

# MODELAGEM DA NAVEGAÇÃO DE INTERFACES COM O USUÁRIO DE SISTEMAS DE AUTOMAÇÃO INDUSTRIAL

ALEXANDRE SCAICO, MARIA DE FÁTIMA Q. V. TURNELL<sup>1</sup>, ANGELO PERKUSICH<sup>1</sup>

<sup>1</sup>*Departamento de Engenharia Elétrica, Universidade Federal da Paraíba  
Caixa Postal 10105 - Campina Grande - Paraíba - Brasil  
E-mails: {scaico,turnellm,perkusic}@dee.ufpb.br*

**Resumo**— Este trabalho apresenta e discute a aplicação de um modelo de navegação ao projeto de interfaces com o usuário. O modelo, construído em Redes de Petri Colorida (CPN) [2], apóia a análise do componente de navegação na interface. O trabalho discute os resultados do uso do modelo de navegação a partir de um estudo de caso relativo a um sistema de controle supervisory para plantas industriais.

**Abstract**— This paper presents and discusses the application of a navigation model in human interface design. The model, built with Coloured Petri Nets (CPN), support the analysis of the navigation component of the interface. The paper discusses the results of using this navigation model in a supervisory control system to industrial application case study.

**Keywords**— User interface, Coloured Petri Nets, supervisory control, navigation.

## 1 Introdução

Na busca pela qualidade da interação usuário computador o projetista necessita considerar as habilidades e limitações do usuário ao escolher os recursos que serão oferecidos pelo componente interativo do sistema. Por sua vez, diante do número crescente de opções de implementação, os projetistas necessitam avaliar soluções alternativas antes da fase de implementação.

A tarefa de concepção do componente de interface do usuário passa também pela compreensão da funcionalidade a ser oferecida, além do como será oferecida.

Turnell e Sousa em [7,8] propuseram um método de projeto de interface do usuário que inicia pela identificação do perfil do usuário seguido da análise da tarefa. A análise da tarefa [4] consiste de um estudo detalhado para determinar a natureza, propósito e componentes (subtarefas e estados) da tarefa, bem como a seqüência de execução. A seqüência de execução é particularmente importante por representar o plano de ação (ou cenário) adotado pelo usuário para resolver um problema. Neste método é proposto um modelo para representar a interação que especifica, entre outros aspectos, a estrutura de navegação da interface. A estrutura de navegação, modelada em CPN, é apresentada em [7,8]. Este modelo é verificado de acordo com um conjunto de propriedades especificadas para garantir a usabilidade de um sistema que dele origine, tais como a reversibilidade das ações do usuário em um contexto específico e a garantia de acesso a pontos específicos na seqüência de interação. Ainda no método proposto, a cada passo de concepção é associada uma etapa de avaliação com o propósito de investigar a adequação da especificação ao perfil do usuário, à descrição da tarefa e a critérios ergonômicos. O último passo do método consiste na construção de um protótipo que deve ser validado com o usuário a partir de testes de usabilidade.

O uso de métodos formais no projeto de interfaces com o usuário permite a investigação do comportamento da interface através de um modelo matemático ainda no início da fase de projeto. Trabalhos anteriores já

atestaram a adequação das Redes de Petri (PN) [3] enquanto ferramenta para modelagem de interface com o usuário [1,4,5].

O modelo proposto em [7,8] representa uma estrutura genérica de navegação encontrada nas interfaces com usuário. O modelo foi originalmente utilizado para representar a navegação na interface com um *browser* para internet. Com o propósito de avaliar o caráter genérico da estrutura deste modelo, este trabalho apresenta o resultado de sua aplicação ao contexto das interfaces com o usuário de sistemas industriais.

Apesar da semelhança entre a interação do usuário com sistemas industriais e com interfaces de outros tipos de aplicações, um estudo mais detalhado das primeiras levam a concluir que há diferenças significativas. O volume de informação apresentado ao operador da planta industrial é tipicamente mais alto, impondo uma grande carga cognitiva. Ademais o operador deve reagir a eventos e completar tarefas levando em consideração o prazo final e restrições de segurança [9]. Portanto, a qualidade do projeto da interação do ponto de vista da navegação pode comprometer a eficiência de execução da tarefa, o desempenho global do sistema de automação e aspectos de segurança.

Este trabalho está organizado em 5 seções. Na seção 2 é descrita a interface com o usuário do sistema industrial utilizada neste trabalho. Na seção 3 é apresentada a aplicação do modelo de navegação a interface industrial. Na seção 4 é apresentada a análise do modelo apoiada pela ferramenta *Design/CPN* [10], que apóia a construção, simulação e análise de modelos em CPN. O trabalho conclui com uma discussão da aplicabilidade do modelo à representação da interface com o usuário de sistemas industriais, e propõe direções futuras para este trabalho.

## 2 Interface Industrial Modelada

A interface com o usuário utilizada como estudo de caso é baseada no demo distribuído junto com o softwa-

re ©InTouch<sup>1</sup>. A escolha por essa interface é devido a ela apresentar os tipos de navegação e os recursos típicos encontrados em interfaces industriais. O demo ilustra os recursos disponíveis ao operador de uma planta química cujo processo consiste na produção e armazenagem de um material radioativo. A interface é composta de nove janelas nas quais o operador pode definir variáveis, navegar entre as janelas, e monitorar o estado de vários aspectos da planta através de informações textuais e gráficas. A produção da planta consiste na mistura de um concentrado químico em um reator, seguida pela armazenagem do produto final em barris. O operador, neste processo, realiza a monitoração do processo por meio de um sinótico animado da planta, e através de diagramas de tendência relacionados com as variáveis sob controle. Através do programa supervisorio é possível controlar o valor percentual dos componentes da mistura; o enchimento e armazenagem dos barris; e o tratamento de alarmes que ocorrem durante a operação da planta. As janelas são descritas a seguir.

- *JanReactor*: Janela principal do sistema. Apresenta um sinótico animado da planta industrial. Nesta janela é apresentada a concentração atual dos elementos químicos utilizados no processo de mistura, a temperatura da mistura e o nível de armazenagem no reservatório onde a mistura final é armazenada. Nesta janela, o operador interage com o sistema através de seis botões usados principalmente para a navegação entre janelas.
- *JanReactorHelp*: Janela de ajuda que é sobreposta a *JanReactor*. Apresenta um texto de ajuda sobre o sinótico.
- *JanBatch*: Janela onde o usuário define o nível de concentração através de uma barra deslizante e de um conjunto de botões para ajuste fino. Esta janela se sobrepõe à janela originalmente ativa na tela.
- *JanBatchSet*: Janela onde o usuário pode (alternativamente) definir o nível de concentração da mistura utilizando o teclado. Quando está ativa se sobrepõe à janela *JanBatch*.
- *JanHist*: Janela que mostra um gráfico de histórico sobre o processo. Apresenta botões para navegação e barras deslizantes para ajuste das características do gráfico.
- *JanHistHelp*: Janela de ajuda sobre a janela *JanHist*, que estando ativa se sobrepõe a janela *JanHist*.
- *JanMaintenance*: Janela que mostra o estado de alguns aspectos da planta. Esta janela se sobrepõe à janela originalmente ativa na tela.
- *JanConveyor*: Esta janela apresenta, através de animação, o enchimento dos barris com a mistura final. Ela oferece um botão para seleção entre o enchimento automático ou manual, botões para navegar entre as janelas, e botões para controlar as variáveis do processo a exemplo da velocidade da esteira que transporta os barris.
- *JanAlarme*: Apresenta as mensagens de alarme do processo. Oferece botões para o reconhecimento dos alarmes individualmente ou em grupo, e botões para a navegação entre as janelas. Esta janela se sobrepõe à janela originalmente ativa na tela.

### 3 Modelo CPN da Interface Industrial

No modelo de navegação apresentado em [7,8], foi considerado que todos os objetos utilizados na interação, a exemplo de botões, estão permanentemente ativos e, portanto, disponíveis ao usuário. No modelo aplicado ao contexto industrial, apresentado na Fig. 3, foi atribuído um estado aos objetos da interface (ativo/desativo) de modo a representar objetos cujo estado são sensíveis ao contexto da interação. Também foram modificados os tipos de navegação disponíveis para contemplar os casos que existem na interface industrial e não foram abordados no modelo proposto em [7,8].

#### 3.1 Descrição do Modelo

O novo modelo de navegação aplicado à interface com o supervisorio é apresentado na Fig.3, e seu nó de declaração é mostrado a seguir.

```

color Janela = with JanReactor | JanHist |
JanMaintenance | JanBatch | JanHist |
JanMaintenance | JanConveyor | JanAlarme |
JanBatchSet | JanReactorHelp | JanHistHelp;
color Botao = with BotReactorHelp | BotBatchExit |
BotBatchSet | BotBatchOK | BotBatchUp |
BotBatchDown | BotBatchBar | BotMaintExit |
BotAlarmeConc | BotAlarmeTemp |
BotAlarmeNivel | BotReactor | BotBatch |
BotHist | BotMaint | BotConveyor |
BotEndProgram | BotBatchSetOK |
BotReactorHelpExit | BotConveyorMA |
BotConveyorEst1 | BotConveyorEst2 |
BotConveyorEjet | BotConveyorBar |
BotHistCTime | BotHistDefault | BotHistV100 |
BotHistV50 | BotHist3m | BotHist1m |
BotHistBarVert | BotHistBarHor | BotHistHelp |
BotAlarme | BotHistHelpExit | BotAlarmeClose |
BotAlarmeAckAll;
color Status = with ativo | desativo;
color Botao_Status = product Botao*Status;
color Opcoes = product Janela*Botao_Status;
color Lista_Botao_Status = list Botao_Status;
color Interface=product Janela*Lista_Botao_Status;
color Lista_Interface = list Interface;
color Prob_Alarme = bool;
color Qual_Alarme = int with 1..3;
var y: Prob_Alarm; var Jan, Jan1: Janela;
var Bot, Bot1:Lista_Botao_Status; var act: Botao;
var status, status1: Status; var lista: Lista_Interface;
var ProbAl: Prob_Alarme; var t: Qual_Alarme;
var interface: Interface; var bot_stat: Botao_Status;

```

<sup>1</sup> Software construtor de Interfaces para Programas Supervisorios distribuído pela *Wonderware Corporation*.

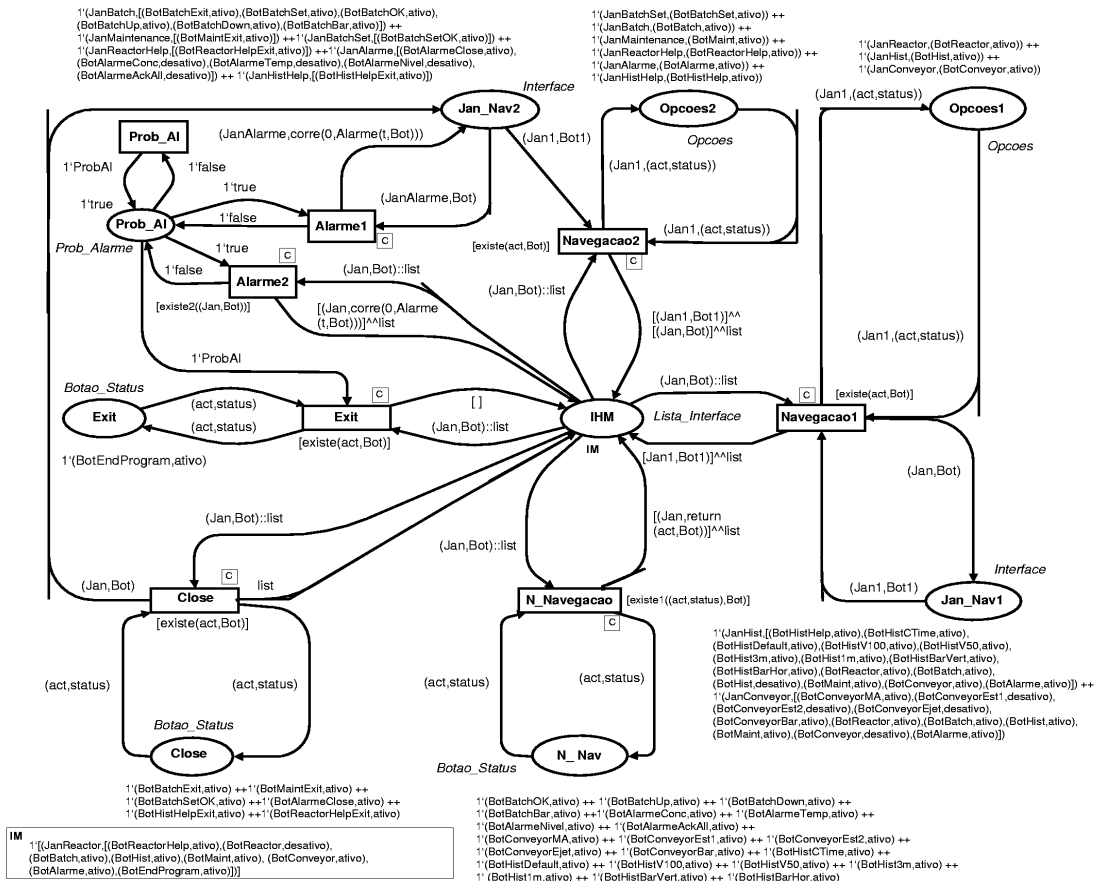


Figura 1. Modelo de Navegação da Interface Industrial

O significado de cada cor é descrito a seguir:

- **Janela:** conjunto com todas as janelas da interface.
- **Botao:** conjunto com todos os botões da interface.
- **Status:** associado aos botões, pode ser ativo ou desativo.
- **Botao\_Status:** par que representa o nome de um botão e seu respectivo estado.
- **Opcoes:** tripla (janela, botão, status) que associa o nome de uma janela ao nome de um botão e seu respectivo estado.
- **Lista\_Botao\_Status:** lista de elementos do tipo Botao\_Status.
- **Interface:** produto, com o nome de uma janela como primeiro elemento, e uma lista do tipo Lista\_Botao\_Status como segundo elemento.
- **Lista\_Interface:** lista de elementos do tipo Interface.
- **Prob\_Alarme:** valor booleano relativo à probabilidade de ocorrência de um alarme.
- **Qual\_Alarme:** inteiro, definido entre 1 e 3, que identifica o tipo de alarme.

O lugar **IHM** contém a informação sobre qual é a janela ativa. No caso representado a janela ativa inicial é *JanReactor*. A partir deste lugar na rede pode ocorrer um dos cinco tipos de navegação a seguir:

### Navegação Tipo 1

Consiste na navegação entre janelas, com o fechamento da janela ativa e a abertura da janela solicitada. Neste caso, a lista no lugar **IHM** possui apenas um elemento, o qual deve ser substituído por um novo elemento. A transição **Navegacao1** e os lugares **Jan\_Nav1** e **Opcoes1** modelam este tipo de navegação.

A guarda (restrição à ocorrência) desta transição é a função "existe", que verifica se a ação de navegação selecionada está disponível na janela ativa. Ou, em outras palavras, se a ação é permitida nesta janela.

### Navegação Tipo 2

Navegação entre janelas, com a janela requisitada se sobrepondo à janela ativa, tornando-se a nova janela ativa. A nova janela ativa é então adicionada à lista de interface.

A transição **Navegacao2** e os lugares **Jan\_Nav2** e **Opcoes2** modelam esta navegação. A guarda desta transição é a função "existe".

### NãoNavegação

Navegação entre opções de uma mesma janela ativa. A transição **N\_Navegacao** e o lugar **Bot\_N\_NAV** modelam esta situação. As fichas no lugar **Bot\_N\_NAV** determinam as ações de navegação em uma janela. Note que todas as fichas estão no estado ativo.

A guarda desta transição é a função "**existe1**", que verifica se a ação selecionada está disponível na janela ativa, e se seu estado é ativo (i.e. disponível para o usuário). Esta transição conduz a quatro ações possíveis de serem executadas. Qual ação será selecionada é decidida pela função "**retorno**" o arco que conecta a transição ao lugar **IHM**:

Se a ação a ser executada for a seleção do botão *BotConveyorMA*, os botões *BotConveyorEst1*, *BotConveyorEst2* e *BotConveyorEjet* terão seus estados invertidos (ativo/desativo ou vice-versa). Esta mudança é implementada pela função "**muda\_status2**".

Se a ação a ser executada for a seleção do botão *BotAlarmeConc* ou, *BotAlarmTemp* ou, *BotAlarmeNivel*; o botão selecionado será desativado. Esta mudança ocorre através da função "**muda\_status**".

Se a ação for a seleção do botão *BotAlarmeAckAll* (disponível apenas quando os três tipos de alarmes ocorrerem), os botões *BotAlarmeAckAll*, *BotAlarmeConc*, *BotAlarmTemp* e *BotAlarmeLevel* serão desativados. Esta mudança ocorre através da função "**muda\_status1**".

Se a ação for a seleção de um botão que não provoca a mudança do seu estado ou de outros botões, a informação é devolvida inalterada ao topo da lista, indicando apenas que uma seleção foi feita.

#### Fechamento

Consiste no fechamento da janela atualmente ativa, e sobreposta a uma outra janela. Esta navegação é modelada pela transição **Close** e pelo lugar **Bot\_Close**. Neste lugar estão fichas que relacionam uma ação de navegação a uma janela indicando qual janela deve ser fechada se esta opção de navegação for selecionada. A guarda desta transição é a função "**existe**".

#### Finalização

A transição **Exit** e o lugar **Bot\_Exit** modelam este tipo de navegação. Quando esta transição é disparada a execução do aplicativo é encerrada. No lugar **Exit** existe apenas uma ficha que representa a ação de navegação de encerramento do aplicativo.

### 3.2 Geração dos Alarmes Aleatórios

Foram criadas funções para simular os alarmes que ocorrem na planta industrial. Neste estudo de caso existem três tipos de alarme: um relativo ao nível de concentração da mistura, um relativo ao nível de temperatura do processo e um relativo ao nível da mistura. Na ocorrência de um alarme, o botão correspondente na janela de alarmes se torna ativo. Os alarmes são modelados pelas transições **Prob\_AI** (que restringe a ocorrência dos alarmes), **Alarme1** (que modela a ocorrência dos alarmes com a janela de alarme desativa) e **Alarme2** (que modela a ocorrência dos alarmes quando a janela de alarme é a janela ativa), e pelo lugar **Prob\_AI**.

### 3.3 Síntese do Modelo

Em suma, no modelo apresentado, para que ocorra uma navegação o objeto de interação correspondente deve

estar estivar disponível e ativo na janela ativa. O ciclo de simulação do modelo é intercalado pela ocorrência de alarmes até a requisição de saída do sistema.

## 4 Análises no Modelo

Entre os vários aspectos de interesse para análise da navegação este trabalho se concentra na análise das seqüências de interação disponíveis ao usuário. O objetivo da análise da navegação é identificar falhas de projeto tais como impasses (*deadlocks*), inexistência de caminhos de acesso entre estados da interação, superestimação ou subestimação de caminhos entre estados de interação, otimizar os caminhos de acesso entre estados (reduzindo sua extensão) e identificar situações de inexistência de caminhos para a reversão de ações na interface (retorno a um estado anterior).

Foram realizadas várias análises no modelo para verificar a sua correteza lógica e seu comportamento. Foram realizadas simulações interativas e automáticas, gerado o grafo de ocorrência (Occ) [2], e gerados Gráficos de Seqüência de Mensagens (MSCs) [6] para mostrar graficamente um cenário de navegação na interface. As análises realizadas com o auxílio da ferramenta *Design/CPN*, são descritas a seguir.

### 4.1 Propriedades de Navegação do Modelo

Esta seção apresenta informalmente, um conjunto de propriedades do modelo que estão associadas à usabilidade da interface, do ponto de vista da navegação. Estas propriedades são utilizadas na análise do modelo de navegação.

Dado que a navegação em uma interface com um sistema supervisorio é representada pelo modelo CPN descrito anteriormente, suas características podem ser expressas em termos de um conjunto de propriedades do modelo, como é ilustrado a seguir [7].

- **Reversibilidade** – É a possibilidade de retorno na interação, cancelando ações previamente realizadas. Não se aplica a todos os estados da interface. Verifica-se a reversibilidade através da propriedade de alcançabilidade da rede.
- **Existência de caminhos de acesso entre pontos específicos da interação** – Por razões de usabilidade, o acesso a pontos específicos (a exemplo dos mecanismos de ajuda) deve ser assegurado em qualquer ponto da interação. Este aspecto pode ser avaliado analisando as marcações da rede, investigando se a partir de um estado  $M_k$  é possível alcançar um estado  $M_j$ .
- **Reinicialização** – Possibilidade de retorno ao início da interação. Este comportamento é verificado através da análise das marcações da rede. Se o estado inicial estiver na lista das *home marking* do modelo, esta propriedade é satisfeita. Note que se todas as marcações são *home markings*, conclui-se que há caminhos de conexão entre todos os estados da interface.
- **Acesso a Saída** – Do ponto de vista de usabilidade, os sistemas devem oferecer mais de um acesso à saída do sistema. Para analisar o a-

cesso à saída é necessário verificar se todas as marcações mortas constantes no relatório do Occ correspondem ao estado de saída. A marcação morta deve sempre ser alcançável se existir acesso à saída a partir de qualquer ponto da interação. Nesta situação, a marcação de saída deve estar na lista das marcações mortas e na lista das *home markings*. Esta propriedade não se aplica a qualquer sistema.

#### 4.2 Simulações Interativas e Automáticas

As simulações interativas foram realizadas durante a construção do modelo para verificar a evolução de partes do modelo (a exemplo dos tipos de navegação entre janelas, a geração de alarmes, mudanças de estado na janela *JanConveyor*, e a existência de acesso a pontos de saída) e para detectar possíveis falhas no modelo. As simulações automáticas foram realizadas no modelo completo para verificar a evolução do sistema como um todo e verificar a existência de impasses, além daqueles relativos à situação de saída.

#### 4.3 Grafo de Ocorrência (Occ)

O Occ foi utilizado para analisar o comportamento lógico do modelo. Através dele verificou-se as propriedades de navegação anteriormente citadas.

Do ponto de vista da propriedade de vivacidade, foi observado que não existem transições mortas, garantindo que todas as transições podem disparar pelo menos uma vez (o que corresponde à garantia de acesso a todos os pontos da interface).

E, foi observado que existem 16 marcações mortas. Essas marcações indicam o estado do sistema quando do encerramento do aplicativo. Uma vez que existem 3 botões de alarmes os quais podem estar no estado ativo ou desativo e que os botões de **JanConveyor** podem estar no modo manual ou automático, há 16 possibilidades de estados no encerramento do aplicativo.

#### 4.4 Gráficos de Seqüência de Mensagens (MSCs)

O uso de métodos formais na concepção e análise de sistemas interativos ainda é um fato restrito ao ambiente acadêmico. Os projetistas não estão familiarizados com a especificação de interfaces em Redes de Petri. Assim, a visão do comportamento dos sistemas através de MSCs é uma abordagem adequada. A representação de um evento por meio de uma mensagem é muito mais intuitiva que a análise das marcações de um modelo CPN.

Gráficos de Seqüência de Mensagens (MSC) consistem em uma linguagem gráfica e textual para a descrição e especificação de interações entre componentes de um sistema. Eles descrevem a comunicação assíncrona entre instâncias e podem ser utilizados como uma especificação geral do comportamento de comunicação de sistemas em tempo real [6]. Uma instância é uma entidade abstrata na qual pode-se observar parte da interação com outras instâncias ou com o meio-ambiente. No MSC uma instância é denotada por uma

barra vertical. O tempo ao longo de cada eixo transcorre do topo para a base [6].

Neste contexto os MSCs foram utilizados para propósito de validação dos cenários de navegação concebidos para a interface modelada. O MSCs foi particularmente útil na verificação do comportamento do sistema em condições específicas tais como: ocorrência de alarmes, navegação entre pontos da navegação, alternância no modo de enchimento (automático/manual); dando uma visão geral do comportamento do sistema.

Os MSCs representados nas Figs. 4 e 5 ilustram o comportamento do sistema em um possível cenário de navegação. Eles representam o comportamento da comunicação entre as instâncias **Jan\_Nav1**, **Jan\_Nav2** e **IHM**, tornando possível à análise do comportamento do sistema a partir dos seguintes pontos de vista: tipos de navegação entre janelas, geração e reconhecimento de alarmes, navegação dentro de uma mesma janela e saída do aplicativo.

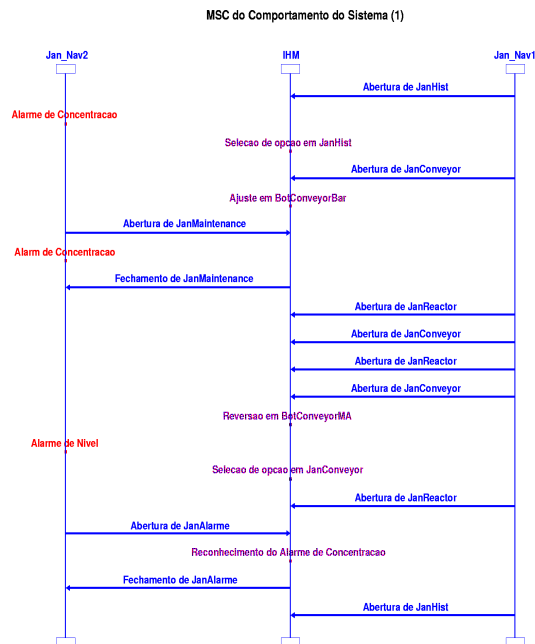


Figura 2. MSC do comportamento do Sistema (MSC1)

No MSC1, o operador entra no programa supervi-sório na janela *JanReactor*. Em seguida checa o gráfico de tendência do processo abrindo *JanHist*, onde a opção de zoom é selecionada. Neste instante ocorre um alarme de concentração da mistura. O operador ignora esse alarme e abre a *JanConveyor* e para ajustar a velocidade de enchimento dos barris. Note que os alarmes são gerados aleatoriamente durante a análise, e a decisão de quando reconhece-los é uma prerrogativa do operador.

Em seguida verifica o estado das partes do sistema em *JanMaintenance*. Um novo alarme de concentração é gerado. O operador então fecha *JanMaintenance* e alterna a visão entre *JanReactor* e *JanConveyor*, mudando o processo de enchimento dos barris do modo automático para manual. Neste ponto o sistema gera um alarme de concentração. Como o processo de enchimento está no modo manual, o operador deve retornar e dar continuidade ao enchimento dos barris. Durante o enchimento, o

operador retorna a JanReactor, abre Jan-Alarme e reconhece o alarme de concentração. Em seguida fecha JanAlarme, que estava sobreposta a JanReactor, e abre JanHist para checar o gráfico de tendência.

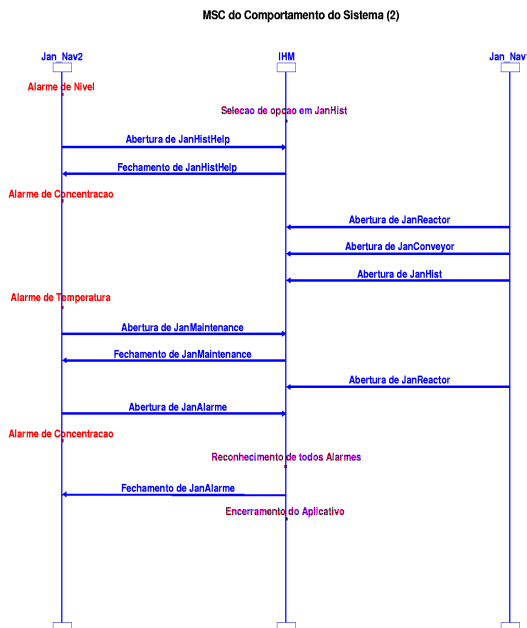


Figura 3. MSC do comportamento do Sistema (MSC2)

No MSC2 – Um novo alarme de nível é gerado. O operador amplia o gráfico de tendência e solicita ajuda, abrindo JanHistHelp. Ele então JanHistHelp e é gerado um alarme de concentração. O operador verifica o processo de mistura em JanReactor e o processo de enchimento em JanConveyor, em seguida abre JanHist quando é gerado um alarme de temperatura. O operador checa o estado das partes do sistema em JanMaintenance e volta a monitorar o processo de mistura em JanReactor. Abre JanAlarme quando é gerado um alarme de concentração. Finalizando, o operador reconhece todos os alarmes, fecha JanAlarme e sai do sistema.

## 5 Considerações Finais e Direções Futuras

O modelo CPN de navegação apresentado neste trabalho possui uma estrutura genérica. Como foi ilustrado neste trabalho, este modelo pode ser ajustado para representar diferentes contextos de interfaces a partir da modificação do conjunto de cores de modo a representar os objetos de interação e as ações de navegação.

A maior limitação do modelo de navegação é o grande número de fichas necessário para descrever a interface, o que torna difícil à análise do comportamento do sistema. Mas com o uso de MSCs essa limitação pode ser superada.

Como direções futuras para este trabalho, os autores pretendem estender o modelo para incorporar a funcionalidade de outros objetos de interação e em particular incluir novas situações de interação considerando estados de interrupção e estados de espera.

Os autores pretendem também aplicar este modelo a casos mais complexos (com maior número de janelas e de objetos de interação) e introduzir considerações temporais de modo a analisar aspectos de desempenho do usuário durante a execução de tarefas.

## Agradecimentos

Os autores gostariam de agradecer a CAPES (Coordenação de Aperfeiçoamento de Pessoal do Ensino Superior) por apoiar esta pesquisa com uma bolsa de estudo para um dos autores.

## Referências Bibliográficas

- [1] Bastide, R.; Palanque, P., *A Petri Net Based Environment for the Design of Event-Driven Interfaces*. Proceedings of the 16<sup>th</sup> International Conference on Application and Theory of Petri Nets, p. 66-83, Turin, June, 1995.
- [2] Jensen. K., *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, Volume 1, Basic Concepts*. Monographs in Theoretical Computer Science, Springer-Verlag, 1992.
- [3] Murata. T., *Petri Nets: Properties, Analysis and Applications*. Proceedings of the IEEE, Vol. 77, No 4, April 1989.
- [4] Palanque P.; Bastide R.; V. Senges, *Validating interactive system design through the verification of formal task and system models*. 6th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'95) Grand Targhee Resort, Wyoming, U.S.A., p. 14-18 August 1995.
- [5] Rosis, F. de. *Tool to Support Specification and Evaluation of Context-Customizes Interfaces*. SIGCHI Bulletin, 82-88, July 1996.
- [6] Andersson, M. and Bergstrand, J., *Formalizing Use Cases with Message Sequence Charts*, Department of Communication Systems at Lund Institute of Technology, May 1995.
- [7] Sousa, M.R.F. and Turnell, M.F.Q.V., *User interface evaluation based on Coloured Petri nets modelling and analysis*. In IEEE - SMC'98 Proceedings, October 1998, SanDiego - CA, EUA.
- [8] Sousa, M.R.F. *Avaliação Iterativa da Especificação de Interfaces com Ênfase na Navegação*, Doctoral Thesis, Electrical Engineering Department, Federal University of Paraíba, Brazil, December 1999.
- [9] Turnell, M.F.Q.V.; Farias, G.F. *The use of Supervisory Software in the Industrial Automation Process Control from the User Interface Perspective*. 1996 IEEE International Conference on Systems, Man and Cybernetics; Beijing China, October, 1996.
- [10] University of Aarhus, *Design/CPN Occurrence Graph Manual, Version 3.0*, Computer Science Department, 1996.