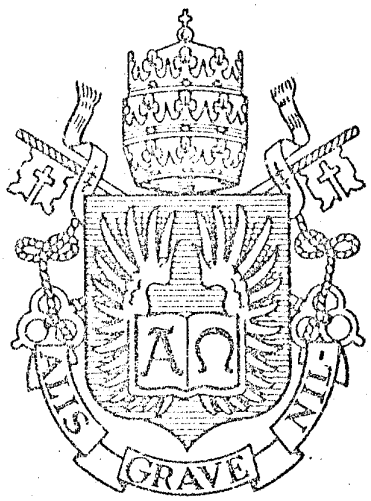

PUC

Série: Monografias em Ciência da Computação, nº 3/89

S.O.S.
SISTEMA ORIENTADO A SOCORRER

Paula Y. dos Guaranys

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 – CEP 22453

RIO DE JANEIRO – BRASIL

PUC/RJ - DEPARTAMENTO DE INFORMATICA

Série: Monografia em Ciencia da Computação, No. 03/89.
Editor: Paulo A.S. Veloso março, 1989.

S.O.S.
Sistema Orientado a Socorrer *

Paula Y.dos Guaranys **

* Apresentado por Donia Scott.
** Parcialmente financiada pela SID Informática e CNPq.
Cursando o programa de doutoramento do DI.

Responsável por Publicações

Rosane Teles Lins Castilho
PUC/RJ-Depto. de Informática
Assessoria de Biblioteca, Documentação e Informação
Rua Marquês de São Vicente, 225 - Gávea
22453 - Rio de Janeiro, RJ

RESUMO

Descrição de um sistema para ajudar usuários durante utilização de softwares interativos e dirigidos por comando. As ajudas fornecidas são baseadas no modelo do usuário, gerado e mantido durante a interação. O modelo começa com transformações a partir de um duplo estereótipo: usuário e conhecimento, fornecidos pelo projetista do software.

ABSTRACT

Description of a system to help users interact with a command-driven software. The help function is based on user model, generated and maintained during the interaction. The model begins with transformations over a double stereotype: user and knowledge, provided by the software designer.

SUMARIO.

LISTA DE FIGURAS.

LISTA DE TABELAS.

1. INTRODUÇÃO:

Neste artigo é descrita e apresentada uma proposta sobre um sistema para "socorrer" usuários com diferentes níveis de conhecimento durante utilização de softwares interativos e dirigidos por comando. Cada comando é tratado ao longo de todo o texto como um plano. Esta proposta é concretizada no sistema denominado S.O.S..

Os tipos de ajuda disponíveis são dependentes das interações existentes no software. Para softwares mais complexos, que algumas vezes requisitam traçado de planos, ou em que diferentes níveis de detalhamento de explicação são possíveis, existe uma maior necessidade e também possibilidade de se fornecer uma ajuda. A geração de um plano, ou apenas seu reconhecimento, uma explicação assim como uma simples mensagem são possíveis ajudas e podem ser classificadas como passivas e/ou ativas.

Nossa opinião é de que se alguma ajuda é dirigida ao usuário, o mesmo terá que compreendê-la. Como o tipo de usuário interagindo com o software é muito variado, diferentes níveis de ajuda são necessários. Isto significa que o sistema reage diferente dependendo do usuário. O mesmo ocorre, por exemplo, em Tailor [2] que necessita de um modelo atualizado e correto do usuário, representando o estado de seu conhecimento, para então ser capaz de dar uma ajuda.

A inexistência de um usuário padrão obriga o sistema a ter informações sobre o usuário interagindo com o software, isto é,

ele necessita de um modelo do usuário propriamente dito. O enfoque maior neste artigo é na geração e manutanção do modelo do usuario próprio.

O modelo do usuário no S.O.S. é dinâmico e baseado em estereótipos iniciais [3]. Durante a utilização do sistema pelo usuário, a cada interação, o modelo do usuário próprio fica menos parecido com o estereótipo e mais com o usuário corrente.

O estereótipo inicial contém informações que relacionam tipos de usuários com conhecimentos sobre o software bem como conhecimentos entre si. Os dados que compõem o estereótipo inicial são obtidos a partir de um experimento com uma variedade de usuários, todos com um mesmo objetivo a ser alcançado usando o software. É este estereótipo inicial, que durante a interação do usuário com o software resultará num modelo do mesmo possibilitando ajuda apropriada no momento adequado.

Apesar da geração do estereótipo ser resultante de um experimento, o processo foi generalizado e, portanto, pode ser aplicado para qualquer software, desde que interativo e dirigido por comando.

2. PLANOS:

O sistema S.O.S. tem seu funcionamento fortemente baseado em planos. Toda ajuda fornecida ao usuário é derivada da sequência de passos que o mesmo executa durante a interação com o sistema, isto é, de seu plano.

Além dos planos que cada usuário executa, o S.O.S. também contém uma base com os planos possíveis de serem executados por um usuário durante sua interação com o software. Estes são dependentes do software, constantes para uma mesma aplicação e independentes do tipo de usuário.

Todos os planos existentes na base, tanto os gerados como os constantes para uma aplicação, possuem os mesmos tipos de informação semântica. A representação sintática de um plano descreve uma relação entre tres tipos de informação semântica.

Um plano é sintaticamente representado através de um predicado com tres argumentos. Semanticamente é um relacionamento entre o objetivo a ser atingido (o efeito final), a maneira como consegui-lo (a ação necessária) e o estado em que a aplicação necessita estar para a sua realização (a pré-condição). Como um exemplo considere um software para processamento de texto e um usuário que deseja retirar uma palavra do texto sendo produzido: o efeito final desejado é "palavra retirada", a ação necessária é "CTRL T" e a pré-condição é "cursor no início da palavra a ser retirada".

Há planos que não são tão simples como os deste exemplo.

Alguns, que para serem executados, necessitam que mais de uma pré-condição independente seja considerada antes, outros ainda teem como ação uma sequência de passos. Para a sua realização, cada pré-condição tem que ser previamente executada. Isto é feito tornando cada pré-condição um outro plano, e que por sua vez tem efeito, ação e pré-condição. No caso de uma ação sendo uma sequência de passos, não se trata de outros planos, mas de sub-ações que são realizadas uma a seguir da outra. Este é o caso de ativar um menu e mover-se ao longo do mesmo para a seleção de um determinado item.

Nos dois parágrafos anteriores foram apresentados dois exemplos de planos com diferentes complexidades. Mas é importante ressaltar que na base de planos estão armazenados planos e não ninhos de planos. E o sitema S.O.S. que tem capacidade de gerar e reconhecer os ninhos de planos.

A descrição sintática assim como a utilização dos planos armazenados é detalhada e discutida nas próximas sessões deste artigo.

3. SOCORROS:

No sistema S.O.S., o usuário recebe auxílio em duas situações bem diferentes. Na primeira, é o usuário, que sentindo necessidade de maiores informações, toma a iniciativa e faz a requisição. A resposta do sistema é denominada de ajuda passiva, isto é, o usuário pede uma ação do sistema. O outro tipo de ajuda possível é denominado ajuda ativa e é uma consequência de decisão do próprio sistema. É o sistema que "percebe" que o usuário está com algum problema e necessitando de ajuda.

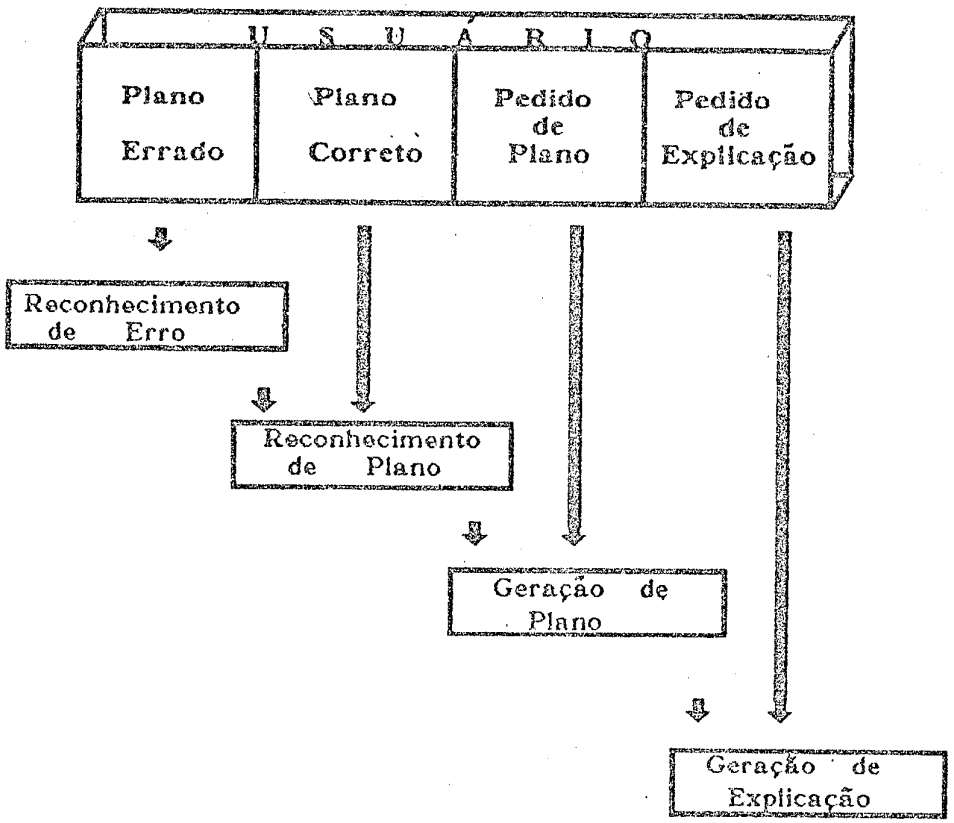

figura 1

O sistema proposto apresenta as seguintes facilidades: reconhecimento de erro e de formação de um plano, geração de um plano completo e de uma explicação. A execução de uma destas facilidades, em geral, está relacionada com a das outras também, e é função do tipo da ação do usuário que provocou o acionamento de uma facilidade. A figura 1 serve para esclarecer, até agora um pouco, o funcionamento geral. Ela é mais detalhada ao longo desta sessão.

As ajudas ativas exigem do sistema bastante processamento e ocorrem em duas situações: usuário comete um erro ou usuário executa um plano corretamente mas de forma pouco eficiente.

O sistema reconhece a presença de um erro quando um usuário envia ao software uma ordem ou comando que não pode ser cumprido no contexto existente; é a facilidade de reconhecimento de erro a responsável por esta parte. O erro pode ser devido a falta de um pré-requisito ou a um comando com a ação diferente da esperada, ou a inexistência de um comando. E a facilidade encarregada de reconhecimento de plano quem detecta a razão do erro. A seguir, a facilidade de geração de plano estabelece qual é o par contexto-ação correto e finalmente a facilidade de explicação fornece ao usuário a razão do erro e como proceder para corrigí-lo.

A ocorrência desta situação descrita é interpretada como: usuário não sabe tal plano. Além da explicação gerada ao usuário, uma outra ação é executada por parte do sistema. Esta é uma ação interna e consiste na atualização da informação que o sistema tem sobre o conhecimento do usuário. E armazenado que

o usuário não sabe sobre dito plano. Se, além disto, o sistema tinha informação contrária até o momento, todos os demais planos relacionados com este são reanalisados, ocorrendo então uma propagação das alterações que determinam a atualização do modelo do usuário.

O procedimento exposto é uma ajuda ativa e se utiliza de todas as facilidades disponíveis no sistema. Cada facilidade é um módulo fechado que é ativado com uma entrada e produz uma saída que ativa outro módulo; excessão feita ao módulo de explicação que exibe ao usuário sua saída.

Se a ação fornecida pelo usuário está "correta" o módulo de reconhecimento de erro não será necessário. A ativação dos demais módulos não ocorre automaticamente como no caso anterior. Em princípio, a sequência é a mesma, mas a ativação do seguinte depende da existência de saída da facilidade sendo processada bem como da saída propriamente dita. Para melhor compreensão serão apresentados os diferentes passos durante a execução.

O usuário fornece um comando que é recebido pelo software como coerente com o contexto e portanto assumido como um plano correto. O módulo de reconhecimento de plano analisa este plano com relação aos entrados até o momento e tenta descobrir a intensão do usuário. Passa a informação obtida ao módulo seguinte, o de geração de plano, para gerar a partir do contexto atual um plano que permita ao usuário alcançar seu objetivo, de acordo com a dedução feita no módulo anterior.

Dependendo do último plano fornecido, isto é, se este é a meta de um plano existente no sistema, inicia um processo de comparação entre os dois planos: o existente no sistema e o executado. Esta comparação tem por finalidade ver se o plano existente é mais eficiente que o executado. Se for, verifica-se se o que torna o plano mais eficiente é algum sub-objetivo desconhecido pelo usuário. Neste caso, o módulo de geração de explicação recebe o plano mais eficiente para exibí-lo ao usuário. Juntamente com o plano, é mostrado ao usuário como realizar as partes por ele desconhecidas. No capítulo seguinte apresenta-se os critérios para a determinação de eficiência.

O fato do usuário executar corretamente um plano dá origem a uma ação interna semelhante a que ocorre ao reconhecer um erro, mas o processo é no sentido contrário. Armazena-se que o usuário conhece o referido comando e todos os outros comandos com algum relacionamento com este comando são reanalisados.

A utilização correta de comandos do software com detecção de planos ineficientes é o outro tipo de ajuda ativa, isto é, o sistema, independente de pedido do usuário fornece-lhe informações adicionais.

Foram descritas as duas situações em que o sistema reage com explicações ao usuário. Há ainda dois outros tipos de ajuda possíveis mas que dependem de iniciativa do usuário para serem acionadas: são as ajudas passivas. Ambas determinam uma geração por parte do sistema e pode ser um plano completo para alcançar alguma meta ou uma simples justificativa para certa atitude do sistema sobre um socorro prestado.

Quando o usuário sabe o que deseja, mas não sabe como fazer, ele "emite" um S.O.S. ao sistema pedindo um planejamento das ações necessárias a sua efetivação. O usuário forneco seu objetivo e o sistema, através do módulo de geração de planos, analisa o contexto presente e a partir deste gera todos os passos necessários até a realização do objetivo. O plano exibido, ao contrário dos sistemas convencionais, não é o mesmo para todo usuário. Ele tem inteligência já que considera além do estado corrente do sistema, o modelo do usuário próprio, isto é, o estado corrente de seu conhecimento. O plano apresentado é então baseado no conhecimento instantâneo.

Esta sequência de passos que compõem o plano é passada ao módulo de geração de explicação que exibe-os; para cada passo é analisado o conhecimento do usuário sobre o mesmo e em função desta informação será mostrado também como executá-lo.

Esta ajuda passiva usa apenas duas das facilidades existentes no sistema e que são as responsáveis por geração. Isto é natural já que é o usuário quem toma a iniciativa e pede a ajuda. Sendo assim, as facilidades relacionadas com reconhecimento não são ativadas.

O fato do usuário perguntar ao sistema como realizar um plano é interpretado como: usuário não sabe tal plano. Esta conclusão é a mesma de quando o usuário comete um erro, e portanto, a ação interna de atualização é também a mesma. Além disso, se a ajuda contém alguma informação nova para o usuário, é armazenado que este recebeu uma primeira explicação sobre tal plano.

O último tipo de socorro disponível é também uma ajuda passiva e usa-se apenas do módulo responsável pela geração do uma explicação. Na realidade é uma explicação sobre a explicação, isto é, o usuário questiona a explicação pedindo uma justificativa.

O usuário só pode então requisitar uma justificação depois que alguma explicação foi apresentada. A ativação da facilidade de geração de explicação pode ser um simples pedido de definição de um conceito ou terminologia, ou então pode ser função do socorro sendo prestado. Neste caso, de acordo com a saída exibida varia o tipo de justificativa que pode ser requisitado.

Quando a explicação é consequência de um erro do usuário, o sistema fornece a razão do erro e como proceder para corrigi-lo, como já foi dito. Diante desta situação, se o usuário requisitar uma justificativa, o sistema mostrará qual a sequência de interações usuário-software que lhe permitiu chegar à conclusão apresentada.

Nos outros dois casos em que a facilidade de explicação é ativada, o resultado é a exibição de um plano com ensinamentos sobre alguns sub-objetivos; os que o sistema acredita que são desconhecidos pelo usuário. Este pode questionar porque um sub-objetivo foi mostrado como ser atingido e porque um outro qualquer não. Resposta para este tipo de pergunta consiste em apresentar ao usuário o modelo que o sistema tem do mesmo, justificando o que levou-o a estas conclusões.

Uma visão mais geral do exposto permite concluir que o sistema com os quatro módulos que o compõem - reconhecimento de erro, reconhecimento de plano, geração de plano e geração de explicação - fornecerá diferentes tipos de socorro ao usuário.

Foram apresentados quatro tipos de ajuda e cada uma usava-se de menos uma facilidade disponível em relação à outra. Além disto, todos os tipos de ajuda, com excessão de um pedido de justificativa, fornecem um retorno ao sistema que contribui para o modelo do usuário sendo gerado.

4. MODELO DO USUARIO:

O sistema S.O.S. é capacitado para ajudar usuários com problemas durante interação com um software. As ajudas são os diferentes tipos de socorros expressos através de mensagens ao usuário. Como é importante que o usuário entenda as mensagens exibidas e considerando que não há um usuário padrão torna-se necessária a existência de um conjunto de informações descrevendo o usuário interagindo, isto é, o usuário próprio.

Modelo do usuário é a descrição sobre o usuário e é composto basicamente por duas informações. Uma relaciona a probabilidade do usuário saber como realizar uma determinada ação sobre o software, isto é, seu grau de conhecimento e a outra informação apresenta uma classificação do usuário gerada pelo sistema a partir do grau de conhecimento que este tem do software. A segunda informação é função da primeira e portanto só existe após o início da interação do usuário com o software.

A parte do modelo do usuário sobre seu grau de conhecimento, é produzida durante a utilização do software pelo usuário considerando um duplo estereótipo [1] que o sistema tem armazenado. Uma parte do estereótipo relaciona o conhecimento do usuário sobre os planos possíveis de serem executados no software e a outra relaciona as conexões entre os planos do software. Este duplo estereótipo é necessário para que o sistema possa a partir da informação sobre o conhecimento do usuário, com relação a um plano X, inferir seu possível conhecimento sobre um plano Y. E esta inferência que permite o

desenvolvimento do modelo do usuário próprio. A validade destes relacionamentos no estereótipo é expressa por meio de um valor numérico discreto, sendo este, portanto, um fator de confiança.

A validade dos relacionamentos no duplo estereótipo é expressa por meio de um fator de confiança (FC). Ele representa a crença do sistema sobre o conhecimento de um usuário com relação a um plano e sobre a conexão entre dois planos quaisquer, dado que o usuário conhece um deles. Seus valores são resultantes de um experimento feito com um grupo de pessoas. Todas recebem uma mesma meta. A quantidade que conseguiu alcança-la e a maneira como realizou-a são dados coletados e analisados.

Esta idéia surgiu devido ao sucesso observado com alunos tentando usar o software Talisman [4]. Todos eram iniciantes na utilização do mesmo. Os dados obtidos resultaram numa avaliação por baixo do nível de familiaridade existente na relação usuário - software, permitindo então ser aplicado a uma quantidade maior de pessoas.

Para o valor do fator de confiança da primeira parte do estereótipo, obtém-se, a partir dos dados do experimento, um valor percentual relacionando o número de usuários que souberam executar um determinado plano com o número total de usuários participantes. Depois, transforma-se o valor percentual de acordo com a tabela 1, tendo-se então um valor discreto. Sua função é indicar o nível de dificuldade da execução de um plano para os usuários como um todo.

INTERVALO PERCENTUAL	VALOR DISCRETO ASSOCIADO
[0]	0
(0,20]	10
(20,40]	30
(40,60]	50
(60,80]	70
(80,100)	90
[100]	100

TABELA I

A tabela é usada para reduzir de cem para sete o número de possibilidades nos valores do fator de confiança. Isto implica na redução do custo de processamento, pois o sistema S.O.S. armazena os diferentes valores que são associados com um determinado plano durante utilização do software por um mesmo usuário.

A segunda parte do estereótipo, isto é, a da classificação do usuário, usa-se da mesma tabela. A diferença está na geração do valor do fator de confiança. Neste caso, relaciona o número de usuários que conhece um determinado plano com o número total de usuários que além deste, conhece também um outro plano determinado. Este valor indica a probabilidade de um usuário conhecer um certo plano, dado que se sabe que ele conhece um outro.

A partir do duplo estereótipo inicial, a medida que o usuário utiliza o software, o sistema se informa sobre o conhecimento que este tem com relação ao software e inicia a geração de parte do modelo do usuário [3].

Há duas maneiras possíveis para o sistema conhecer o

usuário: se informando do que ele sabe e se informando do que ele não sabe. Os dois tipos de informação são úteis e contribuem para a formação do modelo do usuário. Elas atuam sobre uma máquina de inferência que pode ser disparada em dois sentidos: incrementar os valores dos fatores de confiança quanto ao conhecimento do usuário ou decrementar os mesmos.

A cada vez que o usuário executa um plano corretamente é dispara a máquina de inferência incrementando os valores percentuais. Se, por outro lado, o usuário usa erradamente uma instrução do software, e o sistema tem informação contrária quanto ao conhecimento do usuário com relação a este plano, ocorre o processo inverso. Retira-se a informação que indica que o usuário tem o determinado conhecimento e a máquina de inferência é disparada decrementando os valores percentuais.

A inferência visando incremento usa a parte do estereótipo que relaciona a conexão entre os planos. Após o sistema detectar que o usuário sabe um plano, todos os demais planos, relacionados com este no estereótipo, e tendo até o momento um valor de fator de confiança inferior ao existente na relação, terão seus valores atualizados. Isto significa que a probabilidade do usuário saber estes demais planos é agora maior do que era até agora. Esta alteração do valor do fator de confiança juntamente com a informação do que provocou-a é também armazenado pelo sistema e é denominado histórico.

A existência do histórico é que torna possível a inferência no outro sentido. Quando o sistema conclui que o usuário não sabe realizar um plano, desfaz todos os incrementos

do valor do fator de confiança que foram provocados por este plano, em alguma interação anterior. Isto é possível consultando o histórico, pois este determina qual o valor que foi trocado pelo atual. Mas o decremento do atual valor do fator de confiança para o anterior só ocorrerá se não há nenhuma outra informação no histórico que garanta o valor corrente, ou seja, se não há outro valor também determinando a troca para um valor igual ao atual devido a um outro plano diferente do que está sendo analisado. Esta alteração também será incluída no histórico.

As alterações nos valores dos fatores de confiança, seja para aumentar ou diminuir o mesmo, ocorrem para cada usuário em particular, durante sua interação com o software. E através desta propagação das mudanças que o modelo do usuário próprio sofre variações e se amolda a um usuário específico.

A outra parte do modelo do usuário é a responsável pela classificação do tipo do usuário. A maneira como é organizada varia muito de um sistema para o outro. A quantidade de sub-divisões possíveis numa classificação depende de diferentes aspectos. Por simplicidade e porque acreditamos ser suficiente, o usuário pode ser de um dos tres tipos: iniciante, intermediário ou perito.

Este pequeno número de classificações contrasta com a quantidade de valores discretos pré-determinados, os quais estão bem mais sub-divididos. A justificativa é que o tipo de usuário resulta de uma comparação entre a informação que o

sistema tem sobre o conhecimento do usuario proprio e o valor do fator de confiança obtido no experimento com relação a este conhecimento.

Consequentemente, o usuário será classificado como:

. perito, se sabe todos os planos que foram associados com o valor discreto 10 segundo o experimento,

. intermediário, se sabe mais da metade do número de planos que foram associados com o valor discreto 70 segundo o experimento,

. iniciante, se nenhum dos dois anteriores se aplica.

Organizados assim, a variação do tipo atribuído ao usuário é mais lenta acarretando mudanças mais suaves no comportamento da interface.

O leitor pode perguntar por que não classificar como perito um usuário que conhece tudo o que o iniciante conhece, mais tudo o que o intermediário conhece, mais tudo o que apenas alguns conhecem. A resposta é: porque a probabilidade de um usuário saber todos os planos que apenas alguns sabem e não saber um planos que a maioria sabe é tão pequena que decidimos não considerar estas situações. A mesma razão se aplica no caso de classificar um usuário como intermediário.

5. SOS:

Em cada uma das duas seções anteriores foram apresentados, respectivamente, os diferentes tipos de socorros existentes e que conhecimentos são armazenados e compõem o modelo do usuário. Nesta seção, descreve-se como ambas as partes se combinam para gerar a ajuda adequada a um determinado usuário, isto é, o funcionamento geral do S.O.S..

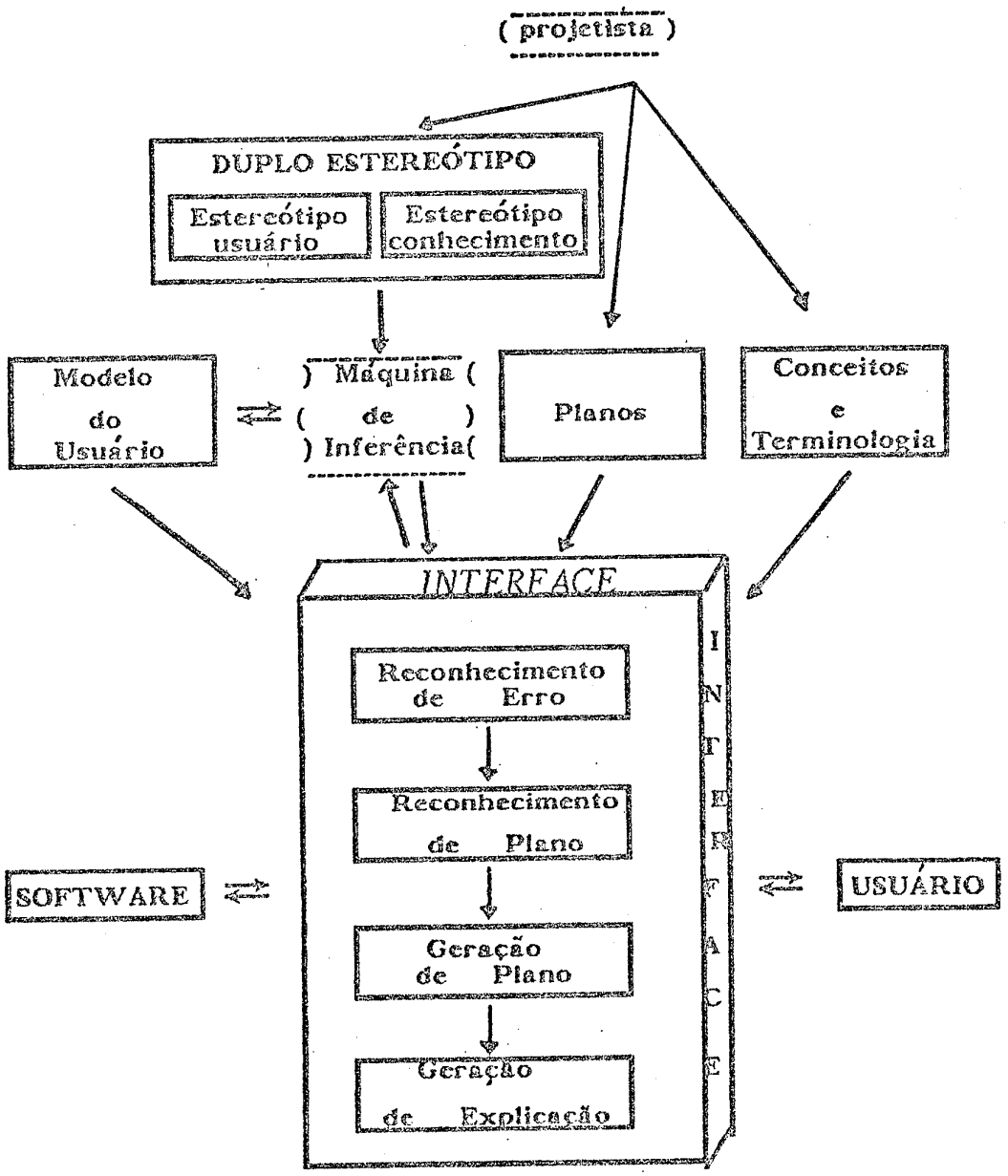

figura 2

Para facilitar a compreensão do geral, os detalhes da implementação vão ser exibidos. A implementação da base de dados do sistema consiste no fornecimento dos planos realizáveis sobre o software, da definição dos conceitos e da terminologia existentes neste, e do duplo estereótipo. Esta parte é fornecida pelo projetista do software. Por outro lado, a implementação da máquina de inferência que é a responsável pela geração e manutenção do modelo do usuário individual, independe do projetista. Esta segunda parte é genérica para softwares interativos e dirigidos por comandos.

A arquitetura geral do sistema é a da figura 2.

5.1 Sobre o Conhecimento:

Nesta primeira sub-divisão são apresentados os detalhes da implementação da parte necessária para o desenvolvimento do modelo do usuário próprio.

Cada plano na base de dados é composto por três informações: a pré-condição para a sua execução, a ação necessária e o efeito resultante de sua execução. A pré-condição pode ser única ou não. No caso de não ser única, ela passa a ser uma sequência de pré-condições, onde cada uma é na realidade um plano composto também pelos três tipos de informação recém descritos.

Um plano é internamente representado através de um fato denominado plano e que é formado por três objetos compostos, isto é, um nome e um parâmetro que no caso é único e é uma

lista de elementos. Os nomes são pré-definidos: pré-condição, ação e efeito. Usando como exemplo um software para processamento de texto, um fato da base de dados pode ser:

```
plano(pré_condição([cursor-inicia-palavra]),ação(["CTRL F"]),
      efeito([palavra_retirada]))
```

Além dos vários fatos descrevendo os planos, na base de dados está o estereótipo duplo: usuário e conhecimento. Este é também descrito por fatos, só que são resultantes de um experimento.

O experimento é realizado com usuários tendo pequenas noções sobre o software. Todos recebem uma mesma tarefa que envolve a execução dos vários planos disponíveis no software. Os dados coletados são analisados para originar os fatos do duplo estereótipo.

O estereótipo do usuário consiste de um conjunto de fatos denominados sabe que possuem dois parâmetros. A quantidade destes fatos é igual ao número de planos que o projetista forneceu ao sistema e o seu formato é:

```
sabe(FC,Plano),
```

onde FC (Fator_de_confiança) representa a proporção de usuários que conhecem Plano. Para obter-se este valor realiza-se a seguinte operação:

$$V = \frac{\text{número de usuários que conhecem Plano}}{\text{número total de usuários no experimento}} \times 100$$

Depois, usando-se a tabela I, localiza-se o intervalo em que o valor V se situa, e por associação determina-se o valor do FC.

O estereótipo do conhecimento, tal como o do usuário, é representado por um conjunto de fatos, neste caso denominados inferencia_conhecimento e tem tres parâmetros. Seu formato é:

inferencia_conhecimento(FC,PlanoX,PlanoY),

onde, FC representa a proporção de usuários que conhecendo PlanoX, conheceram também PlanoY, no experimento. Assim como no caso anterior usa-se a tabela I. A diferença está na forma de calcular o valor V, isto é:

$$V = \frac{\text{número de usuários que conhecem ambos: PlanoX e PlanoY}}{\text{número total de usuários que conhecem PlanoX}} \times 100$$

E o predicado inferencia_conhecimento que possibilita a atualização do valor do parâmetro Fator_de_confiança nos predicados sabe, isto é, a propagação da alteração em todos os fatos da base de dados garantindo assim a consistência das informações do modelo do usuário. Considerando a fórmula acima, temos que o valor do FC associado com o PlanoY é aumentado porque o valor associado com o PlanoX é agora igual a cem. E, a partir do momento em que as alterações começam, passa a existir o modelo do usuário próprio e não mais o simples estereótipo do usuário.

O ultimo tipo de fato que existe na base de dados antes de iniciar a interação do usuário com o software, é o que especifica sua classificação. Conforme já apresentado esta pode

ser: iniciante, intermediário ou perito. O fato recebe o nome de tipo_usuário e só existe uma instanciação possível para o mesmo, a cada vez. Seu único parâmetro, na primeira vez em que o usuário próprio usa o software; tem como valor iniciante. Seu estado é então:

tipo_usuário(iniciante)

Pelo exposto até agora, o usuário começa a interação com o software, tendo na base de dados vários fatos que se enquadram em um dos tres tipos abaixo relacionados e mais o predicado tipo_usuário instanciado como acima:

plano(pré_condição(Pré_cond),ação(Ação),efeito(Efeito))
sabe(FC,Plano),
inferencia_conhecimento(FC,PlanoX,PlanoY),

Os possíveis valores para Plano, PlanoX e PlanoY nos predicados sabe e inferencia_conhecimento são os de Efeito no parametro composto efeito no predicado plano.

A partir destes predicados, e durante a interação usuário - software, o modelo do usuário próprio é gerado. A parte do sistema encarregada desta tarefa é a máquina de inferência. Esta atua em dois sentidos, tanto para incrementar o valor de FC do conhecimento do usuário, propagando a alteração com relação a um plano, como para decrementá-lo.

Após o sistema inferir que o usuário conhece um determinado plano, tres ações principais são realizadas:

1. armazenar que o usuário conhece o plano.

Isto é representado através do valor de FC relativo a este plano, no fato sabe, que passa a ser no máximo cem.

2. incrementar os demais valores de FC nos fatos denominados sabe, isto é, propagar a mudança para garantir a consistência da base de dados.

Para isto é necessário usar:

inferência_conhecimento(FC,PlanoX,PlanoY).

Os valores de FC são determinados pelo fato inferência_conhecimento acima. O novo valor de FC no fato sabe é o indicado por FC no fato inferência_conhecimento.

O que ocorre é que para todo o fato inferência_conhecimento tendo como PlanoX o plano que o usuário demonstra conhecer, altera-se o valor de FC no fato sabe referente a PlanoY, para o novo valor indicado em inferência_conhecimento.

Afim de poder fornecer ao usuário, posteriormente, um histórico de suas atividades, esta alteração é armazenada em um outro fato denominado mudança_conhecimento. A máquina de inferência também se usa deste mesmo fato para decrementar os valores de FC nos fatos sabe.

O fato mudança_conhecimento, como mostrado abaixo, tem quatro parâmetros e representam respectivamente o atual valor de FC e o valor anterior, referentes a PlanoY no fato sabe e que foi alterado devido a PlanoX:

mudança_conhecimento(Atual_valor,Antigo_valor,PlanoY, PlanoX).

Quando o que determina a alteração dos valores de FC não é a realização correta de um plano por parte do usuário, mas sim um socorro prestado, o valor de PlanoY é alguma palavra_chave representando este socorro.

3. verificar se a atual classificação atribuída ao usuário necessita ser mudada.

A análise é feita a partir dos fatos sabe, existentes no estereótipo inicial do usuário, e dos fatos sabe com valor de FC igual a cem, alterados durante a interação e como consequência do sistema inferir que o usuário é capaz de executar um determinado plano.

Primeiro, a máquina de inferência recupera em uma lista todos os planos existentes em fatos sabe com valor de FC igual a cem. A seguir, classificará o usuário como perito se todos os planos com FC igual a dez nos fatos sabe do estereótipo inicial pertencerem a esta lista. Se isto não ocorrer, classificará o usuário como intermediário, desde que mais da metade dos planos com FC igual a setenta nos fatos sabe do estereótipo inicial pertencerem a tal lista. Caso contrário, a classificação do usuário é iniciante.

O resultado da análise é o fato:

tipo_usuário(Classificação)

onde Classificação é uma das tres possibilidades já descritas.

Para a execução da máquina de inferência no sentido contrário, é necessário o sistema inferir que o usuário desconhece um determinado plano. Quando isto ocorre, duas ações principais são realizadas, a semelhança de antes:

1. atualizar os demais valores de FC nos fatos sabe.

Esta ação consiste em decrementar o valor de FC, caso este seja igual a cem.

O que permite a atualização são os fatos inferência_conhecimento existentes no estereótipo do conhecimento e os fatos mudança_conhecimento gerados com a interação do usuário com o software.

Através do fato inferência_conhecimento descobre-se todos os planos que podem ter alterado seu valor de FC nos fatos sabe. O fato mudança_conhecimento permite recuperar o valor de FC existente antes da alteração. Mas é necessária uma análise para definir o valor atual para FC. Este será o maior valor que o plano sendo processado teve ao longo de seu histórico, isto é, entre os fatos mudança_conhecimento.

A alteração do valor de Fator_de_confiança, mesmo que para um valor inferior, tal como no caso anterior, também será armazenado para constar do histórico.

2. verificar se a atual classificação atribuída ao usuário

necessita ser mudada.

Exatamente o mesmo apresentado antes se repete nesta situação.

5.2 Sobre o Socorro:

A finalidade do S.O.S. é servir ao usuário. Até agora não foi mostrado como o sistema realmente socorre o usuário. Mas já foi descrito o conteúdo da base de dados e como o sistema altera esta base, de tal forma que ela seja cada vez mais uma reprodução do usuário próprio. A parte apresentada é o que possibilita a presença do socorro. Nesta sub-divisão é detalhada a ação dos módulos que compõem a interface como mostrado na figura 2:

Dependendo do que levou o sistema a inferir que o usuário sabe como executar um determinado plano, ou que não sabe, diferentes ações são automaticamente realizadas. Isto implica na ativação dos módulos de reconhecimento de erro, reconhecimento de plano, geração de plano e geração de explicação. Qualquer atitude por parte do sistema é registrada na base de dados e passa a influenciar nas próximas ações.

As ajudas que decorrem de processamento automático são as ajudas ativas. O processamento começa no momento em que o usuário tenta realizar um plano, isto é, ele executa uma ação que é analisada pelo módulo de reconhecimento de erro.

Um plano é reconhecido como errado em tres situações. A primeira é quando não existe, isto é, não há nenhum fato plano

em que a ação corresponde à fornecida pelo usuário. As outras duas ocorrem quando a ação fornecida não está de acordo com o contexto atual. O erro pode ser porque o usuário não sabia da necessidade de pré-condição ou porque conhecendo a pré-condição não sabia a ação correta para a realização do plano. Se um destes tipos de erro foi reconhecido neste módulo, é passado ao seguinte que houve erro, e se a natureza do mesmo é por inexistência do plano ou por conflito entre pré-condição e ação.

O módulo seguinte, o de reconhecimento de plano, é ativado em duas situações bem diferentes. Uma delas é quando o plano executado não apresentou erro e a outra, é em função da presença de erro sendo portanto ativado pelo módulo de reconhecimento de erro.

No primeiro caso, o sistema assume que o usuário conhece o plano e dispara as atualizações no modelo do usuário já descritas. Além disto armazena o plano executado, que juntamente com os executados até agora são passados ao módulo seguinte, o de geração de plano.

A ativação do módulo de reconhecimento de plano a partir do módulo de weconhecimento de erro determina a procura na base de dados de possíveis planos. Usa-se de duas informações: a pré-condição no momento em que a informação foi fornecida e a ação propriamente dita. Os planos possíveis, juntamente com os anteriormente executados são então passados ao módulo de geração de planos que analisa a sequência que vem sendo executada afim de que o módulo de explicação possa gerar uma

explicação clara e adequada. Se o erro detectado no módulo anterior foi devido a ausência da ação fornecida entre os planos da base de dados, o processamento continua, usando-se que a pré-condição está correta e procura-se a ação correspondente. Mas chega ao módulo de explicação a informação de que a ação inicialmente fornecida inexiste.

O módulo de geração de plano é ativado tanto quando a sequência de planos executados até o momento não apresenta problemas, como quando devido a um problema encontrado necessita-se saber qual o próximo plano mais provável. Os valores de Fator_de_confiança nos fatos sabe são muito utilizados neste módulo e no seguinte, o de geração de explicação.

A sequência de planos executados sem problemas exige deste módulo, a geração de outras possíveis sequências que envolvam planos, que sejam mais eficientes e que de acordo com o modelo do usuário tem FC baixo, e que por isso o sistema acredita que o usuário desconhece o mesmo. Neste texto, um plano X existente no sistema é considerado mais eficiente que um plano Y executado pelo usuário, se o plano Y é uma sequência de dois planos: Y1 e Y2, e a pré-condição do plano X é a mesma do plano Y1 e o efeito do plano X é o mesmo do plano Y2. A nova sequência gerada é passada ao módulo de geração de explicação que a exibirá.

Quando o módulo está com a responsabilidade de gerar um plano para o usuário, o que mais pesa inicialmente é a sequência que vem sendo realizada até o momento. Se esta

sequência de planos já executados existe em um fato plano como sendo uma ação composta de várias sub-ações, a geração do plano seguinte é trivial. Apenas para que o módulo de geração de explicação seja coerente é fornecido se o erro está na pré-condição ou na ação. Se o erro inferido pelo sistema tem FC baixo para o tipo de usuário, maior a probabilidade do sistema estar correto.

Caso a sequência sendo executada não contribui, o sistema supõe que o usuário deseja realizar um outro plano. Assume a ação como correta e analisa a pré-condição. Se o usuário, segundo o modelo que o sistema tem do mesmo, sabe executar a pré-condição, o sistema infere que ele não sabe realizar o plano, isto é, não sabe que uma pré-condição específica é necessária. Isto implica na atualização do modelo do usuário próprio, além da geração da explicação.

A outra possibilidade é o usuário não saber realizar a pré-condição. Se esta for, por sua vez, um plano na base de dados o mesmo de antes se processa.

O último módulo é o encarregado de gerar uma explicação. Ele é ativado por uma sequência de planos ou mesmo um plano. Ele exibirá ao usuário o que o ativou, além de outras informações que são função do tipo de usuário, da probabilidade de ele conhecer um determinado plano, de ele já ter recebido explicação anterior do mesmo acontecimento e das informações que são trazidas dos outros módulos.

A apresentação ao usuário consiste primeiro da descrição

do erro detectado pelo sistema. Neste ponto, a existencia do modelo do usuário é notada, pois a descrição é mais detalhada ou menos detalhada de acordo com o usuário. Também em função da classificação à qual pertence o usuário é a exibição de cada plano.

Para usuarios classificados como perito não se diz como realizar um determinado plano, apenas se informa de que este consiste; a menos que o usuário requisite explicitamente - ajuda passiva.

Se o usuário é classificado como intermediário, e o sistema tem que exibir um único plano onde o problema é que este desconhece a necessidade de uma pré-condição, esta será detalhada apenas se no fato sabe correspondente, o FC tem valor igual ou inferior a trinta. Caso contrário, apenas a necessidade é citada. Mas se o sistema tem que exibir um ninho de planos, para cada um é feita a mesma análise e a mesma regra é aplicada. Também está incluída na regra a verificação de se o usuário já recebeu ou não uma explicação anterior, pois em caso afirmativo esta não será repetida. Caso o usuário incida no erro, nova explicação é fornecida.

No caso do usuário ser iniciante, a menos que o valor de FC no fato sabe tenha valor noventa ou superior, ou que ele já tenha recebido a mesma explicação anteriormente, a exibição de um plano é sempre acompanhada de qual ação realizar e como.

Independente do tipo de usuário, sempre que o fornecimento de um plano implica em mostrar como fazer para executá-lo, é

armazenado que o usuário recebeu explicações relacionadas com este plano.

A sequencia de ativação dos módulos exposta ocorre automaticamente e constitui a ajuda ativa que o sistema está preparado para dar. As ajudas passivas são instanciações de fatos da base. A primeira delas, o pedido de geração de um plano, significa para o sistema que o usuário desconhece tal plano. A consequência é a exibição do mesmo seguindo as mesmas regras do caso da explicação, bem como dispara uma atualização do modelo do usuário e a inclusão na base sobre o pedido de explicação que ocorreu. Só que neste caso, é feito um mapeamento do plano desejado pelo usuário, para um efeito existente em um fato plano. Uma vez tendo-se instanciado algum fato plano, obtem-se simultaneamente a ação necessária e sua pré-condição. O sistema verifica se a pré-condição está satisfeita no estado atual da interação. caso isto não se realize, procura qual fato plano apresenta como efeito a pré-condição desejada. Este processo continua até o momento em que a pré-condição já está satisfeita. A partir daí, exibe os passos necessários para atingir o efeito final esperado, partindo do estado atual da interação.

A outra ajuda passiva é uma explicação sobre alguma explicação. Esta pode consistir de exibir uma definição de um conceito ou terminologia, desde que previamente fornecido pelo projetista. Também pode consistir de exibir o histórico armazenado, isto é, o que levou o sistema a inferir que devia detalhar mais um determinado plano e outro não. O que permite

estes esclarecimentos são os fatos mudança_conhecimento que armazenam porque um determinado FC aumentou ou diminuiu.

6. CONCLUSÃO:

O duplo estereótipo, formado considerando o conhecimento e o usuário, e os planos são fornecidos pelo projetista como mostrado na figura 2. A partir destas informações, o sistema S.O.S. orientará o usuário na utilização de um software.

Os estereótipos são formados com dados coletados de um experimento, realizado com principiantes quanto a utilização do software. Este enfoque, que é contrário ao de um especilista fornecendo o conhecimento, foi o adotado, porque assim minimiza-se os erros introduzidos por simples sentimento por parte do perito. A justificativa para a utilização de iniciantes no experimento, é para que a análise seja feita no pior caso, resultando então em estereótipos dimensionados por baixo. Acredita-se que assim será maior a quantidade de satisfação gerada pelo seu uso.

Com o projetista fornecendo a parte que cabe a ele, mas nos moldes apresentados, o usuário pode interagir com um software orientado a comandos, tendo apenas noções de sua utilização. Basta para isto incorporar o conhecimento fornecido pelo projetista no S.O.S., pois o modelo do usuário próprio é desenvolvido e consultado sempre que um erro é reconhecido e uma explicação necessita ser exibida.

A abordagem que foi exposta pode parecer pouco eficiente no que se refere a tempo de resposta por parte do sistema. A impressão é de que o processo de formação do modelo do usuário próprio é extremamente lento. A realidade é de que existe

vagarosidade, mas é apenas no início, pois a proporção em que o sistema vai conhecendo o usuário, o número de atualizações na base de dados diminui e portanto o tempo de resposta aumenta. A razão pela qual este problema não é considerado, é que durante as primeiras interações de um usuário com um software novo para o mesmo, é aconselhável reações lentas, pois o contrário pode causar inibição [5]. À medida em que o usuário se familiariza com o software o tempo de resposta aumenta. Esta maneira para socorrer foi a mais apropriada que encontarmos.

É objetivo deste projeto, a continuação de estudos para o desenvolvimento de uma interface amiga, afim de que o projetista possa fornecer o conhecimento dependente do software, de forma mais amigável e portanto interativa. No presente é feita por digitação direta para a base de dados.

7. BIBLIOGRAFIA:

[1] Chin, David N.; "Knome: Modeling What the User Knows in UC"; In Alfred Kobsa and Wolfgang Wahlster, editores, User Models in Dialog Systems, Springer-Verlag, 1987.

[2] Paris, Cecile L.; "Tailoring Object Descriptions To a User's Level of Expertise"; In Alfred Kobsa and Wolfgang Wahlster, editores, User Models in Dialog Systems, Springer-Verlag, 1987.

[3] Rich, Elaine; "User Modelling via Stereotypes", Cognitive Science 3, 329-354; 1979.

[4] Staa, A.v.; "Talisman Manual de Referencia"; Staa Informatica; 1988.

[5] Shneiderman, B.; "Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley Publishing Company, 1988.