

MODELAGEM DO FLUXO DE INFORMAÇÕES EM INTERFACES USUÁRIO-COMPUTADOR

Marckson Roberto Ferreira de Sousa, MSc.
Doutorando do curso de Pós-Graduação em Eng Elétrica..
e-mail: marckson@dee.ufpb.br

Maria de Fátima Q. V. Turnell, PhD, membro IEEE e ACM
Professora do DEE/UFPB
e-mail: turnellm@dee.ufpb.br

Jorge Cesar Abrantes Figueiredo, membro IEEE
Professor do DSC/UFPB
e-mail: abrantesc@dsc.ufpb.br

Cursos de Pós-Graduação em Engenharia Elétrica - UFPB
Av. Aprígio Veloso, 882 - Campina Grande/PB, Brasil
CEP 58109-970.

RESUMO

A avaliação das interfaces com o usuário nem sempre ocupa um lugar de destaque no desenvolvimento de produtos, apesar de influenciar sobre sua aceitação. A avaliação pode ocorrer em diferentes fases do ciclo de vida do produto, levando a um contínuo refinamento. Este trabalho propõe um método de análise formal que possibilite a análise dos aspectos estruturais e comportamentais das interfaces ao longo do seu ciclo de vida. Aqui é proposta uma interface genérica a qual representa as situações de transição mais frequentemente encontradas nas interfaces e que serviu de referência para seleção do formalismo. O trabalho propõe o formalismo de redes de Petri como o mais adequado neste contexto e conclui com a discussão do uso de redes de Petri na modelagem de interfaces para fins de avaliação.

PALAVRAS CHAVE: Avaliação de Interfaces com o usuário. Modelagem de interfaces. redes de Petri.

1. Introdução

A qualidade das interfaces nem sempre ocupa um lugar de destaque na avaliação de um produto embora represente para a grande maioria dos usuários a concretização da funcionalidade do produto e o referencial para sua opinião sobre ele.

A avaliação da qualidade das interfaces usuário-computador pode ser mais efetiva se baseada em um método de análise formal capaz de representar suas características estruturais e comportamentais [PALANQUE 94]. Esta representação pode então ser utilizada para avaliar os sistemas modelados e levar a recomendações para eliminar problemas encontrados e sugerir novas alternativas de interação.

O processo avaliatório de interfaces pode ocorrer em diferentes fases do projeto e ao longo de todo o ciclo de vida do produto, levando a um contínuo refinamento. Para que o processo de avaliação seja completo, é necessário

considerar tanto a satisfação dos usuários quanto os aspectos ergonômicos e o desempenho do conjunto usuário-sistema na execução da tarefa.

Segundo Scapin [SCAPIN 92], existem três categorias de métodos para a avaliação de interfaces: métodos baseados no desempenho do usuário durante a interação, métodos baseados em modelos teóricos da tarefa e métodos baseados no julgamento do especialista. Os métodos baseados no desempenho do usuário geram grandes volumes de informação tanto qualitativa quanto quantitativa que vão desde a opinião do usuário sobre o produto até o registro automático ou não de dados da interação, tais como: seqüências de execução de tarefas, seqüências alternativas, seqüências mais freqüentemente utilizadas e o tempo gasto na execução de cada tarefa [DUMAS 94]. A eficácia destes métodos quanto aos resultados da análise depende da sistematização e representação dos dados disponíveis. Os métodos de avaliação baseados em modelos teóricos objetivam prever o desempenho do usuário durante sua interação com o sistema, no entanto são limitados a situações de baixa complexidade. Por sua vez, os métodos baseados no *julgamento do especialista* dependem da capacidade de análise do avaliador que se apoia em critérios derivados de princípios e diretrizes de projeto de interfaces além de sua experiência em avaliação.

Enquanto o processo de especificação de interfaces tem buscado bases formais o mesmo não tem ocorrido com o processo de avaliação. Embora vários métodos e técnicas de avaliação sejam encontrados na literatura, a avaliação de interfaces ainda necessita de formalismos. Segundo [PALANQUE 94], um projeto de interfaces usuário-computador necessita de um formalismo, da definição de uma estrutura de dados e de uma estrutura de controle: de mecanismos de estruturação e da descrição de processamento paralelo. A partir da introdução destes elementos na fase de projeto, acreditamos que seja possível realizar a análise da qualidade das interfaces de modo mais efetivo.

- Formalismo. Por se tratar de um sistema em tempo real, a interface necessita de uma representação formal que facilite sua especificação e análise. Todavia, a dificuldade na escolha de um formalismo começa no entendimento da própria funcionalidade do sistema.
- Definição de uma estrutura de dados e de uma estrutura de controle. Um sistema complexo pode ser bem compreendido se seus aspectos estruturais (ou estáticos) e comportamentais (ou dinâmicos) puderem ser representados através de uma notação única.
- Mecanismos de estruturação. O projeto de uma interface usuário-computador precisa ser bem compreendido, reutilizável e aberto para evolução. Seus componentes devem ser passíveis de uma validação simples, possibilitando sua expansão ou reutilização em diferentes contextos.
- Descrição de processamento paralelo: Na maioria das interfaces, sobretudo onde há a possibilidade de interação com múltiplas janelas, existe a necessidade de sincronizar e cooperar tarefas executadas concorrentemente. Para especificar e analisar estas características é necessário um formalismo capaz de representar concorrência.

Desta forma, a avaliação da qualidade das interfaces usuário-computador pode ser melhor compreendida através da representação de suas características e resultados, utilizando-se um método de análise formal. Do ponto de vista deste trabalho, a análise formal dos sistemas interativos possibilitará a análise do seu comportamento tanto a partir de simulações com o modelo ainda na fase de especificação, quanto a partir da análise dos dados obtidos sobre o desempenho dos usuários interagindo com protótipos ou mesmo o produto final, mapeados neste modelo. Por sua vez, o processo de análise demanda um método de representação também formal. Na fase de análise, é necessária uma representação que auxilie na compreensão do funcionamento dos sistemas, contribuindo para a identificação de erros de projeto e facilitando o projeto de modificações.

Uma das razões pelas quais construir e manter sistemas é um processo caro e propenso a erros é a dificuldade em comunicar claramente as idéias. A dificuldade de compreensão por sua vez, aumenta a incidência de erros. A representação diagramática é mais eficiente, tende a ser menos ambígua do que a descrição narrativa e além disso demanda um tempo inferior para representar um sistema do que aquele que seria gasto para escrever um documento

narrativo contendo a mesma quantidade de informação. Diagramas claros são facilmente compreendidos e passam a ter um papel importante no projeto e análise de sistemas complexos.

Com o propósito de sistematizar a escolha do método de representação mais apropriado aos propósitos de análise, a seção 2 propõe uma Interface Genérica, que representa as situações de transição mais freqüentemente encontradas nas interfaces usuário-computador. A partir desta Interface Genérica, e com base nos critérios definidos por Palanque [PALANQUE 94], a seção 3 apresenta os resultados do estudo realizado para seleção do método mais apropriado para representação de interfaces. De acordo com este estudo, concluiu-se que o formalismo de representação mais adequado aos propósitos de avaliação é aquele proposto nas redes de Petri, apresentado na seção 4. O artigo conclui, na seção 5, com a discussão do uso de redes de Petri na modelagem de interfaces para fins de avaliação e as direções a serem seguidas por este trabalho.

2. Proposta de uma Interface Genérica

A dificuldade na escolha de um formalismo para representação das interfaces começa no entendimento de sua funcionalidade. A diversidade de dispositivos e mecanismos de interação dificulta a modelagem de suas características e comportamento. Com o objetivo de representar de forma sintética as ações mais comuns de uma interface com o usuário, independentemente dos dispositivos de interação, esta seção propõe uma interface genérica, que servirá como referência para comparação de formalismos para representação e modelagem de interfaces encontrados na literatura.

A independência de dispositivos significa que uma ação solicitada pelo usuário resulta na execução de um procedimento que independe do dispositivo ou mecanismo de interação utilizado. Assim, se um usuário solicita a ação *undo* através de um comando no teclado, um comando de voz, a seleção de um item de menu ou o acionamento de um botão, a ação resultante é a mesma, ou seja *undo*.

A transição entre pontos da interface pode se dar de forma direta ou indireta, condicional ou incondicional mantendo o estado atual, causando uma mudança de estado ou ainda restaurando o contexto da interação ao estado anterior.

A ligação indireta entre pontos significa que para se deslocar entre eles é necessário que estágios intermediários da interação sejam percorridos de modo a construir o contexto necessário a transferência do controle para o ponto destino. Na transição direta entre pontos a transferência de controle independe destes passos intermediários.

A partir do estudo de interfaces realizado foi possível sintetizar as seguintes situações: retorno na interação, deslocamento incondicional, deslocamento condicional, retorno ao ponto inicial da interação e saída do sistema. Estas situações representam de forma genérica a transição entre pontos da interface durante a interação.

Retorno na interação - a operação *undo* consiste no deslocamento do ponto atual da interação para um ponto anterior recuperando o contexto original e pode ocorrer de duas formas: *Undo* Simples - deslocamento do ponto atual da interação para o antecessor imediato restaurando o contexto original e devolvendo-lhe o controle da interação. *Undo* Sucessivo - deslocamento do ponto atual da interação para um ponto antecessor da interação a partir de uma seqüência de *undo* simples ao longo de um caminho da interação, restaurando passo a passo o contexto de cada ponto predecessor até devolver o controle para o ponto desejado, e tendo como limite o ponto inicial da interação.

Deslocamento incondicional - Esta solicitação resulta na transição para um outro ponto da interação independentemente do estado atual e transferindo-lhe o controle da interação.

Deslocamento condicional - Esta solicitação pode resultar em uma transição para o nó subsequente ou para um nó qualquer da interação, dependendo da condição testada ser ou não satisfeita. Esta transição significa a transferência do controle para o ponto destino da interface.

Retorno ao ponto inicial da interação - Esta solicitação resulta na transição para o ponto inicial da interação a partir de um ponto qualquer, mantendo o contexto atual e transferindo-lhe o controle.

Saída do sistema - Esta solicitação finaliza a interação com este sistema retirando-lhe o controle da interação.

A Figura 1 apresenta a representação da Interface Genérica. Nela estão representadas as situações: retorno na interação, deslocamento incondicional, deslocamento condicional, retorno ao ponto inicial da interação e saída do sistema.

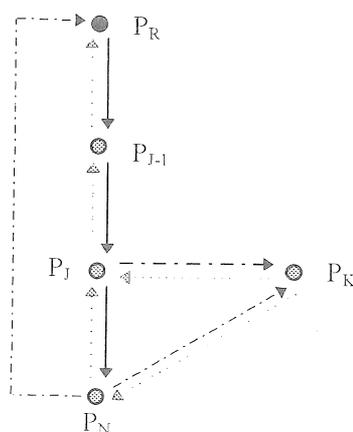


Figura 1: Diagrama da Interface Genérica

No Diagrama de Interface Genérica (DIG):

- representa os pontos da interação (nós do diagrama)
- representa o ponto inicial da interação (nó raiz)
- representa os caminhos entre pontos da interação
- ⋯→ representa os caminhos de retorno na interação
- - - - -> representa os caminhos de acesso direto a um ponto

a partir de um nó P_J do sistema, onde um nó representa um ponto na interação, é possível:

- Solicitar um serviço a um nó subsequente P_N .
- Retornar ao nó antecedente P_{J-1} através de *undo* simples.
- Retornar a um nó qualquer através de “*undo* sucessivo”.
- Retornar diretamente ao nó raiz P_R (transição direta para raiz).
- Solicitar um serviço diretamente a um nó P_K independentemente de sua posição relativa.

Na Figura 1, os seguintes caminhos de interação são representados:

⇒ Transição incondicional de um nó para outro:

$$P_R \longrightarrow P_{J-1} \longrightarrow P_J \longrightarrow P_N$$

Ex: Solicitação do próximo serviço disponível na seqüência da interação.

⇒ Transição condicional de um nó para um nó subsequente:

$P_R \longrightarrow P_{J-1} \longrightarrow P_J \dashrightarrow P_K$

Ex: Seleção de um item de uma lista.

⇒ *Undo* simples:

$P_R \longrightarrow P_{J-1} \longrightarrow P_J \dashrightarrow P_{J-1}$

Ex: Solicitação de retorno na interação para correção de um erro.

⇒ *Undos* sucessivos:

$P_R \longrightarrow P_{J-1} \longrightarrow P_J \longrightarrow P_N \dashrightarrow P_J \dashrightarrow P_{J-1} \dashrightarrow P_R$

Ex: Solicitação de retorno a um ponto para desfazer todo um trecho da interação.

⇒ Retorno direto à raiz:

$P_R \longrightarrow P_{J-1} \longrightarrow P_J \longrightarrow P_N \dashrightarrow P_R$

Ex: Retorno a janela principal a partir de um ponto da interação.

⇒ Ligação direta entre pontos:

$P_R \longrightarrow P_{J-1} \longrightarrow P_J \longrightarrow P_N \dashrightarrow P_K$

Ex: Solicitação de ajuda em um ponto qualquer da interação.

Na pesquisa por um método de representação formal capaz de modelar a interface genérica proposta verifica-se que este deverá ser capaz de representar:

- Transição entre pontos no diagrama
- Retornos a pontos anteriores do diagrama
- Situações de testes de condição para a execução de um determinado fluxo
- Estrutura hierárquica
- O tempo de ocorrência de eventos, ou o intervalo de tempo entre dois eventos.

Além de atender a estes critérios este método deverá possuir um formalismo, ser capaz de definir uma estrutura de dados e uma estrutura de controle; oferecer mecanismos de estruturação e ser capaz de representar o processamento paralelo. A partir deste referencial foi realizada a pesquisa descrita na próxima seção.

3. Pesquisa de um Método de Representação

Na busca por um método de representação formal das interfaces foram pesquisadas várias técnicas de diagramação uma vez que estas envolvem a combinação de notações gráficas e narrativas, aumentando a facilidade de compreensão da tarefa. Estas representações foram priorizadas porque: tendem a ser menos ambíguas do que a descrição narrativa, tendem a ser mais compactas, e demandam um tempo bem menor do que seria gasto para escrever um documento narrativo contendo a mesma quantidade de informação. Estas técnicas possibilitam os responsáveis pelo desenvolvimento a manipulação de um grande volume de detalhes, gerados durante o processo de desenvolvimento e avaliação dos sistemas.

De acordo com o estudo realizado em [SOUSA 96], a maioria dos métodos de representação pesquisados foi incapaz de representar todas as características da Interface Genérica ou atendiam apenas parcialmente aos demais critérios estabelecidos. O resultado desta análise é apresentado na Tabela 1. Nesta tabela são representados apenas dois métodos de diagramação além de redes de Petri, considerados mais apropriados para a modelagem de interfaces.

CAPACIDADE DE REPRESENTAÇÃO	Deslocamento incondicional	Retorno na interação	Transição direta entre pontos	Deslocamento condicional	Mecanismo de estruturação	Rep. de aspectos estruturais	Rep. de aspectos comportamentais	Descrição de proces. paralelo	Representação de concorrência	Rep. do tempo
Diagrama de Fluxo de Dados	S	S	S	S	S	S				
Fluxogramas	S	S	S	S	S	S				
Redes de Petri	S	S	S	S	S	S	S	S	S	S

Tabela 1 - Adequação dos Métodos de Representação

Os Diagramas de Fluxo de Dados (DFD), apesar de serem capazes de representar os caminhos do fluxo de informações através da interface, não são adequados para representar situações de transição condicional, o processamento paralelo, concorrência e o tempo de ocorrência de eventos, não sendo portanto adequados para uma representação completa de sistemas de interface. A tentativa de representar o DIG, segundo o diagrama de fluxo de dados é apresentada na Figura 2.

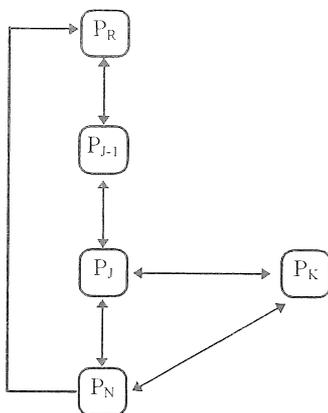


Figura 2: Representação da Interface Genérica em DFD

Os Fluxogramas, tornam inviável a representação de retornos na interação e a transição direta entre pontos gerando uma sobrecarga visual que inviabiliza a análise pretendida. Os fluxogramas são limitados à representação de pequenos sistemas e não possuem técnicas de análise automatizadas, o que torna difícil o processo de avaliação, sobretudo para os sistemas de interface com suas muitas opções e um grande número de pontos. A representação da Interface Genérica em fluxograma é mostrada na Figura 3, onde já se percebe dificuldade para a representação de todas as possibilidades de interação.

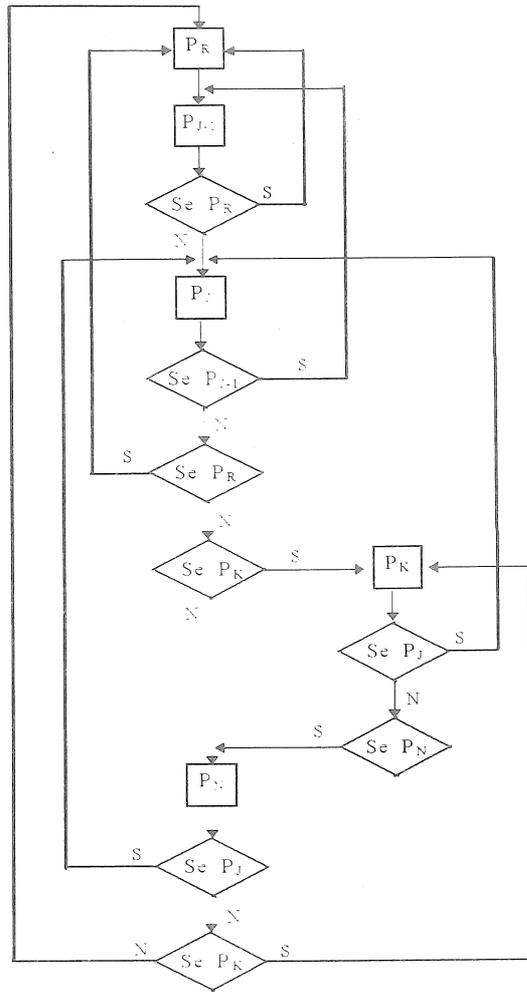


Figura 3: Representação da Interface Genérica em Fluxograma

Como pode ser observado, dentre as técnicas pesquisadas o Diagrama de Fluxo de Dados [MARTIN 91] e os Fluxogramas [MARTIN 91], apesar de serem capazes de modelar aspectos estruturais das interfaces, não são capazes de representar seus aspectos comportamentais. Assim, foi analisada a representação de interfaces utilizando-se redes de Petri, que além dos recursos para diagramação consiste de um formalismo matemático e dispõe de ferramentas automáticas para análise do sistema modelado. A representação de interfaces através de redes de Petri é tratada na próxima seção.

4. Representação da Interface Genérica em redes de Petri

Redes de Petri (PN - Petri Nets) é uma ferramenta gráfica e matemática de modelagem que pode ser aplicada a diversos tipos de sistemas [JENSEN 92][PETERSON 81][MURATA 89]. Aplica-se apropriadamente a sistemas assíncronos e com alto índice de concorrência ou paralelismo. Através da utilização de PN, é possível estudar a estrutura e o comportamento dinâmico dos sistemas.

Uma rede de Petri pode ser vista como um grafo mais uma marcação inicial (que representa um estado inicial). A representação gráfica de redes de Petri consiste de duas classes de nós, denominadas de lugares e transições, onde arcos são utilizados para conectar os lugares e as transições. Os lugares são representados por círculos e as

transições por barras ou retângulos. Podem existir arcos partindo dos lugares para as transições ou arcos partindo das transições para os lugares. Um arco partindo do lugar p_i para uma transição t_j define o lugar como sendo uma entrada para a transição. Um lugar de saída é identificado através de um arco partindo da transição para o lugar. Múltiplas entradas ou saídas podem ser representadas através de múltiplos arcos ou através de um único arco com um peso k .

Os lugares podem conter fichas (desenhadas como pontos negros), onde o estado atual do sistema modelado (a marcação) é identificada através do número de fichas em cada um dos lugares. Através do disparo das transições, há a possibilidade de observar-se o comportamento dinâmico do sistema. Quando as transições disparam, o estado do sistema se modifica (a marcação da rede de Petri) e ocorre o fluxo de fichas ao longo da rede. Para que ocorra o disparo de uma transição é necessário que esta esteja habilitada, o que ocorre se todas as pré-condições forem satisfeitas (a existência suficiente de fichas disponíveis nos lugares de entrada). Quando do disparo das transições, fichas são removidas dos lugares de entrada e depositadas nos lugares de saída. O número de fichas removidas/depositadas depende do peso de cada arco. Os disparos iterativos das transições e as subsequentes marcações da rede são semelhantes a um "jogo de fichas", onde as fichas simulam o comportamento dinâmico e concorrente do sistema.

Redes de Petri possuem uma modelagem matemática capaz de descrever o comportamento do sistema modelado. A análise pode ser realizada através de equações de estado e grafos de alcançabilidade [MURATA 89]. O grafo de alcançabilidade mostra o comportamento dinâmico do sistema, enfatizando todas as possíveis marcações a partir da marcação inicial, bem como todos os possíveis disparos a partir de cada marcação.

Através de ferramentas matemáticas [JENSEN 92], é possível realizar uma análise automática do sistema, sendo possível localizar entre outras situações: *dead-lock* e conflitos, encontradas em sistemas concorrentes. A análise é feita com base em um conjunto de propriedades das redes que são classificadas em propriedades comportamentais, aquelas que dependem do estado inicial do sistema, e propriedades estruturais, aquelas que independem do estado inicial.

A figura 4 apresenta a modelagem da Interface Genérica em redes de Petri.

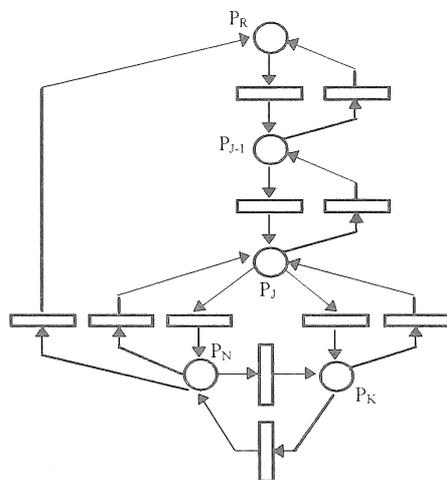


Figura 4: Representação da Interface Genérica em redes de Petri

Neste trabalho, no que diz respeito ao Diagrama de Interfaces Genérica, restringiremos a discussão às propriedades de PN a Alcançabilidade, Vivacidade, Reversibilidade e Limitação que exemplificam o potencial para análise utilizando-se redes de Petri.

Analisando-se a **alcançabilidade** de uma rede pode-se verificar se é possível alcançar um determinado estado a partir do estado inicial. Assim é possível analisar a existência de acessos entre estados da rede e o seu estado inicial. Do ponto de vista da análise das interfaces deve-se assegurar a existência de acessos entre os vários pontos da interação, uma vez que deve-se assegurar ao usuário o acesso a qualquer um dos pontos da interação.

Analisando-se a **vivacidade** de uma rede pode-se verificar a existência de bloqueios. Assim é possível analisar a existência de acessos entre estados da rede e o estado inicial. Do ponto de vista da análise das interfaces deve-se assegurar que não haja situações em que estando em um ponto na interação o usuário fique impossibilitado de deslocar-se para um outro ponto (direta ou indiretamente), uma vez que o usuário deve ter acesso a todos os pontos da interação.

Analisando-se a **limitação** de uma rede pode-se verificar se as opções de interação estão limitadas a um número finito. Do ponto de vista da análise das interfaces deve-se assegurar que o alcance das funções na interação está limitado a um certo número de passos, a exemplo da função de retorno na interação.

Analisando-se a **reversibilidade** de uma rede pode-se verificar se é sempre possível retornar à situação anterior da interação. Do ponto de vista da análise das interfaces deve-se assegurar que haja meios, a exemplo da função *undo*.

As redes de Petri apresentam-se como a forma mais adequada para representação de sistemas de interface, mas apresentam limitações no seu universo de modelagem. Além da representação dos aspectos estruturais e comportamentais das interfaces, para que seja possível representar dados oriundos da avaliação dinâmica (comportamental), se faz necessária a representação do tempo associado à ocorrência de eventos. Conforme a semântica de redes de Petri clássicas, uma transição é instantânea, não cabendo portanto a representação do tempo. Existem, no entanto, extensões de redes de Petri: redes de Petri temporais e temporizadas, em que parâmetros de tempo são incorporados ao modelo, associados aos lugares ou transições.

A utilização de redes de Petri clássicas para a modelagem de sistemas de interface, que possuem um grande número de componentes idênticos, tais como janelas, botões, menus, etc.; levam a existência de redundâncias no modelo quando da representação destes elementos, uma vez que a única maneira de diferenciar dois componentes idênticos é através da especificação de uma estrutura idêntica de sub-rede para cada componente. Uma importante extensão é a introdução de *fichas individuais*, que permitem às fichas carregar *diferentes tipos de informação*. Com esta extensão, pode-se modelar o componente comum apenas uma vez e associar-se diferentes fichas para cada componente idêntico, o que conduz as redes de Petri coloridas. A grande vantagem em utilizar-se redes coloridas, reside na sua capacidade de simplificação da representação da rede, uma vez que interfaces reais possuem complexidade e tamanho bastante consideráveis.

5. Considerações Finais

Este trabalho apresentou redes de Petri como o método mais apropriado para a modelagem das interfaces. Embora esta abordagem já tenha sido utilizada em outros trabalhos de interface [ROSIS 96][PALANQUE 95], o propósito da modelagem das interfaces neste trabalho foi o de introduzir um formalismo no processo de avaliação das interfaces que possibilitasse a análise do seu comportamento. Embora tendo se mostrado adequado na modelagem da Interface Genérica, atualmente estão sendo desenvolvidos estudos que procuram validar a adequação das redes de Petri no processo de análise das interfaces em várias etapas de seu ciclo de vida. A validação das interfaces consiste em mapear dados obtidos do processo avaliatório nestas diferentes fases. Na fase de especificação os dados são obtidos de simulações com o modelo. Na fase de análise dados são obtidos sobre o desempenho dos usuários interagindo com protótipos ao longo do desenvolvimento do produto. É possível ainda obter dados sobre a avaliação do produto final.

Os estudos em andamento buscam determinar a adequação de redes de Petri durante um processo de avaliação de interfaces, visando analisar os caminhos disponíveis para a interação, diagnosticar o excesso ou ausência de

alternativas de acesso a um determinado ponto da interface, e ainda identificar a partir das propriedades das redes situações tais como: bloqueios ainda na fase de especificação. Por outro lado, pretende-se mapear no modelo os dados oriundos do registro do desempenho do usuário durante uma interação com um protótipo, ou mesmo com o produto final, de modo a averiguar aspectos comportamentais do sistema tais como: os caminhos mais utilizados na realização de uma tarefa e obter informações tais como: a frequência de utilização das funções disponíveis, a incidência de erros, a frequência de solicitação de ajuda e o tempo gasto na realização de tarefas específicas. Esta análise possibilitará a verificação da qualidade dos mecanismos de navegação disponibilizados na interface.

6. Referências Bibliográficas

[DUMAS 94] Dumas, Joseph S. e Redish, Janice C. A practical guide to usability testing. Alex Publishing Corporation, 412p, 1994.

[JENSEN 92], Jensen, Kurt. Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1. Basic Concepts. Monographs in Theoretical Computer Science. Springer-Verlag, 1992.

[MARTIN 91] Martin, James e McClure, Carma. Técnicas Estruturadas e Case. MAKRON Books do Brasil Editora Ltda., São Paulo, 1991.

[MURATA 89] Murata, Tadao. Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE. Vol 77, No 4, abril 1989.

[NIELSEN 93] Nielsen, Jakob. Usability Engineering. Academic Press, Cambridge, MA, 1993.

[PALANQUE 94] Palanque P., Bastide R. Theoretical Foundations of recent formal approaches in HCI design. Research Symposium CHI'94, Boston, 23-30 April 1994.

[PALANQUE 95] Palanque P., Bastide R. A Petri Net Based Environment for the Design of Event-Driven Interfaces. Lecture Notes in Computer Science n° 935, p66-83, Springer-Verlag, Torino, Italy, June 1995.

[PETERSON 81] Peterson, James L. Petri Net Theory and the Modeling of Systems. Prentice-Hall, Inc., Englewood Cliffs, 290p, 1981.

[ROSIS 96] Rosis, Fiorella de, Pizzutilo, Sebastiano e Carolis, Berardina. A Tool to Support Specification and Evaluation of Context-Customized Interfaces. HCI in Italy, July 1996. Volume 3, pg 82 a 88.

[SCAPIN 92] Scapin, Dominique L. e Bastien, C. A validation of Ergonomic Criteria for the evaluation of Human-Computer Interfaces. International Journal of Human-Computer Interaction, 1992.

[SOUSA 96] Sousa, Marckson R. F. e Turnell, Maria de F. Q. V. Coletor de Dados Dinâmicos gerados a partir da interação do usuário com uma interface no ambiente UNIX. Relatório Técnico da COPELE/CCT/UFPB, Fevereiro/96, 70p.