

Integrando Fatores Humanos no Processo de Desenvolvimento de Interfaces Homem-Computador Adaptativas

Elizabeth Furtado

Universidade Estadual do Ceará e

Universidade de Fortaleza

Av. Washington Soares 1321, Edson Queiroz

Fortaleza, CE, -001 Brasil

+55 (0)85 273 2833

elizabet@feq.unifor.br

RESUMO

Este artigo apresenta uma nova metodologia de construção automática de interfaces homem-computador adaptativas. Esta metodologia integra fatores humanos no processo de desenvolvimento das interfaces, a fim de diminuir a complexidade do projeto das interfaces e melhorar a usabilidade das interfaces construídas. Fatores humanos considerados aqui, são relativos ao comportamento dos usuários, estudado pelos modelos cognitivos e às recomendações ergonômicas, encontradas em guias de concepção de interface. Mais precisamente, realizamos estudos sobre o comportamento do usuário tanto para facilitar a modelagem de suas necessidades, como para definir um mecanismo de adaptação das interfaces construídas. As recomendações ergonômicas são usadas para deduzir automaticamente aspectos relativos à interação homem-computador.

Palavras chave

Interfaces adaptativas; Recomendações ergonômicas; Fatores Humanos; Geração automática.

ABSTRACT

This work is focused on presenting a new method to build automatically adaptive user interfaces. This method integrates human factors into development process of interfaces in order to both decrease the interface design complexity and to ensure the design of more effective human-computer interactions. Human factors are closely linked on the one hand, to the user's behavior studied by the cognitive models and on the other hand, to the ergonomic principles found into documents, such as user interface guidelines. More specifically, cognitive models are considered to help the designer both to define the user's needs and to design interfaces that could be able to adapt themselves to the user's informational needs. Ergonomic principles are considered to deduce automatically some aspects of human-computer interactions.

Keywords

Adaptive interfaces; Ergonomic principles; Human factors; Automatic generation.

1. INTRODUÇÃO

Os problemas de desenvolvimento dos sistemas interativos são, em grande parte, devido à complexidade da modelagem das especificações conceituais (dados e funções de um sistema). Os métodos de desenvolvimento e/ou linguagens de modelagem, que apóiam esta modelagem, apresentam uma grande quantidade de diagramas com extensivas notações ((Foley 1991), (Shneiderman 1992) (Ujita 1992)). Além disto, aspectos interativos relativos ao controle do diálogo homem-computador (tais como, de tratamento de eventos, de manutenção da coerência entre os dados mostrados na interface e os dados gravados na base de dados) não são fáceis de modelar nem de implementar. Ferramentas computacionais que construam interfaces automaticamente, a partir de especificações conceituais, visam atenuar estes problemas. No entanto, a usabilidade (fácil de usar e de aprender) das interfaces construídas fica comprometida, caso elas se adaptem mal às necessidades do usuário (Hartson 1993). Os usuários possuem necessidades diversas em função de suas diferenças individuais. A maioria das ferramentas de desenvolvimento existentes não tratam as diferenças individuais durante a modelagem de um sistema interativo.

Nosso programa de pesquisa concerne o desenvolvimento de um método para construir interfaces automaticamente a partir de especificações conceituais de sistemas de supervisão, mais precisamente, de seus sistemas de assistência. Um sistema de assistência determina as assistências (por exemplo, alarmes, planos de ações, diagnósticos, etc.), que ajudam o operador a solucionar um problema de funcionamento no sistema, quando ocorre uma falha qualquer. No nosso trabalho, as interfaces construídas que mostram estas assistências, são capazes de se adaptarem às necessidades do operador durante a

solução de um problema. Nós desenvolvemos um mecanismo que controla a adaptação das interfaces, a partir de um modelo conceitual do domínio (modelo de tarefa), descrito em função do comportamento do operador. Além disso, as interfaces respeitam várias recomendações ergonômicas. As recomendações ergonômicas permitem determinar a melhor maneira que as informações devem ser mostradas ao usuário durante sua interação com o sistema (por exemplo, o fato de mostrar somente as informações mais necessárias, de permitir ao usuário controlar o diálogo com o sistema, ...)

Neste artigo, depois de uma exposição sobre os trabalhos relacionados às interfaces adaptativas para os sistemas de supervisão, nós descreveremos a Metodologia de Assistência à Concepção e à realização de Interfaces Adaptadas (MACIA) (Furtado1999a). Concluiremos, ressaltando nossa contribuição e delineando pesquisas futuras.

2. INTERFACES ADAPTATIVAS NOS SISTEMAS DE SUPERVISÃO

A complexidade da estrutura dos processos complexos bem como de suas condições de funcionamento (manutenção, intervenção, ...) não param de crescer. Diante deste problema, sistemas de assistências são sempre implantados nas salas de controle para explorarem dados dos processos complexos, a fim de fornecer ao operador ajuda ao diagnóstico, à correção de um evento anormal, etc. Assim, as tarefas do operador de controlar o funcionamento de um processo complexo e de fazê-lo voltar a uma situação normal de funcionamento tornam-se cada vez mais complexas, devido à grande quantidade de assistências que ele pode receber ao mesmo tempo dos sistemas de assistência.

Para melhorar o problema de comunicação entre o operador e estes sistemas, é necessário interfaces que sejam capazes de se adaptar às necessidades do operador, para determinar quais assistências devem ser oferecidas para ele, como e quando elas devem ser mostradas. A fim de projetar interfaces adaptativas, nós temos estudados modelos cognitivos humanos definidos na psicologia cognitiva. Estes modelos tentam entender o comportamento humano e podem ajudar o engenheiro a identificar as diferenças individuais.

Modelos Cognitivos Humanos

O modelo cognitivo de resolução de problema definido por Rasmussen (1983), e recentemente modificado por Hoc (1996) é capaz de modelar o conhecimento usado pelo operador quando ele está tentando resolver um problema. Esse modelo descreve as fases que o operador pode passar neste momento, tais como: identificação do evento anormal, avaliação da situação do processo complexo, tomada de decisão e execução de ações corretivas do problema. As fases seguidas pelo operador dependem de seu nível de experiência. Um operador experiente pode

identificar uma situação anormal, em seguida, efetuar uma correção imediatamente. Quando isto acontece, pode significar, por exemplo, que ele agiu baseado nos seus automatismos adquiridos por um treinamento excessivo diante das situações passadas. Já um operador novato pode verificar todas as assistências oferecidas a ele, antes de tomar uma decisão. Tal situação acontece, quando ele tem que raciocinar sobre as diferentes alternativas diante de novas situações. É importante destacar, que um operador pode ser experiente diante do tratamento de uma determinada falha e novato no tratamento de um outro tipo de falha.

De acordo com Ujita (1992), um operador realiza muitas ações no tratamento de uma falha. Por exemplo, para tratar um problema ocorrido em função do desnível da água, é necessário abrir a válvula X e diminuir a pressão da turbina Y. Cada ação corresponde ao processo cognitivo (ou ciclo) de resolução de problema do operador, definido acima.

Interfaces Adaptativas

Existem diversos trabalhos no domínio da supervisão (tais como : Tendjaoui, Moussa, Ujita, Normand (1992), Bodart (1994), Tarby (1993) e Foley (1991), que desenvolveram ferramentas para controlar a adaptação das interfaces, usando técnicas de inteligência artificial e modelos cognitivos humanos. No entanto, nós constatamos que existem problemas com a usabilidade das interfaces. Os problemas são parcialmente porque recomendações ergonômicas são raramente levadas em consideração. Além disso, o processo de adaptação consiste em mudar a interface tão frequentemente que o operador torna-se confuso. A maior parte destas ferramentas constróem um modelo do usuário que é atualizado em função de suas interações com o sistema informático. Modelo este, que registra as preferências do usuário (cor, tamanho de janela) para qualquer comando efetuado. Assim, as adaptações efetuadas nem sempre são de interesse do usuário.

No próximo parágrafo, apresentamos o método MACIA, que descreve um processo de construção automática de interfaces inteligentes a partir de especificações conceituais e integrando fatores humanos.

3. MACIA

MACIA se caracteriza por descrever um processo incremental et iterativo próprio ao desenvolvimento de um sistema interativo de supervisão, onde as interfaces são geradas automaticamente a partir do modelo de tarefa feito pelo analista de sistema.

A Figura 1 ilustra as cinco fases de MACIA (a citar : análise do domínio, modelagem do domínio, especificação da interface conceitual, elaboração da arquitetura da interface real e geração/utilização/adaptação das interfaces reais) e seus recursos associados (modelos e regras). Uma

interface conceitual é aquela interface com estrutura e comportamento definidos, mas cuja especificação não

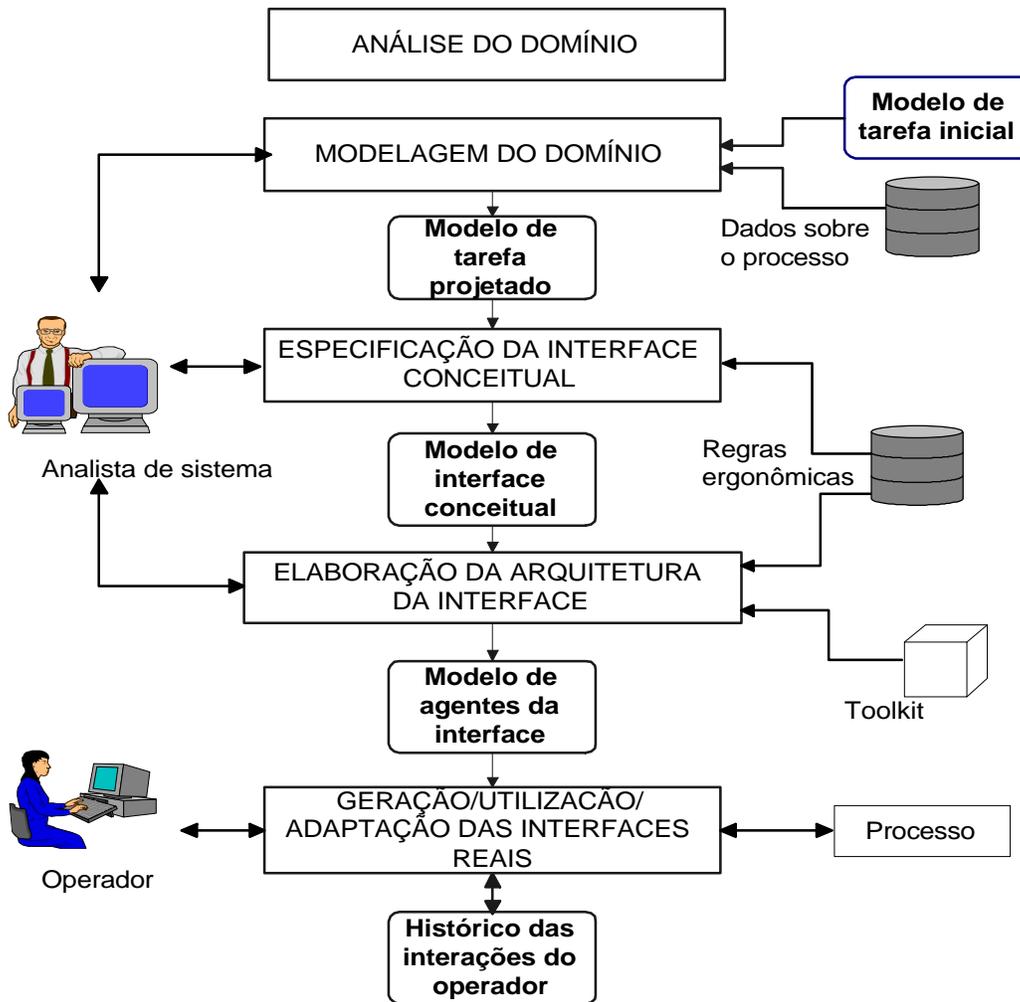


Figura 1- A metodologia MACIA

possui nenhuma consideração sobre a maneira como a interface será mostrada na tela (tais como, a posição das informações, cor e tamanho das janelas, etc.). A noção de *interface real* significa a interface final que será usada pelo operador para interagir com o sistema.

MACIA objetiva diminuir o problema da complexidade da modelagem das especificações conceituais e aumentar a usabilidade das interfaces construídas.

- Para diminuir o problema da complexidade da modelagem, o método integra fatores humanos, que são expressos através do modelo de tarefa inicial e de regras ergonômicas. O modelo de tarefa inicial, no qual é baseado no modelo cognitivo de resolução de problema, é usado para guiar o analista na modelagem das tarefas do operador. As regras ergonômicas, nas quais foram deduzidas de recomendações ergonômicas, são usadas para especificar automaticamente as interfaces conceituais, nas quais

são representadas no modelo de interface conceitual (MIC). Em seguida, regras ergonômicas relativas aos aspectos visuais da interface, são aplicadas sobre a estrutura do MIC para gerar as interfaces reais.

- Para aumentar a usabilidade das interfaces construídas, MACIA define um mecanismo de adaptação das interfaces. A adaptação das interfaces é relativa à assistência cognitiva fornecida ao operador no tratamento de um evento anormal, variando as assistências que devem ser mostradas e a maneira como elas devem ser mostradas (gráfica e/ou textual e/ou diagrama).

Analisemos agora em detalhe cada uma das fases de MACIA.

Fase 1 : análise do domínio

Esta fase começa por um estudo preliminar do domínio para identificar os usuários, os problemas que eles enfrentam com o sistema existente e seus objetivos com o

novo sistema. No fim deste estudo, o analista já está familiarizado com o vocabulário do domínio e pode identificar as tarefas realizadas pelos operadores e os módulos que determinam as assistências sobre o processo em supervisão.

Fase 2 : modelagem do domínio

Esta fase é relativa à modelagem das tarefas do domínio da aplicação. Nesta fase, o analista de sistema deve definir, a partir de um *modelo de tarefa inicial* que nós elaboramos, o modelo de tarefa da sua própria aplicação, chamado *modelo de tarefa projetado*.

Modelo de tarefa inicial

O modelo de tarefa inicial é usado para descrever as tarefas iniciais efetuadas pelo operador de supervisão. As tarefas iniciais são relativas à supervisão do processo industrial e à resolução de um problema, as quais correspondem ao processo cognitivo do operador. O modelo de tarefa inicial é um esqueleto, no qual é usado para guiar o analista na descrição das tarefas de controle de um processo para uma aplicação de supervisão. A fim de definir um modelo de tarefas específico, o analista deve acrescentar outras tarefas ao modelo inicial e inicializar ou modificar algumas características das tarefas (tais como, condições de execução, dados usados durante a execução da tarefa, etc.).

As tarefas de resolução de um problema são as seguintes: *Identificar alarmes* and *Resolver Problemas*. Esta última tarefa é composta de sub-tarefas, as quais são responsáveis por determinar e mostrar as assistências ao operador. Para definir estas tarefas, nós associamos cada fase do processo cognitivo do operador a uma ou mais tarefas. Por exemplo, quando o operador está na fase cognitiva de *avaliar uma situação*, ele pode receber como assistência um diagnóstico e/ou um plano de ação. A Figura 2 ilustra a notação gráfica hierárquica do modelo inicial, onde uma tarefa é representada por um retângulo. Os relacionamentos entre

as tarefas podem ser alternativo ou paralelo. Tarefas alternativas são mutuamente exclusivas. Tarefas paralelas são executadas em qualquer ordem, embora elas possam também ser executadas de forma sequencial. A execução de uma tarefa paralela pode interromper a execução de outra tarefa paralela e/ou ser interrompida.

Basicamente, o funcionamento do modelo inicial é o seguinte: o operador supervisiona o processo enquanto não recebe nenhum alarme. Quando uma falha ocorre, a tarefa de supervisão é interrompida e o operador entra na fase de resolução de um problema, provocado pela falha. Neste momento, o operador faz o diagnóstico do problema e sua correção. Entre a tarefa de diagnóstico e a de correção, o operador obtém algumas assistências em função de seu nível de experiência.

Usando o modelo de tarefa inicial no processo de adaptação de interface

Nós temos definido um mecanismo para calcular o nível de experiência do operador, quando ele está tratando um evento anormal. Este mecanismo é baseado no modelo definido por Ujita (1992) e implementado graças ao modelo de tarefa inicial. O modelo de tarefa inicial permite identificar os diferentes ciclos que o operador passa quando ele está tratando uma falha. Um ciclo começa quando a tarefa *Identificar alarmes* é disparada e é composto pelas sub-tarefas da tarefa de resolução de um problema. Um ciclo termina quando o operador valida a tarefa de correção. A fim de analisar o comportamento do operador, todos os ciclos são registrados num arquivo de histórico do operador com o sistema. Além disto, as preferências do operador com relação às assistências recebidas são também gravadas (tais como se ele pediu uma assistência, ou se recusou uma assistência recebida).

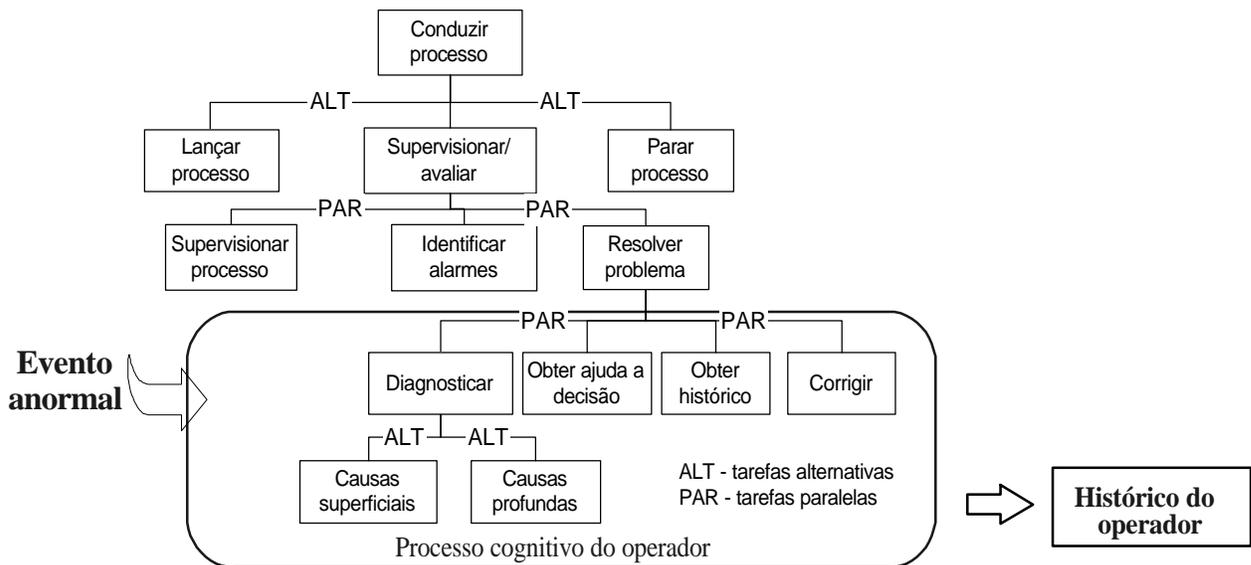


Figura 2 - Modelo de tarefa inicial

A análise do arquivo de histórico do operador é usada para determinar o nível de experiência do operador. Como é difícil saber todas as características do operador relativas ao seu nível de experiência, nós temos escolhido categorizar os operadores, que têm características em comum em estereótipos. Os fatores que constituem o estereótipo são os seguintes : frequência de tratamento do evento, preferência do operador, gravidade do evento tratado e familiaridade do operador com o processo complexo. (Ver mais detalhes sobre este cálculo em Furtado (1999b)). O operador pode ser novato, experiente ou especialista dependendo do conhecimento, que ele tinha quando estava tratando o evento.

O modelo de tarefa projetado

A Figura 3 ilustra um exemplo do modelo de tarefa projetado para a tarefa de supervisionar o processo complexo, chamada *Surveiller Procédé*. O objetivo desta tarefa é de mostrar a estrutura hierárquica do processo complexo, para os atributos chaves de uma entidade desse processo digitados pelo operador. A estrutura apresentada depende dos modos de pesquisa escolhidos pelo operador e que são determinados pelas sub-tarefas da tarefa *Trouver procédé*. Por exemplo, a sub-tarefa *Trouver père* encontra o nó do nível hierárquico superior àquele dos atributos chaves digitados pelo operador.

Todas as características das tarefas modeladas devem ser descritas no formulário mostrado na parte direita desta figura. Uma tarefa é identificada por um número e um nome, associada a um objetivo e possui condições a serem respeitadas antes e depois de sua execução, bem como parâmetros. Um parâmetro corresponde a uma lista de dados da aplicação que vão ser eventualmente modificados na execução da tarefa. Quando o analista vai realizar estas definições, ele pode descobrir um dado que ainda não foi definido, então ele deve retornar à fase de análise do domínio, a fim de analisar de novo o processo complexo. Esta maneira de percorrer as fases de MACIA comprova o seu caráter incremental, onde as fases não são descritas numa seqüência muito rígida.

Após a modelagem de todas as tarefas para uma determinada aplicação, a organização das interfaces e a seqüência das interações possíveis são deduzidas automaticamente das características das tarefas e representadas no modelo de interface conceitual.

Fase 3 : especificação da interface conceitual

Esta fase de especificação da interface conceitual consiste em gerar o Modelo de Interface Conceitual (MIC) a partir de um modelo de tarefa e otimizar a estrutura do MIC gerado. Tanto a estrutura do MIC como a maneira de gerá-lo constitui uma das originalidades de nosso trabalho.

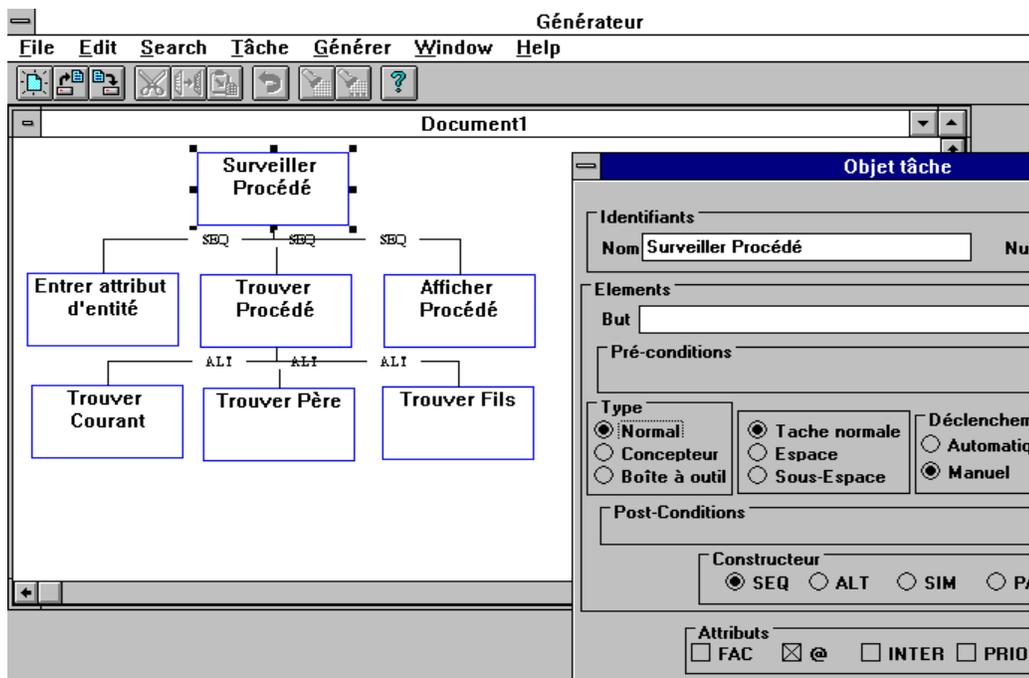


Figura 3 - Um exemplo de um modelo de tarefa projetado para uma aplicação

Modelo de Interface Conceitual

Como o modelo de tarefa representa somente a parte não interativa de um sistema, nós elaboramos o MIC para representar sua parte interativa. A organização do MIC é similar àquela do modelo de tarefa, a fim de que o analista possa perceber facilmente a maneira como as informações serão mostradas nas interfaces que vão ser construídas.

O MIC é composto de dois componentes : espaço de interação e objeto de interação. Um espaço de interação é o lugar de atividade virtual que oferece ao operador uma visão dos dados e tarefas que ele pode manipular. Um objeto de interação é o componente elementar da interface conceitual. Ele permite o diálogo com o operador, definindo vários tipos de interação (entrada, saída e entrada/saída), e representando um dado ou tarefa manipulados pelo operador.

Processo de especificação do Modelo de Interface Conceitual

O processo de especificação da interface conceitual é realizado pela aplicação de regras ergonômicas, deduzidas de recomendações ergonômicas, sobre as características das tarefas. As características das tarefas consideradas foram aquelas que influenciam sobre os aspectos da interface, tais como: se a tarefa é disparada pelo operador ou automática, se a tarefa pede ou mostra informações na tela, etc. As recomendações ergonômicas consideradas foram aquelas que procuram definir e simplificar as visões de trabalho do operador. O objetivo em integrar recomendações, tanto do domínio da supervisão em Kolski (1993) como do domínio dos sistemas informáticos em geral, Scapin (1990) e Shneiderman (1992), na geração do MIC é de assegurar que as interfaces respeitem os critérios tratados por estas recomendações. As regras ergonômicas, que nós elaboramos, foram empiricamente deduzidas das recomendações ergonômicas. Estas deduções foram facilitadas porque qualquer modelo de tarefa é deduzido do modelo de tarefa inicial. O modelo de tarefa inicial permitiu-nos determinar parcialmente o caminho que o operador deve seguir para realizar uma tarefa. As regras podem definir a estrutura do MIC ou sua lógica de funcionamento, assim como podem otimizá-lo.

Durante a passagem do modelo de tarefa para o MIC, cada tarefa pode gerar um espaço ou ser associada a um espaço gerado. Esta escolha depende das características das tarefas e de seus dados. Por exemplo, nós elaboramos uma regra, na qual define que cada tarefa paralela deve dar origem a um espaço diferente. Esta regra objetiva mostrar precisamente ao usuário o momento que uma tarefa paralela interrompe a execução de outra tarefa e/ou o momento que a execução da tarefa deve continuar. Esta regra foi baseada na seguinte recomendação definida no

domínio da supervisão: *uma janela que mostra algum tipo de alarme não deve ser coberta por nenhuma outra janela* (kolski 1993). De acordo com o modelo de tarefa inicial, a tarefa *Identificar alarmes* identifica e mostra alarme e é uma tarefa paralela. Quando tarefas de um mesmo nível são sequenciais ou alternativas, elas podem ser colocadas juntas no espaço já criado por uma das tarefas deste nível. No entanto, restrições podem ser feitas quando o número dos objetos de interação do espaço ficar muito grande ou o tipo do dado for gráfico.

Para definir os objetos de interação de um espaço, nós temos considerado as seguintes recomendações de Beaudouin (1991) e Scapin (1990): *dar oportunidade para o usuário tanto para realizar uma tarefa quanto para cancelá-la, tornar acessível todo dado manipulado pelo usuário durante a execução de uma tarefa, requerer do usuário disparar somente as tarefas que não são executadas automaticamente*, etc. Especificamente, para a primeira recomendação, a regra deduzida foi a seguinte: *Se uma tarefa deve ser somente realizada quando o usuário a valida, então dois objetos de interação devem ser criados. O primeiro objeto serve para ativar a tarefa e o segundo para cancelar sua possível execução.*

A Figura 4 ilustra o MIC deduzido do modelo de tarefa da Figura 3.

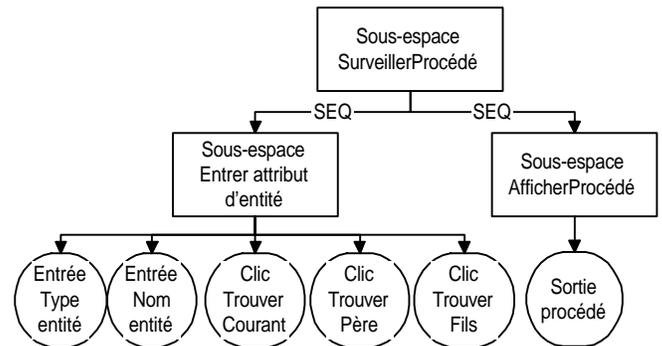


Figura 4 - MIC deduzido do modelo de tarefa projetado

Para obter este modelo, nós aplicamos a regra ergonômica que agrupa tarefas sequenciais num mesmo espaço. Assim a tarefa *Trouver* e suas sub-tarefas (modeladas na Figura 3) são agrupadas no espaço *Entrer attribut d'entité*. O analista pode intervir neste momento, se ele não concordar com a quantidade de informações no espaço. O espaço *Entrer attribut d'entité* fica então composto de 5 objetos de interação representando as características interativas de suas tarefas associadas. Os dois primeiros objetos de interação *Entrée Type entité* e *Entrée Nom entité* representam duas informações que devem ser digitadas pelo operador. Os três outros objetos de interação *Clic Trouver courant*, *Clic Trouver père* e *Clic Trouver fils* representam as sub-tarefas de *Trouver procédé*. A palavra *Clic* significa que o objeto de interação é de entrada e que

sua sub-tarefa associada será executada quando solicitada pelo operador. Com relação à tarefa *Afficher procédé* (Figura 3), ela é separada (gerando um espaço), porque seu parâmetro *procédé* é gráfico. Um parâmetro gráfico ocupa geralmente todo um espaço, a exceção de um pedido do analista.

Quando o MIC é construído, o analista pode validar o modelo para adaptá-lo ao seu próprio estilo realizando as operações seguintes : mudar as características dos componentes MIC, como, o nome do componente que será mostrado na tela, quem dispara um componente), eliminar um objeto de interação, juntar dois espaços de maneira vertical (figura 5 a) ou de maneira horizontal (figura 5 b) e mudar a quantidade de informações de um espaço. Quando o analista concorda com o esboço das interfaces conceituais, ele solicita a elaboração da arquitetura das interfaces, que são representadas no modelo de agentes da interface.

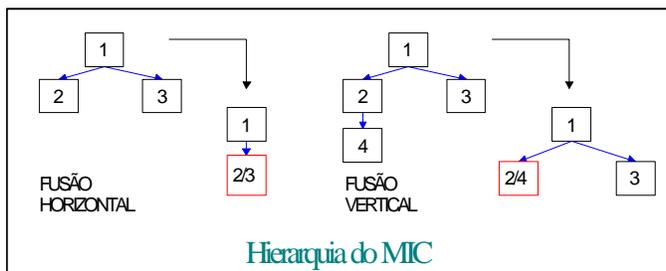


Figura 5 – Hierarquia MIC

Fase 4 : elaboração da arquitetura das interfaces

A fase de elaboração da arquitetura das interfaces consiste em gerar automaticamente o modelo de agente da interface. Este modelo, cuja estrutura é uma decomposição hierárquica de agentes PAC (Coutaz 1991), descreve e organiza a estrutura da interface real e suas relações em termos de agentes. Esta organização hierárquica favorece a distribuição modular das funções relativas à estrutura e ao comportamento das interfaces em todos os níveis hierárquicos.

Um agente PAC é composto de três partes (Figura 6) : aPresentação, abstração e controle. Somente a parte *controle* de um agente é obrigatória e ela corresponde a um componente do modelo de interface conceitual. Ela é responsável por manter a coerência entre os dados mostrados na tela (objetos interativos) e os dados da aplicação e por efetuar o tratamento dos eventos do usuário sobre esses objetos. A parte *aPresentação* de um agente define sua imagem na tela por meio de um objeto

interativo da *toolkit*. Esse objeto é escolhido pela aplicação de regras ergonômicas, que nós deduzimos de recomendações ergonômicas estudadas pela ergonomia cognitiva. Um agente pode ter várias apresentações. A parte *abstração* de um agente corresponde a um dado ou tarefa da aplicação. Se um agente precisa trocar dados com a aplicação, nós definimos na sua parte abstração um mecanismo que permite esta troca.

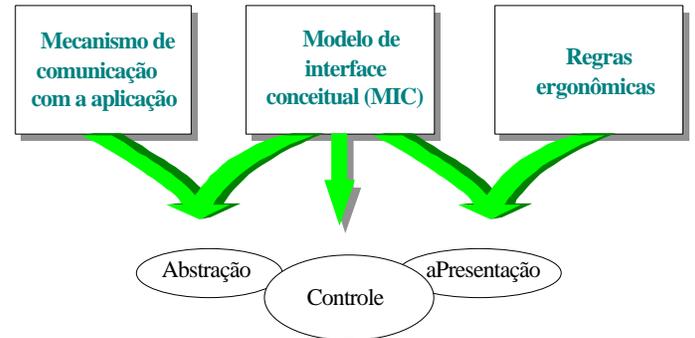


Figura 6 - Um agente PAC

A Figura 7 (parte central) ilustra o modelo de agente da interface para o espaço *Entrer attribut d'entité*. Esse modelo corresponde ao MIC desse espaço mostrado na Figura 4. A parte direita da Figura 7 ilustra a interface que será mostrada ao operador. Cada um dos componentes da interface é associado à parte apresentação dos agentes do modelo. A parte esquerda da Figura 7 ilustra os dados da aplicação que representam a parte abstração dos agentes do modelo. No exemplo, nós mostramos a descrição da classe *Entité* e a descrição da instância dessa classe, cujos valores são mostrados na interface. A parte apresentação do espaço *Entrer attribut d'entité* é representada por uma sub-janela. Essa sub-janela é composta de cinco objetos interativos. Cada um desses objetos representa a parte apresentação dos agentes *objetos de interação*. A parte abstração desses agentes corresponde aos dados da aplicação (como : *Nom d'entité* e *Type d'entité*) ou às tarefas da aplicação (como : *Trouver courant*, *Trouver père* e *Trouver fils*). No caso, por exemplo, do agente objeto de interação *clic Trouver père*, quando o usuário dispara seu botão associado, a parte controle desse agente pede a sua parte abstração para enviar uma mensagem a uma instância da classe, que representa a tarefa da aplicação *Trouver père*, solicitando que a mesma se execute. É assim que se efetua a comunicação da interface com a aplicação.

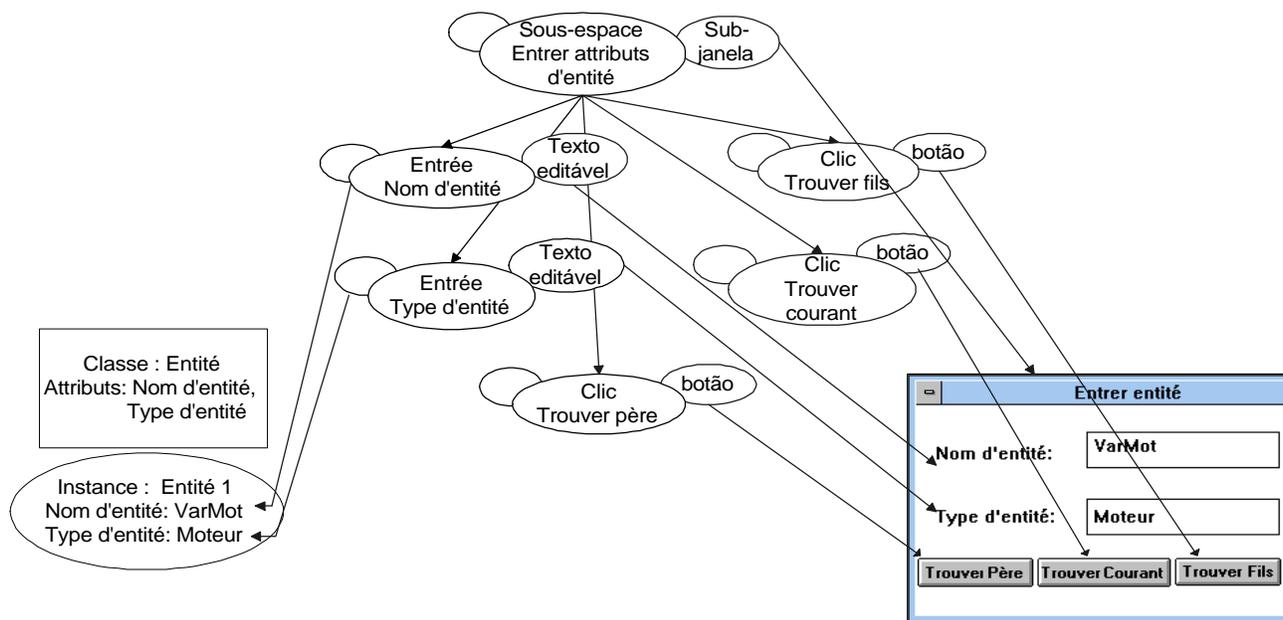


Figura 7 - O modelo de agente da interface para o espaço « Entrer attributs d'entité »

Neste artigo, nós descrevemos unicamente a geração da parte *a* *Presentação* dos agentes. É necessário que essa geração respeite os critérios ergonômicos definidos através das *regras ergonômicas de apresentação da interface*. As regras ergonômicas de apresentação, deduzidas de várias recomendações ergonômicas (Bodart (1994), Vanderdonck e Kolski (1993) e Moussa (1992)), permitem selecionar os objetos interativos e definir seus atributos gráficos (por exemplo, a cor e o tamanho de uma janela). Peguemos um exemplo de uma regra que seleciona um objeto interativo do grupo de apresentação objeto de interação de controle : *se (o número de valores possíveis de um objeto de interação > 10), então (apresentação deste objeto = lista rolante)*.

Fase 5 : geração/utilização/adaptação da interface real

Esta última fase de MACIA consiste, num primeiro momento, em gerar a interface real implementando as várias funcionalidades dos agentes. Vale salientar que o analista de sistema deve ter definido o tratamento algorítmico de cada tarefa. Este algoritmo diz respeito ao tratamento ligado à operação que a tarefa deve realizar e à comunicação entre a tarefa e as primitivas de acesso aos dados da aplicação e as primitivas de acesso aos dados da interface. A ligação entre o código das funções dos agentes e o código das tarefas permite gerar o módulo principal da aplicação.

Num segundo instante, as interfaces, agora reais, são utilizadas pelo operador. Durante esta utilização o

histórico do operador é atualizado para o tratamento de um evento anormal e as interfaces são adaptadas ao operador em questão.

Nós ressaltamos aqui a importância de ser realizada uma avaliação das interfaces pelo analista de sistema e por ergonômicos. O objetivo é de verificar se o sistema informático corresponde bem aos objetivos dos operadores e se a adaptação está coerente com o modelo cognitivo dos operadores.

4. PROTÓTIPO

Atualmente, um protótipo de nosso UMIS já está implementado : i) se trata do *modelador de tarefas*, onde a partir do modelo de tarefa inicial o analista pode incluir novas tarefas, modificar as existentes e definir as características das tarefas e, ii) do *controlador de diálogo*, que é responsável pelo gerenciamento dinâmico da interface por meio de um controle hierárquico dos agentes da interface de um sistema informático. Este gerenciamento consiste em verificar se as tarefas podem ser executadas diante de um evento de interação e em atualizar a estrutura da interface (iniciar, instanciar e terminar os objetos interativos que fazem parte dessa estrutura).

Este protótipo foi desenvolvido em C++ no ambiente WINDOWS. Ele é composto de um conjunto de classes cujas funcionalidades representam a estrutura interna dos *agentes* e das *tarefas*.

5. CONCLUSÃO

Neste artigo nós apresentamos uma nova metodologia de construção automática de interfaces adaptativa. Esta metodologia integra fatores humanos no processo de desenvolvimento do software, realizando assim a ponte entre as duas ciências - Engenharia de Software e Ciências Cognitivas. Fatores humanos relativos ao comportamento do usuário e recomendações ergonômicas são considerados tanto para deduzir automaticamente aspectos relativos à interação homem-computador, como para definir um mecanismo de adaptação das interfaces construídas.

Além do processo automático de geração de interfaces descrito por MACIA, nossa metodologia ajuda o analista de sistema durante o desenvolvimento de um sistema informático através de dois modelos : i) modelo de tarefa inicial, baseado no modelo de Rasmussen, funciona como o “pontapé inicial”, ajudando o analista a compreender melhor as tarefas devendo ser realizadas pelo operador do sistema e evitando que o analista parta do início na modelagem do domínio e, ii) modelo de interface conceitual, funciona como um protótipo das interfaces, permitindo ao analista validá-las antes mesmo de suas gerações definitivas.

6. LIMITES E PERSPECTIVAS

Sendo o modelo de tarefa inicial, o meio usado para modelar as tarefas do operador em função de seu processo cognitivo e para efetuar a adaptação das interfaces, pode-se então garantir a adaptação para as tarefas essenciais da aplicação da supervisão. Porém, apesar da originalidade em realizar a adaptação através de uma modelagem conceitual baseada num modelo cognitivo, é necessário avaliar profundamente os fatores que constituem um estereótipo, e que determinam o nível de experiência do operador.

Visto a importância do modelo de tarefa inicial no desenvolvimento de um sistema, nós almejamos um trabalho de pesquisa para criar um esqueleto genérico para qualquer aplicação e não somente para os casos da supervisão. Isto requer também um estudo aprofundado nos modelos cognitivos a fim de estudar o comportamento do indivíduo em várias situações.

No que diz respeito à especificação da interface, nós insistimos no fato de que várias recomendações foram usadas para definir a estrutura e o funcionamento do MIC. Regras ergonômicas foram então deduzidas. A utilização destas regras visa otimizar o melhor possível as informações manipuladas na interface realizando agrupamentos de tarefas e fusionamentos de espaços. No entanto, ainda podem existir espaços contendo informações redundantes, se eles não são adjacentes na hierarquia. Um método de otimização percorrendo toda a hierarquia deve ainda ser implementado.

REFERÊNCIAS

- M. Beaudouin-Lafon. (1991). « Interfaces Homme-machine : Vue d'ensemble et perspectives ». Actes du congrès Génie logiciel & Systèmes experts, Interfaces homme-machine, Maquettage & prototypage. Vol. 24, pp. 4-16.
- F. Bodart, A. Hennebert, J. Leheureux, I. Provot, J. Vanderdonck. (1994). "A model-based approach to presentation: A continuum from task analysis to prototype". Eurographics. pp. 25-39.
- J. Coutaz. (1991). "Architectural design for user interfaces", Proc of ESEC'91, European software engineering conference. Pp. 7-22.
- J.Foley, W.C.Kim, S.Kovacevic, K.Murray. (1991). « UIDE - An intelligent user interface design environment ». In J.Sullivan & Tyler (Eds) Architectures for intelligent interfaces: elements and prototypes. Reading MA: Addison-Wesley, pp. 339-385.
- E.S. Furtado. (1999). « Taking into account Human Factors in the Automatic Generation of Adaptive User Interfaces ». ASTC99 – HPC (conferência internacional) – EUA – San Diego- CA.
- E. S. Furtado. (1999). « An Approach to improve design and usability of user interfaces for Supervision systems by using Human factors », HCI' 99 International, Munich, Germany.
- Ch. Kolski. (1993). "Ingénierie des interfaces homme-machine ", Hermes.
- Hoc, J.M., (1996). « Supervision et contrôle de processus », La cognition en situation dynamique. Presses universitaires de Grenoble. 1996.
- F.Moussa. (1992). “Contribution à la conception ergonomique des interfaces de supervision dans les procédés industriels: application au système Ergo-Conceptor”. Thèse de doctorat. Université de Valenciennes.
- V. Normand. (1992). "Le modèle SIROCO - de la spécification conceptuelle des interfaces utilisateur à leur réalisation", Thèse de doctorat. Université de Grenoble I.
- J. Rasmussen. (1983). "Skills, rules, knowledge; signals, signs, symbols and other distinctions in human performance models", IEEE trans on SMC, SMC 13(3), pp 257-266.
- Scapin D.L., (1990). "Organizing Human Factors Knowledge for the evaluation and design of interfaces", Int journal of HCI. Vol 2, n° 3, pp. 203-229.
- B. Shneiderman. (1992). « Designing the user interface ». Addison Wesley.
- Tarby J.C. (1993). « Le dialogue homme-machine: des spécification à sa gestion automatique ». Thèse de doctorat, Toulouse.

M. Tendjaoui. (1992). « Contribution à la conception d'interface intelligente pour le contrôle de procédés industriels : application au module décisionnel d'imagerie ». Thèse de doctorat. Université de Valenciennes.

H.Ujita. (1992). "Human characteristics of plant operation and man-machine interface", Reliability engineering and system safety. Vol 38, pp. 119-124, 1992.

7. J. Vanderdonckt, F.Bodart. (1993). "Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection", in Proc. INTERCHI'93, Amsterdam, pp. 424-429.