sobre o domínio do problema. O encontro normalmente ocorre em um local neutro onde os interessados passam a trabalhar exclusivamente em função da análise e definição do sistema. O encontro é coordenado por um *mediador*, que é uma pessoa neutra com domínio da técnica e que determina quais as atividades devem ser realizadas.

Adiante apresentaremos uma técnica baseada em cenários que complementa o processo de observação com uma técnica para condução da entrevista e meetings para a avaliação final. Os cenários visam descrever narrativamente o domínio e podem ser conduzidos através de observações no contexto, entrevistas com especialistas e um meeting final, onde usuários, analista e projetistas validam o conhecimento adquirido.

6.1.2 Modelagem

Os modelos que descrevem as informações que foram analisadas devem ser precisos, não ambíguos e de fácil compreensão pelos clientes e usuários. Ele deve ainda estimular o debate entre analistas, de um lado, e clientes e usuários, do outro com o objetivo de levantar todas as informações sobre os requisitos do sistema.

Existem diferentes formas de modelar o domínio com vistas a descrever as informações necessárias para se especificar o sistema. Um modelo é descrito utilizando uma notação ou linguagem específica. Dependendo do propósito do modelo ou do objeto a ser modelado pode-se optar por modelos informais, semi-formais ou formais.

Os modelos informais podem ser descritos em linguagem natural (na língua do usuário) e são utilizados nas etapas iniciais da análise com o objetivo de descrever os principais requisitos. Os *cenários*, que serão discutidos mais adiante, permitem uma modelagem informal que estimula a investigação de maiores detalhes a respeito do domínio que serão necessários em etapas posteriores quando poderão ser utilizadas linguagens mais formais.

Os modelos semi-formais utilizam-se de uma notação gráfica com uma sintaxe precisa que permite a descrição dos elementos do domínio de maneira clara, abstrata e com pouca ambiguidade. A maioria das metodologias de análise utilizam notações semi-formais. Estes modelos permitem uma visão ampla do sistema e podem ser utilizadas para a descrição de informações (o Diagrama Entidade-Relacionamento, é um exemplo bastante comum), de funções ou processos do sistema (Diagrama de Fluxo de Dados) ou de tarefas do usuário (GOMS ou HTA).

Modelos formais requerem uma linguagem formalmente definida, isto é, com a semântica e sintaxe são definidas matematicamente. Modelos formais são bastante utilizados para a especificação formal da funcionalidade do sistema.

Análise e modelagem estão bastante ligados. Existe uma circularidade nessas duas tarefas. A análise só se consolida quando é possível descrevê-la através de algum modelo. Esse modelo é utilizado para a avaliação do processo de análise junto aos usuários. Essa avaliação muitas vezes leva a um novo processo de análise, ou a uma análise complementar.

6.1.3 Complementando a análise com um design inicial

O processo de análise pode, e muitas vezes deve, ser complementado, de maneira iterativa, por um design inicial. Esse design inicial consiste de decisões preliminares de como deverá ser a interface e o processo de interação. Esta interface pode ser descrita na forma de um *storyboard* ou de um *protótipo*.

O storyboarding, que será visto adiante, é uma ferramenta valiosa na descrição do design inicial. Ele consiste na elaboração e apresentação gráfica situada de aspectos específicos da interface, onde algumas opções estão sendo julgadas.

Também será visto adiante que certos tipos de protótipos podem ser usados para complementar informações no processo de aquisição de conhecimento (análise) a respeito do domínio do problema. Esses protótipos com função exploratória e de apresentação do modelo do domínio não são operacionais. Eles são constituídos de apresentações que possam descrever parte do modelo visual da interface e exemplos de uso, a fim de demonstrar os modos de interação com o sistema.

6.2 Cenários como técnica de análise e modelagem

A elicitação de requisitos sobre as tarefas e os usuários pode ser melhor realizada quando os analistas procuram descrever situações do processo de trabalho. Os métodos baseados em cenários consistem de uma coleção de narrativas de situações no domínio do problema que permitem a identificação de componentes de design. Assim, eles são um meio de representação de fácil compreensão para os usuários envolvidos (muitas vezes de formação bastante heterogênea) que podem avaliar, criticar e fazer sugestões, proporcionando a reorganização de componentes e tarefas do domínio. Isso ocorre porque quando se introduz novas tecnologias, parte do contexto de tarefa de uma organização é alterado. Nessa reengenharia, novas tarefas e práticas são incorporadas enquanto que outras são eliminadas.

6.2.1 A técnica: Questionamento sistemático [J.Carroll et al 94]

A descrição de informações do domínio através de narrativas só é efetivamente realizada se o processo de compreensão por parte dos analistas e projetista for realizado de maneira sistemática [J. Carroll et al. 94]. A técnica apresentada aqui pode ser utilizada, segundo os autores, tanto para métodos de análise e design orientado a objetos quanto para interação usuário-sistema.

O *questionamento sistemático* é uma técnica psico-linguística que permite a psicólogos e lingüistas examinar o conteúdo e a estrutura de informações contidas numa narrativa. Uma narrativa é um sumário de um conjunto de eventos e ações envolvendo agentes e objetos do mundo. Nem todas as informações integrantes do contexto são passadas através da narrativa. Muitas dessas informações são inferidas do conhecimento cultural de cada indivíduo. Outras, entretanto, precisam ser obtidas diretamente na fonte, isto é, junto ao autor da narrativa.

Essa técnica foi desenvolvida para se entender melhor o processo de compreensão de histórias em narrativas. O objetivo é compreender tudo o que envolve o contexto daquilo que está sendo passado na narrativa.

Aplicando essa técnica no processo de análise de domínios, os especialistas devem descrever em narrativas seu conhecimento do domínio. Entretanto esse conhecimento é muito mais abrangente. O questionamento sistemático permite obter

todo o conhecimento que está além do que foi comunicado na narrativa. Assim, analistas e projetista podem utilizar essa técnica para adquirir mais eficazmente conhecimento sobre o domínio e inferir objetos e interações que não estão descritos na narrativa. Isto ocorre usando-se cada sentença ou afirmação da narrativa como ponto de entrada na complexidade do problema.

Nessa técnica, cada questão é uma ponte entre uma idéia e outra. Uma resposta a uma questão sobre um componente da narrativa revela outras conexões que são críticas para o seu entendimento. Realizando, sistematica e exaustivamente muitos tipos de questões sobre componentes da narrativa e iterativamente fazendo perguntas sobre as respostas geradas, o analista elabora um mapa conceitual (rede de proposições) que representa as estruturas do conhecimento do domínio.

Os passos do método consistem de:

1. **Geração do cenário** - as narrativas que compõe o cenário devem ser decritas pelo especialista no domínio. O analista pode motivá-lo fazendo perguntas como num processo convencional de entrevista (questões de elicitación do cenário).
2. **Elaboração da rede de proposições** - as narrativas devem ser simplificadas e expressas através de proposições.
3. **Análise** - a partir das proposições pode-se determinar as tarefas (ações e objetos) e usuários (agentes das ações).
4. **Questionamento sistemático** - novas proposições podem ser elaboradas através de questões que são feitas sobre elementos das proposições anteriores, num processo iterativo.

O método permite ir além do necessário para o projeto da interface de usuário. Nas iterações iniciais obtem-se informações sobre agentes, interações e objetos abstratos do domínio. Usando a técnica, pode-se chegar a um conhecimento mais profundo do domínio que permite aos analistas e projetista a elaboração de modelos funcionais do sistema.

Exemplo

Considere o seguinte cenário sobre um caixa eletrônico.

Questão de elicitación do cenário:

Como posso sacar R\$ 100,00 do caixa eletrônico?

Cenário:

Primeiro ponha o seu cartão do banco no caixa eletrônico, digite a sua senha e pressione o botão "saque rápido". Depois pegue o dinheiro.

As duas sentenças do cenário acima contém quatro proposições:

CARTÃO -- entra -- CAIXA-ELETRÔNICO

CLIENTE -- digita -- SENHA

CLIENTE -- pressiona -- BOTÃO-DE-SAQUE-RÁPIDO

CLIENTE -- pega -- DINHEIRO

A partir dessas proposições, o analista determina que o *cartão* e o *cliente* são agentes de ações. Numa análise voltada para a elicitación de requisitos da interação, seria determinado como usuário do sistema, o *cliente*. As ações são portanto: *digita*, *pressiona* e *pega*. São objetos da interação a *senha*, o *botão* e o *dinheiro*. Outros

objetos são o *caixa eletrônico* e o *cartão*. É preciso determinar que tipo de objetos são esses. Uma outra dúvida é a respeito do cartão ser objeto ou agente.

Obviamente, como esse exemplo é bastante simples e a aplicação também é muito conhecida, parece desnecessário obter mais conhecimentos a respeito. Entretanto, como o objetivo aqui é exemplificar a técnica, mostraremos como pode-se questionar a respeito dessa aplicação.

O analista deve então realizar uma série de questões sobre as proposições. Nesse questionamento o analista precisa determinar qual o relacionamento entre a resposta e a proposição que originou a pergunta.

As questões da categoria *por que*, visam responder “razões” (metas) e “causas” a respeito de eventos da narrativa. As questões da categoria *como* oferecem maiores detalhes a respeito de determinados eventos, permitindo determinar sub-tarefas e maiores detalhes sobre a interação. Questões da categoria *o que* podem ser feitas sobre objetos, e revelam atributos e hierarquias de objetos. Perguntas de *verificação* podem ser feitas para se saber se as proposições estão sendo entendidas corretamente. As perguntas de verificação são as que têm resposta *sim/não*. Uma taxonomia mais completa ainda está sendo pesquisada pelos autores do método.

Continuando o exemplo anterior a tabela abaixo descreve uma seção de questionamento sistemático:

Questões “por que”

Por que o cartão entra no caixa eletrônico?

- ☐ Para iniciar. (evento consequente)
- ☐ E então a máquina saberá quem é o cliente. (estado consequente)

Por que o cliente digita sua senha?

- ☐ Para provar que ele é o usuário autorizado. (meta)

Por que o cliente pressiona o botão de saque rápido?

- ☐ Porque é o botão para saques de R\$ 100,00 (critério de discriminação).
- ☐ Para evitar a digitação (cenário alternativo).
- ☐ Porque ele está com pressa (causa)

Questões “como”

Como o cartão entra no caixa eletrônico?

- ☐ O cliente tira o cartão de sua bolsa e
- ☐ insere no local apropriado.

Como o cliente digita a senha?

- ☐ Pressionando os botões adequados.

Como o cliente pressiona o botão de saque rápido?

- ☐ Colocando o seu dedo nele.

Questões “o que”

O que é um botão de saque rápido?

- ☐ É um tipo de botão que se pode pressionar.

O que é um botão?

_ É um dispositivo de controle no painel do caixa eletrônico.

Observe que se o analista estivesse utilizando essa técnica para um método orientado a objetos, outras questões poderiam ser realizadas com outros objetivos de acordo com as necessidades do método, como, por exemplo, para determinar classes e hierarquia de classes.

6.2.2 Comentários sobre a técnica de cenários

Após os cenários estarem desenvolvidos, os analistas devem trabalhar em conjunto com os usuários para avaliá-los e refiná-los. Isto pode ser feito, por exemplo, colocando-se posters numa sala pela qual os usuários podem circular livremente e observar os diversos cenários. Cada cenário deve apresentar quadros (desenhos, gráficos, fotografias, etc.), usando storyboards por exemplo, e uma descrição narrativa da tarefa. Os usuários, munidos de papeis e lápis, podem fazer anotações (críticas e sugestões) e anexá-las a cada poster.

O resultado da modelagem através de cenários é uma base para a compreensão de quem são os usuários, quais as tarefas envolvidas e quais funções a interface e a aplicação devem provê, de maneira que, já se possa ter meios de garantir a usabilidade do sistema.

A idéia de cenários em análise não está necessariamente associada à técnica de questionamento sistemático. Os cenários são extremamente úteis para descrição do domínio. A técnica sistematiza o processo de compreensão do conhecimento adquirido.

É importante observar que esta técnica é bastante recente e ainda deve ser aperfeiçoada. Entretanto, a idéia se adequa perfeitamente a necessidade de se extrair conhecimento de cenários de uso. Os métodos em geral, e esse não deve fugir à regra, devem ser usados, não como uma camisa-de-força que limite o processo de análise, mas como ferramentas que orientem os analistas e aumentem sua capacidade intelectual.

6.2.3 Cenários e Storyboards

Uma técnica que está bastante relacionada com cenários, por permitir descrever situações de uso, é o *storyboarding*. Essa técnica envolve a especificação através de imagens que descrevam certas situações que estão sendo planejadas.

Os cenários se diferenciam dos storyboards por utilizar basicamente narrativas para a descrição dos casos de uso, enquanto que storyboards utiliza a mídia gráfica usando papel ou computadores. Cenários são mais adequados para análise, enquanto que os storyboards permite a elaboração de protótipos não operacionais para designs iniciais. A técnica de cenários é mais apurada e com fundamentação teórica (ver adiante). O storyboarding é usado de maneira mais livre.

6.3 Análise e modelagem de usuário

O usuário deve sempre ser o foco central de interesse do projetista ao longo do design da interface. O propósito dessa atividade é indentificar quem são os usuários

e caracterizá-los, isto é, quais funções eles têm, quais níveis de habilidades possuem, etc.

O modelo de usuários é construído no processo de Análise de Usuários. Apresentaremos, primeiramente, alguns fatores comuns de análise: função do usuário na organização, nível de habilidade com computadores, nível de habilidade no domínio da aplicação, padrão de uso, “*background*” social/cultural. Em seguida apresentaremos um procedimento para identificar fatores de análise específicos da aplicação e grupos de usuário distintos.

6.3.1 Fatores de Análise de Usuários

6.3.1.1 Função

A função (ou papel) específica de cada usuário é definida pelas tarefas que eles realizam. Numa organização, as pessoas trabalham juntas, de maneira estruturada. A estrutura da organização define relacionamentos entre as pessoas. A implicação imediata das diferentes funções de cada usuário são as diferentes tarefas que eles irão realizar. Algumas tarefas podem ser comuns a diferentes funções de usuários, enquanto outras podem ser exclusivas de funções específicas.

6.3.1.2 Nível de Habilidade com Computadores

Essas habilidades são medidas pela proeficiência no uso de dispositivos de entrada (p.ex., teclado, mouse) e pela familiaridade com o modelo de interação do ambiente de interface. As habilidades podem ser adquiridas pela experiência prévia com outras interfaces gráficas ou pela frequência de uso.

- **Iniciante** - Pouco conhecimento de um ambiente de interface gráfica ou de como usar um dispositivo de entrada. Costuma cometer erros. Precisa de auxílio e aprendizado extensivo (p. ex. help on-line e tutorial). As modificações de informações e mudanças de estado da interface precisam ser lentas para que o usuário possa acompanhar e aprender.
- **Experiente** - Tem experiência considerável em aplicações com interfaces gráficas. Usuários tornam-se experientes com a prática frequente. Não precisa de suporte extensivo, prefere um interação mais rápida. Precisa de interfaces que possibilitem “*short-cuts*” (atalhos), para que possa atingir alta performance.

6.3.1.3 Nível de Habilidade no Domínio da Aplicação

Como as aplicações são desenvolvidas para ajudar usuários na realização de suas tarefas, eles próprios devem ter conhecimentos, em diversos níveis de habilidades, sobre as tarefas e como executá-las

- **Novato** - Não tem conhecimento prévio do domínio de aplicação. Precisa aprender. A interface deve proporcionar informações sobre o estado da aplicação para guiar o usuário e oferecer dicas claras para recuperação de erros.
- **Especialista** - Tem extensivo conhecimento do domínio da aplicação e conhece diferentes maneiras de realizar as tarefas da aplicação. A

interface deve oferecer sofisticadas facilidades que permitam a ele modificar e estender a capacidade da aplicação.

6.3.1.4 Padrão de Uso da Aplicação

1. **Usuário Ocasional** - Não melhora suas habilidades com o computador ao longo do tempo e pode até perder algumas delas.
2. **Usuário Frequente** - Suas habilidades avançam e suas necessidades de ajuda e aprendizado diminuem ao longo do tempo. A necessidade de interação mais sofisticada aumenta.

6.3.1.5 Diferenças Socioculturais

Problemas da língua e tradição cultural têm maior importância em software que serão distribuídos em diversos países do mundo, como software básicos (sistemas operacionais, editores, etc.) e software de companhias multinacionais.

6.3.2 *Procedimento para condução da análise de usuários*

Para melhor compreendermos esse procedimento, vamos utilizar o seguinte exemplo: Uma empresa de software carioca pretende desenvolver uma aplicação de contabilidade, mono-usuário, para computadores pessoais, que será vendida a pequenas firmas de contabilidades em vários estados do Brasil.

6.3.2.1 Identificar fatores de análise críticos para a aplicação

Alguns dos fatores de análise citados acima podem não ter importância para determinada aplicação. No exemplo acima, como a aplicação é mono-usuário, a função de usuário não é um fator crítico para análise. Além disso, a aplicação será destinada a usuários de um mesmo país de um ambiente social comum, as firmas de contabilidade, o que elimina o fator de diferenças socioculturais. Os fatores críticos são, portanto, as habilidades com o computador e do domínio da aplicação e o padrão de uso.

6.3.2.2 Descobrir outros fatores críticos para a aplicação

Os fatores citados acima, são padrões para uma plataforma de análise básica. Para uma determinada aplicação, pode-se considerar que a familiaridade com algum estilo de interface gráfica (Windows, Macintosh, Motif, etc.) seja um fator crítico.

6.3.2.3 Estimar a distribuição de usuários para cada fator

Fazer um levantamento estimado do percentual de usuários em cada grupo de fator crítico. Para o exemplo acima:

Nível de habilidade com computador

Iniciante: 25%
Intermediário: 30%
Experiente: 45%

Nível de habilidade no domínio da aplicação

Iniciante: 5%
Intermediário 15%
Especialista: 80%

Padrão de uso

Ocasional: 20%
Frequente: 80%

Ambiente gráfico preferido

Windows 3.x: 50%
Macintosh: 35%
Outros: 10%
Não sabe: 5%

6.3.2.4 Identificar grupos majoritários de usuários

Verificar de acordo com o percentual de cada categoria qual é o grupo majoritário de usuários. Esta análise implica em certas decisões de design que dará à interface certas características específicas, visando satisfazer a maioria dos usuários.

6.3.2.5 Analisar a implicação coletiva da distribuição de usuários

A distribuição não deve ser um fator absoluto e alguns aspectos subjetivos podem ser considerados. Deve-se considerar, por exemplo, que usuários iniciantes e novatos vão adquirir experiência, fato que pode mudar o modelo final de usuários.

6.3.3 Conclusão

Concluindo, o processo de análise que acabamos de mostrar leva à elaboração do modelo de usuários da aplicação, que é essencial para o design da interface. Nesse modelo deve estar descrito o perfil dos usuários, isto é, suas características seus níveis de habilidade e conhecimento, etc.

O modelo de usuário final deve apresentar também resultados conclusivos que indiquem os requisitos mais importantes e que devem estar presentes na interface a ser projetada. Os projetistas devem poder, a partir desse modelo, tomar decisões sobre as características da interface.

O modelo pode apresentar uma descrição do usuário padrão, isto é, aquele que descreve a maioria dos usuários. Em casos específicos quando as características de usuários forem bastante diversificadas, a análise de usuário deve apresentar diversos modelos e enfatizar a necessidade de interfaces com características diferentes para cada usuário.

6.4 Análise e modelagem de tarefas

A análise de tarefas ajuda ao projetista da interface ter uma visão da aplicação sob a perspectiva do usuário, isto é, um modelo das tarefas do usuário quando executando sessões da aplicação.

Abordaremos a análise de tarefas através de um modelo GOMS¹ simplificado. O modelo **GOMS** (*Goals, Operators, Methods, and Selection Rules*)² oferece uma abordagem de análise da tarefa baseada num modelo do comportamento humano que possui três subsistemas de interação: o **perceptual** (auditivo e visual), o **motor** (movimentos braço-mão-dedo e cabeça-olho), e **cognitivo** (tomadas de decisão e acesso a memória). O modelo GOMS descreve o comportamento dinâmico da interação com um computador, especificando-se:

Metas - Uma aplicação é desenvolvida para auxiliar os usuários atingirem metas específicas. Isso requer uma série de etapas. Dessa forma, uma meta pode ser decomposta em várias submetas, formando uma hierarquia.

Operadores - São as ações humanas básicas que os usuários executam.

perceptual - operações visuais e auditivas (*olhar a tela, escutar um beep*).

motor - movimentos braço-mão-dedo e cabeça-olho (*pressionar uma tecla*).

cognitivo - tomadas de decisão, armazenar e relembrar um item da memória de trabalho.

Métodos - Um método é uma sequência de passos para se atingir uma meta. Dependendo do nível da hierarquia, os passos num método podem ser submetas, operadores ou a combinação de ambos.

Regras de Seleção - Pode existir mais de um método para se atingir uma meta. Uma regra de seleção especifica certas condições que devem ser satisfeitas antes que um método possa ser aplicado. Uma regra de seleção é uma expressão do tipo “condição-ação”.

O GOMS permite que se construa modelos de tarefas bem mais complexos e detalhados do que o necessário numa tarefa de análise para a construção de interfaces. Usaremos uma versão simplificada do GOMS, pois:

- o modelo da tarefa não deverá descrever informações de design da interface, uma vez que ela ainda não foi construída;
- o analista não é um especialista em psicologia cognitiva;
- o modelo simplificado pode ser expandido para o original, o que é bastante útil.

6.4.1 Diretrizes do Modelo de Tarefas Simplificado

Uma vez que a hierarquia de metas representa um aumento no nível de detalhe, se nos limitarmos à descrição de metas e submetas de mais alto nível, nenhuma decisão de design será envolvida.

¹O modelo GOMS original foi desenvolvido por [S.Card, T.Moran & A.Newell 93]. A versão simplificada é de autoria de [G.Lee 93].

²Metas, Operadores, Métodos e Regras de Seleção.

- **Faça a análise “top-down”** - comece das metas mais gerais em direção as mais específicas.
- **Use termos gerais para descrever metas** - não use termos específicos da interface.
- **Examine todas as metas antes de descer para um nível mais baixo** - isso facilita reuso de metas.
- **Considere todos os cenários de tarefas** - as regras de seleção possibilitam representar alternativas.
- **Use sentenças simples para especificar as metas** - estruturas complexas indicam a necessidade de decomposição da meta em submetas.
- **Retire os passos de um método que sejam operadores** - os operadores são dependentes da interface.
- **Pare a decomposição quando as descrições estiverem muito detalhadas** - quando os métodos são operadores ou envolvem pressuposições de design.

6.4.2 Modelagem da Tarefa para aplicações com múltiplas funções de usuários.

Se, para uma determinada aplicação, a função do usuário for um fator crítico dominante na análise de usuários, deveremos ter modelos de tarefas diferentes para cada função de usuário. No GOMS simplificado, veremos como representar as tarefas para cada usuário num só modelo. Antes de estudarmos a notação do modelo, vejamos algumas regras para modelos com múltiplas funções de usuários:

Inicie especificando metas de alto nível para cada função de usuário.

Se mais de um compartilham a mesma meta, agrupe-os sob uma só.

Se todos compartilham a mesma meta, retire as referências a funções de usuário.

6.4.3 Notação

6.4.3.1 Notação para Funções de Usuários.

- Funções de usuários distintas serão denotadas pelo símbolo **FU** seguido por um número.
ex.: Gerente de vendas (**FU1**), balconista (**FU2**), caixa (**FU3**)
- A descrição de cada função de usuário é a primeira parte do modelo de tarefas.
ex.: Gerente de vendas (**FU1**): responsável pela vendas nas lojas. Tem acesso a todos os dados do sistema.
- Um ponto-e-vírgula (;) é usado para separar o símbolo da função do usuário do restante da notação do modelo de tarefas.

ex.: FU1;...

- Se uma meta é usada para mais de uma função de usuário, elas devem ser separadas por uma vírgula (,).

ex.: FU1, FU2; ...

- Um asterisco (*) é usado se uma meta é aplicada a todas as funções de usuários.

ex.: *;...

6.4.3.2 Notação para especificação de metas

- Cada passo num método é numerado numa ordem seqüencial, com cada nível de decomposição separado por um ponto (.), e com a indentação apropriada para reforçar a noção de hierarquia. Após o último número usa-se dois-pontos (:).

ex.: FU2; 2.1: Anotar correções.

- Um comentário começa com duas barras inclinadas para direita (//) e acaba ao final da linha.

ex.: // Para fazer as anotações o balconista deverá examinar as

// listagens produzida durante as vendas do dia.

FU2; 2.1: Anotar correções

6.4.3.3 Notação para Métodos e Regras de Seleção

- Se uma meta possui mais de um método para ser atingida, uma letra do alfabeto é usada de forma ordenada após o número que descreve a meta.

ex.:

FU2; 2: Fazer relatório de vendas (meta)

FU2; 2A: ... (método A)

FU2; 2B: ... (método B)

- As regras de seleção e o método associado são descritos como pares “condição-ação”, logo após a notação numérica da meta.

ex.: FU2; 2A: se (dia de hoje for sábado)

então (fazer relatório semanal)

6.4.3.4 Reutilizando Metas

Um aspecto importante dessa notação é que pode-se reutilizar metas, simplesmente usando o mesmo número de uma meta preexistente.

ex.:

FU2; 2.1: Anotar correções. (meta preexistente)

...

FU1, FU3; 3: Modificar livro-caixa.
FU1, FU3; 3.1: Procurar lotes em aberto.
FU1, FU3; 2.1: Anotar correções. (meta reusada)
FU1, FU3; 3.3: Recalcular valores.

6.4.3.5 Diretrizes adicionais

- Delimite os passos de um método entre chaves: {}.
- Em aplicações com apenas uma função de usuário, não é necessário incluir a notação de função usuário antes de cada meta.
- Num nível de decomposição onde todas as metas têm o mesmo número-prefixo, apenas o número que indica a sequência naquele nível é necessário.
- A diretriz anterior se aplica também à notação de função de usuário.
- Ao reutilizar uma meta anterior é necessário usar a notação completa para ela.

ex.: FU1, FU3; 3: Modificar livro-caixa.
1: Procurar lotes em aberto.
2.1: Anotar correções. (meta reusada)
3: Recalcular valores.

6.5 Prática de análise

Suponha que você tenha identificado um amplo potencial para o desenvolvimento de um sistemas “Agente de viagens”. A idéia é ter um sistema que possa interagir com bancos de dados de companhia aéreas e hotéis do Brasil e possibilite ao usuário planejar a sua viagem realizando a reserva e compra de passagens e a reserva de hotéis.

Identifique os potenciais usuários do seu sistema, trace um perfil representativo e descreva as principais tarefas que eles necessitam realizar. Finalize a modelagem descrevendo os requisitos funcionais e os requisitos de usabilidade que você considera adequado aos perfis dos usuários.