

IMPLEMENTAÇÃO EM SISTEMA SCADA DE CONTROLE SUPERVISÓRIO BASEADO EM MODELOS EM REDES DE PETRI DE ALTO NÍVEL

Lucas Nasciutti Prudente, lukaznp@yahoo.com.br¹
Horácio Medina da Rocha, horaciomedina@gmail.com¹
Eduardo José Lima II, eduardo@demec.ufmg.br²

¹Engenharia de Controle e Automação, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 - Pampulha, CEP 31270-901, Belo Horizonte, Minas Gerais, Brasil

²Departamento de Engenharia Mecânica, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 - Pampulha, CEP 31270-901, Belo Horizonte, Minas Gerais, Brasil

Resumo: Em sistemas de manufatura integrada por computador, a implementação de sistemas supervisórios SCADA (Supervisory Control And Data Acquisition) e a programação de Controladores Lógico-Programáveis (CLPs) são realizadas com base na experiência do programador e em seu conhecimento das etapas da manufatura. Essa implementação normalmente não é realizada seguindo uma metodologia formal, o que facilita a ocorrência de erros, bem como ineficiência e ociosidade de recursos. Adiciona-se a isso o fato de sistemas flexíveis de manufatura lidarem com um grande número de variáveis, como tipos de matérias primas, tarefas a serem executadas, tipos de produtos acabados etc. Uma técnica para se modelar sistemas flexíveis de manufatura é a rede de Petri. Para se modelar um sistema de manufatura de alta complexidade utilizam-se redes de Petri de alto-nível, por permitirem uma melhor utilização de todos os recursos disponíveis, bem como melhor organização do processo produtivo e visualização das informações sobre as variáveis do processo. Neste trabalho é apresentada a implementação e a aplicação de técnicas de controle supervisório utilizando redes de Petri de alto-nível em um software SCADA para uma planta conhecida, com o objetivo de gerar regras de controle e verificar sua aplicabilidade a um sistema real. O uso da técnica de controle supervisório em um software SCADA foi dividido em duas partes: controle de alto nível e controle de baixo nível. O primeiro, executado no supervisório, implementa o jogador de redes de Petri colorida e o segundo, executado no CLP, comanda os sinais da planta baseado em comandos enviados pelo supervisório. O modelo foi implementado no supervisório por meio de scripts respeitando todas as restrições e prioridades de funcionamento da planta e considerando as funcionalidades da programação orientada a objetos.

Palavras-chave: redes de Petri coloridas, controle supervisório, SCADA

1. INTRODUÇÃO

Em sistemas de manufatura integrada por computador, a implementação de sistemas supervisórios SCADA (Supervisory Control And Data Acquisition) e a programação de Controladores Lógico Programáveis (CLPs) são realizadas com base na experiência do programador e em seu conhecimento das etapas da manufatura (Lima II, 2007). Essa implementação normalmente não é realizada seguindo uma metodologia formal, o que facilita a ocorrência de erros, bem como ineficiência e ociosidade de recursos. Adiciona-se a isso o fato de sistemas flexíveis de manufatura lidarem com um grande número de variáveis, como tipos de matérias primas, tarefas a serem executadas, tipos de produtos acabados etc.

Sistemas de manufatura se caracterizam pela ocorrência de eventos discretos no tempo, como início e conclusão de determinada tarefa. Sistemas a eventos discretos (SEDs) são sistemas em que o espaço de estados é discreto e que sua evolução é dirigida pela ocorrência de eventos discretos, com duração instantânea no tempo (Ribeiro et al., 2007). Uma das técnicas utilizadas para modelar SEDs são as redes de Petri.

Lima II (2007) mostra a implementação em um único CLP do controle supervisório baseado em modelos em redes de Petri ordinárias. A metodologia modular utilizada permite a implementação de modelos relativamente complexos sem aumentar a complexidade do programa, uma vez que o controle supervisório divide o sistema em um modelo de alto nível (supervisão) e em um modelo de baixo nível. O modelo de alto nível se comunica com o modelo de baixo nível para gerar e observar sinais de início e final de tarefas e assim permite uma análise mais fácil e um modelo mais compacto.

A rede de Petri de alto nível permite modelar SEDs com alta complexidade. A utilização de metodologias formais como as redes de Petri de alto nível permite uma melhor utilização dos recursos disponíveis, bem como melhor organização do processo produtivo e visualização das informações e das variáveis do processo. Além disso, futuras alterações no processo seriam mais facilmente implementadas através de alterações do modelo. Dessa forma, o uso desta metodologia formal para a modelagem e a geração das regras para o sistema de controle supervisório possibilita a diminuição de erros durante a implementação, a redução da ociosidade de recursos, assim como a utilização de técnicas de análise do funcionamento do sistema, como verificação de *dead-locks* etc.

Considerando a modelagem de sistemas de manufatura utilizando redes de Petri de alto nível, foi desenvolvida uma metodologia para a implementação de regras para o controle supervisório de plantas flexíveis de manufatura integrada por computador utilizando modelos em redes de Petri coloridas. Foram considerados aspectos como: integração entre os diferentes equipamentos e diferentes tarefas executadas por cada equipamento, redução de tempo ocioso dos processos de fabricação, representação de receitas (seqüências de tarefas a serem executadas em cada peça), concorrência entre processos e paralelismo.

Considerando uma planta flexível de manufatura analisada e modelada anteriormente (Oliveira e Lima II, 2008), a metodologia desenvolvida será utilizada para a geração de regras de controle e verificação de sua aplicabilidade a um sistema real.

Neste trabalho é apresentada a implementação em sistema SCADA de controle supervisório baseado em modelos em redes de Petri coloridas. Para validação e testes, foi utilizado um sistema SCADA, executando, em computadores diferentes, o controlador propriamente dito e uma simulação da planta. Os dois computadores foram interligados por um CLP que realizou o envio de comandos para a planta e retornou ao controlador os sinais advindos da planta, de modo que, para o controlador, é indiferente se a planta controlada é real ou simulada.

2. REDES DE PETRI

Redes de Petri utilizam uma metodologia formal para modelar sistemas a eventos discretos (SEDs). Uma rede de Petri é um grafo bipartido cuja estrutura é composta de *lugares* e *transições* (Murata, 1989; Villani, 2004). Os lugares são representados por círculos e as transições por barras. Os lugares e as transições são conectados por arcos orientados que representam a relação do fluxo do lugar para a transição e da transição para o lugar. Uma rede de Petri marcada contém um número inteiro de marcas (ou fichas) que são distribuídas entre os lugares. Esta repartição descreve o estado discreto do modelo.

Quando uma transição é disparada, um número de fichas é retirado de determinado(s) lugar(es) e adicionado a outro(s). Um arco com peso D^+_{ij} da transição j para o lugar i indica que quando a transição j dispara o lugar i recebe D^+_{ij} fichas. Um arco com peso D^-_{ij} da transição j para o lugar i indica que quando a transição j dispara o lugar i perde D^-_{ij} fichas. Uma transição está habilitada se todos os seus lugares de entrada contêm um número mínimo de fichas. Uma transição que se encontra nesta condição está apta a disparar. Quando uma transição dispara todos os seus lugares de entrada perdem um número de fichas e todos os seus lugares de saída ganham um número de fichas (Moody and Antsaklis, 1998).

Definindo-se a matriz de incidência D , o vetor de disparo de transições s , e o vetor de marcação da rede M , a equação dinâmica do sistema é dada por:

$$M_{k+1} = M_k + Ds \quad (1)$$

Ao se modelar sistemas de manufatura, as transições estão relacionadas a eventos do tipo início de operação, final de operação, início de movimento, final de movimento etc. Os lugares estão relacionados a estados parciais do sistema, como peça pronta para ser processada, peça processada e peça pronta para ser transferida.

2.1. Redes de Petri Coloridas

Redes de Petri coloridas permitem modelar Sistemas a Eventos Discretos (SEDs) de alta complexidade. Essa técnica permite se ter um modelo muito mais simples e compacto do SED que se fossem utilizadas redes de Petri ordinárias. Além disso, permite a manipulação de dados associados às fichas.

Para se diferenciar as fichas, diferentes cores (números inteiros ou conjunto de nomes) são associadas a cada uma delas. Para cada lugar é associado um conjunto de cores de fichas que podem pertencer a este lugar. Para cada transição é associado um conjunto de cores que corresponde às diferentes maneiras de se disparar a transição. Em casos simples, quando todos os processos têm a mesma estrutura e são independentes uns dos outros, as cores das transições estão diretamente associadas aos processos e o conjunto de cores dos lugares e transições são idênticos (Cardoso and Valette, 1997).

3. IMPLEMENTAÇÃO DE CONTROLE SUPERVISÓRIO UTILIZANDO REDES DE PETRI ORDINÁRIAS

O controle supervisório visa controlar o funcionamento do sistema de modo a assegurar que certas restrições de coordenação sejam respeitadas entre seus diversos componentes. Este controle age apenas no sentido de inibir a

ocorrência de eventos que levem à violação das restrições. Assim, essa abordagem considera que os subsistemas funcionam independentemente da existência do controlador. Com isso, apenas o início e final de operação de cada subsistema são considerados pelo modelo do supervisor.

Torna-se importante então o conceito de transição controlável e transição não controlável (Moody and Antsaklis, 1998). Transições controláveis são aquelas que podem ser inibidas pelo controlador, e representam, portanto, o início de cada operação. Já o fim das operações são representadas por transições não controláveis, podendo apenas serem observados pelo supervisor.

Em alguns casos é necessário que o CLP controle também as seqüências de operações dos subsistemas. Em Lima II e Dórea (2002), foi apresentada uma metodologia para implementação de controladores baseada em Queiroz et al. (2001). Nessa metodologia, o programa em CLP é organizado em três partes: o jogador da rede de Petri (JOG), que atualiza a marcação da rede de acordo com o disparo das transições; o executor (EXE), que determina se há alguma transição controlável habilitada e comanda seu disparo e as seqüências operacionais (SEQOP), que implementam os comandos dos sinais para controlar cada um dos subsistemas da planta. Essa última pode ser implementada da forma que o projetista preferir, mas devido à sua simplicidade, sugere-se a abordagem baseada em máquinas de estados (Fabian e Hellgren, 1998).

A Fig. (1) mostra como o programa é organizado. O vetor M de marcação da rede é determinado pela rotina JOG a partir de um vetor S de disparo de transições. O disparo de transições controláveis é comandado pela rotina EXE a partir do vetor SC e a rotina SEQOP implementa o controlador de baixo nível iniciando as operações determinadas por SC e indicando o disparo de transições por S (Lima II, 2007).

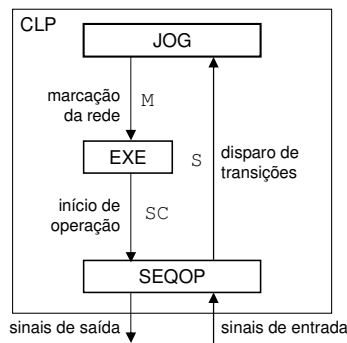


Figura 1. Estrutura do controlador.
Fonte: Lima II, 2007

As rotinas JOG e EXE independem do modo de funcionamento físico da planta, implementando somente o controle de alto nível (ou supervisão). Já a rotina SEQOP, por sua vez, independe do modelo do supervisor, implementando apenas o controle de baixo nível. Essa modularidade do programa permite, por exemplo, que programadores diferentes (que dominem técnicas de modelagem ou que dominem o controle das seqüências das operações) trabalhem em conjunto em um mesmo controlador.

Essa metodologia, contudo, não permite que sistemas de elevada complexidade sejam programados, pois o modelo se tornaria extenso e de grande peso computacional para execução pelo CLP. Na seção 5 será apresentada uma metodologia para implementação em sistema SCADA de sistemas flexíveis de manufatura utilizando redes de Petri coloridas.

4. DESCRIÇÃO DO PROCESSO

A planta utilizada para neste trabalho foi a Planta CIM do SENAI-CIMATEC localizada em Salvador - BA. A Planta CIM é uma planta que, mesmo com o propósito didático, possui equipamentos industriais. Dentre estes equipamentos estão: manipuladores robóticos, centros de usinagem com torno e fresa, esteira transportadora, atuadores pneumáticos, estação de armazenamento de peças, estação de processos, estação de montagem, entre outros.

A planta funciona da seguinte maneira: uma esteira transportadora contém vagões que levam pallets de uma estação a outra, sendo que cada pallet pode estar com algum produto ou subproduto de alguma estação ou não conter nada. Quando uma estação está disponível, um sinal é enviado informando o status desta estação para o CLP, então o CLP retorna um sinal informando-a para entrar em estado de espera. Quando um pallet com uma peça necessária para o funcionamento da estação estiver próximo, o CLP informa à estação que ela pode retirar a peça da esteira. A estação então inicia a sua operação e informa ao CLP quando terminá-la. Quando um pallet vazio chegar em frente a esta estação, o CLP a informa que a peça pode ser colocada novamente na esteira. Após isto, a estação informa ao CLP que está novamente disponível.

A Fig. (2) mostra a arquitetura física da Planta CIM.

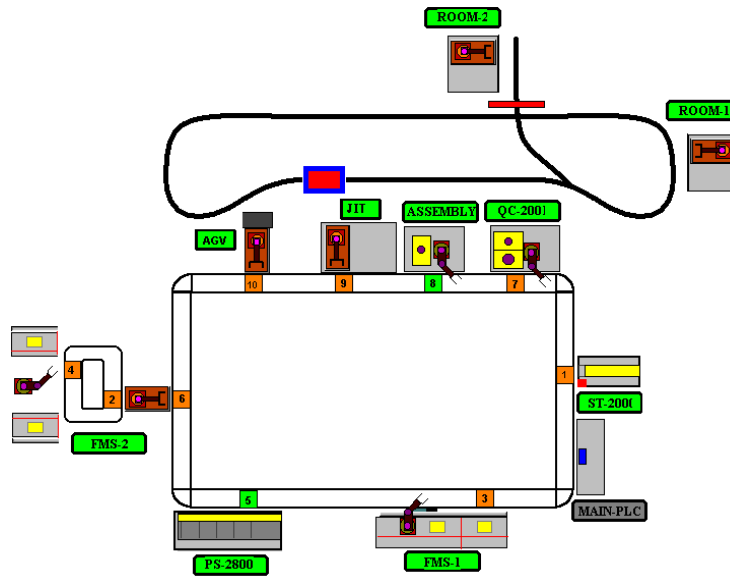


Figura 2. Arquitetura da Planta CIM

A planta possui várias estações, cada uma com a sua determinada funcionalidade. As estações FMS-1 e FMS-2 são estações compostas de torno e fresa, que recebem peças para serem usinadas. A estação ASSY-2001 é responsável pela montagem da peça final. Esta estação pega todas as peças que compõem o produto e faz a sua montagem. Já a estação QC-2001, realiza o controle de qualidade das peças usinadas na FMS-1 e FMS-2. Outra estação é a JIT, para onde as peças são enviadas para trabalhos manuais. A estação AGV é responsável pelo transporte das peças para estações que não pertencem às esteiras ou para outros locais distantes. A estação ST-2000 é responsável pelo armazenamento das peças já trabalhadas e em colocar na esteira peças para serem processadas. Por fim, a estação PS-2008 é o local onde são feitas as pinturas nas peças.

Neste trabalho foram abordadas as estações FMS-1, QC-2001, PS-2008, ASSY-2001 e ST-2000, já que essas são fundamentais para o funcionamento da planta (produção de uma peça completa). As demais estações aumentariam o tamanho do modelo e do controlador, sem, contudo, aumentar a complexidade do sistema, ou seja, as 5 estações trabalhadas representam bem todas as estratégias de controle da planta.

5. METODOLOGIA DE CONTROLE

Para realizar o controle da planta descrita na seção anterior, serão utilizados modelos em redes de Petri Coloridas de cada estação da planta adaptados dos modelos de Oliveira e Lima II (2008). Os programas de controle baseados nos modelos irão rodar em um PC contendo o software supervisor da planta. Tais modelos contém todas as informações necessárias para o funcionamento da planta, respeitando, assim, todas as restrições e prioridades de funcionamento.

No modelo apresentado abaixo os círculos representam os lugares da rede, podendo modelar tanto lugares físicos como estados. As barras representam as transições da rede, estas transições representam término (transição não controlável) ou início (transição controlável) de algum processo.

No modelo de Oliveira e Lima II (2008) os arcos são etiquetados descrevendo as propriedades das fichas retiradas e as propriedades das fichas depositadas em cada lugar quando do disparo das transições.

As fichas do modelo possuem várias cores que também podem ser chamadas de atributos. Estes atributos irão identificar o tipo da peça, o número do vagão etc. Para o modelo da planta estudada foram definidos doze atributos para as fichas.

O primeiro atributo foi nomeado *vagao*, e marca a presença ou não de um vagão na ficha. O segundo atributo se chama *pallet* e indica se há presença do pallet. Para o terceiro atributo utilizou-se o nome *peca*, o qual informa se a ficha possui uma peça. Esses três atributos representam, portanto, todas as combinações possíveis entre vagão, pallet e peça. Por exemplo, um vagão pode estar vazio (*vagao*=verdadeiro, *pallet*=falso, *peca*= falso), ou um pallet pode estar em um buffer com uma peça (*vagao*=falso, *pallet*=verdadeiro, *peca*=verdadeiro).

O quarto atributo chama-se *numvagao* e serve para indicar o número do vagão na esteira. O quinto atributo recebe o nome de *tipopeca*, e mostra se a peça é do tipo 1 (cilindro-1), 2 (cilindro-2), 3 (chapa-1) ou 4 (chapa-2). O sexto atributo se chama *usinada*, e informa se a peça já foi usinada ou não. Já o sétimo atributo recebe o nome de *posicao* e indica qual lado da peça já foi usinado. Para o oitavo atributo foi utilizado o nome *cor* que serve para indicar a cor da peça sendo 1 azul e 2 verde. O nono atributo recebeu o nome de *colorida* e serve para indicar se a peça já foi pintada.

Para o décimo atributo foi utilizado o nome de *testada* que informa se a peça já foi testada. O atributo número onze recebe o nome de *aprovada* e indica se a peça testada foi aprovada. O último atributo chama-se *montada* e representa se a peça já foi montada.

Na seção a seguir é apresentado o modelo da estação FMS.

5.1. Modelo da Estação FMS

A Fig. (3) representa o modelo em rede de Petri Colorida da estação FMS. Para entrar na estação, os vagões devem conter peças brutas (atributo *usinada*=falso), que não tenham sido pintadas (*colorida*=falso) nem montadas (*montada*=falso). As peças devem ter sido também testadas (*testada*=verdadeiro) e aprovadas (*aprovada*=verdadeiro). Esses valores são definidos pelos pesos dos arcos. Na Tab. (1) são apresentadas as definições dos pesos dos arcos das transições *iPRG1* (início de execução do programa 1 do robô) e *fPRG1* (final de execução do programa 1 do robô). Nesta tabela, pode-se observar o valor de todos os atributos da ficha de cada lugar de entrada para disparar a transição. Os atributos em negrito são copiados para a ficha de saída da transição e os atributos marcados com um traço são os irrelevantes para o disparo da transição. O lugar com apenas um atributo representa uma ficha ordinária. Na transição *iPRG1* os três primeiros lugares representam as fichas de entrada da transição e o quarto lugar representa a ficha de saída da transição. Na transição *fPRG1* o primeiro lugar representa a ficha de entrada da transição e os três lugares seguintes representam as fichas de saída da transição.

Os lugares CT, CF e CR (Fig. (4)) representam os recursos da estação. CR evita que o robô seja requisitado pelo supervisor para executar mais de uma tarefa simultaneamente. Já os lugares CT e CF evitam que a estação aceite mais de uma peça para ser usinada pelo torno ou pela fresa simultaneamente. Inicialmente, esses lugares possuem ficha, já que os recursos se encontram disponíveis.

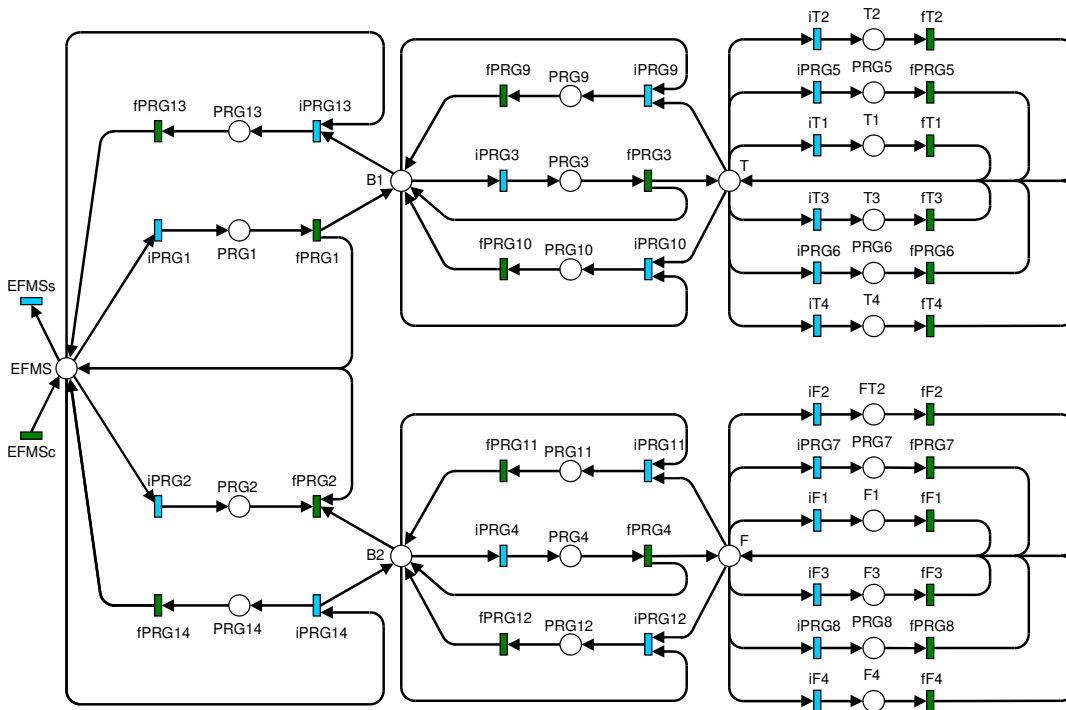


Figura 3. Modelo em rede de Petri Colorida para a estação FMS

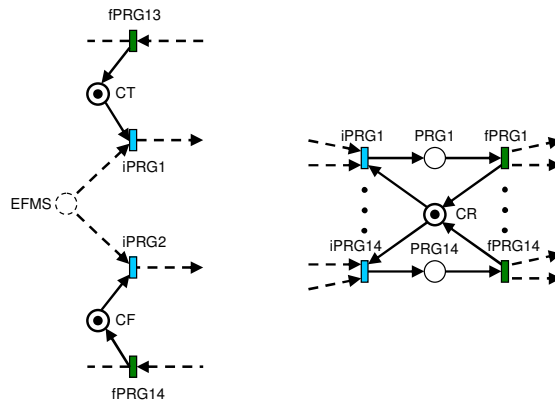


Figura 4. Modelagem dos recursos do torno e da fresa e recurso do robô

Tabela 1. Trecho da tabela de arcos FMS

Transição	Lugar	<vagão, pallet, peça, numvagão, tipopeça, usinada, posição, cor, colorida, testada, aprovada, montada>
iPRG1	EFMS	< V, V, V, n, t, F, -, c, F, V, V, F >
	CT	1
	CR	1
fPRG1	PRG1	< V, V, V, n, t, F, -, c, F, V, V, F >
	PRG1	< V, V, V, n, t, F, -, c, F, V, V, F >
	CR	1
	EFMS	< V, F, F, n, -, -, -, -, -, -, -, - >
	B1	< F, V, V, -, t, F, -, c, F, V, V, F >

Para exemplificar o funcionamento do modelo, considera-se a chegada de um vagão contendo um cilindro bruto com fabricação definida para cilindro do tipo 1 (lugar EFMS). Inicialmente, o pallet com o cilindro é transportado pelo robô (lugar PRG1) para o buffer (lugar B1). Então, o robô pega o cilindro no pallet (lugar PRG3) e o coloca no torno (lugar T). O torno usina o primeiro lado da peça (lugar T1). Depois, o robô vira a peça (lugar PRG5) para que seja usinado o outro lado desta (lugar T2). A peça é em seguida levada pelo robô (lugar PRG9) para o buffer (lugar B1) contendo o pallet. Por fim, o buffer contendo o cilindro usinado é colocado pelo robô no vagão (lugar EFMS). Após a fabricação a ficha passa a ter o atributo *usinada* igual a verdadeiro e os atributos *colorida*, *testada*, *aprovada* e *montada* iguais a falso.

6. ARQUITETURA DO CONTROLADOR

Com base nos modelos das estações, o controle da planta estudada foi feito através da técnica de controle supervísório implementado em um software SCADA. Esta técnica, baseada no método apresentado na seção 3, divide o controle do processo em 3 rotinas: JOG, EXE e SEQOP. As rotinas JOG e EXE rodam em um PC contendo o software SCADA e fazem o controle de alto nível. O modelo de cada estação será implementado nesta parte, e estará rodando em forma de script no software Elipse E3. A rotina SEQOP estará rodando em um CLP. Esta rotina faz o controle de baixo nível da planta, recebendo ordens para execução de tarefas (transições controláveis) e informando ao supervísório o término das mesas (transições não controláveis). A Fig. (5) mostra um esquema de como estas rotinas estão divididas no processo.

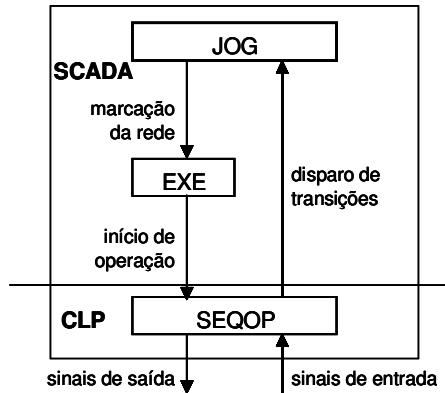


Figura 5. Estrutura das rotinas do supervisor

Um exemplo de seqüência de sinais de comunicação entre os sistemas é mostrado na Fig. (6). Neste exemplo, o supervísório envia para o CLP um comando ordenando que o robô deve iniciar o programa 4 (transição controlável iPRG4). Então o CLP envia para o robô uma seqüência de sinais para que esse execute o referido programa. Assim, o robô começa a executar o programa e envia para o CLP um sinal informando que está executando-o (EXEC). Quando o robô termina de executar esse, o programa pára de enviar para o CLP o sinal EXEC indicando que terminou a execução. Com isso, o CLP percebe que o robô terminou de executar o programa e envia para o supervísório um sinal informando que o robô terminou de executar a tarefa (transição não controlável fPRG4). Enfim, o supervísório realiza o disparo da transição não controlável, atualizando a marcação da rede de Petri.

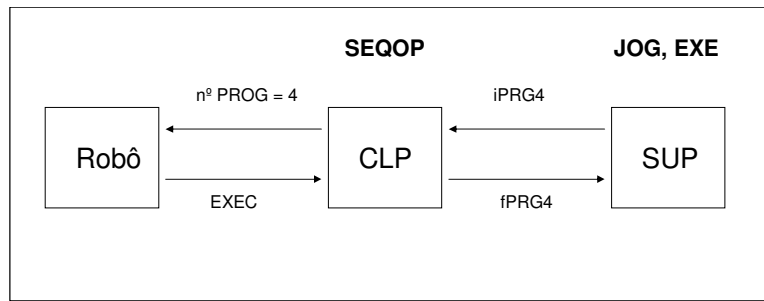


Figura 6. Exemplo de funcionamento do controle supervísório usando sistema SCADA

7. IMPLEMENTAÇÃO DO CONTROLADOR

Para se implementar o controlador foi utilizado o software supervísório Elipse E3. Primeiramente, foi configurada a comunicação do supervísório com o CLP por meio de drivers específicos para o modelo do controlador SIEMENS. Cada estação foi representada no supervísório por uma tela, sendo que cada tela apresenta a imagem do modelo da respectiva estação. Para a criação das fichas coloridas da rede, utilizou-se o objeto objectX do software e por fim foram feitos os scripts contendo as instruções de controle da planta.

7.1. Telas

Para se representar cada estação no supervísório foram utilizadas as imagens dos modelos. Sobre essas imagens, foram criados círculos no Elipse E3 em cada lugar da rede. Estes círculos representam a existência ou não de fichas em cada lugar da rede. Cada círculo tem uma propriedade chamada *Visible* permitindo que o círculo esteja visível ou não. Assim, quando houver uma ficha em um determinado lugar, o círculo deste lugar estará visível representando a ficha do modelo gráfico. A Fig. (7) mostra em detalhe o lugar PRG1 com uma ficha e os lugares PRG13 e B1 sem fichas.

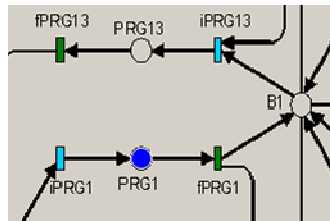


Figura 7. Representação da marcação da rede

7.2. Criação do ObjectX

Para representar as fichas da rede de petri foi utilizado o recurso ObjectX do Elipse E3. Este recurso permite que se criem classes com vários atributos que, nesse caso, representam as propriedades que compõem as cores das fichas. Para cada lugar das redes de Petri foi instanciado um objeto da classe “FichaColorida”, que possui 12 atributos representando as propriedades da ficha. Um 13º atributo foi criado para identificar se a ficha existe ou não no respectivo lugar da rede. Também foi criada a classe “Ficha”, que possui apenas 1 atributo e assim cada objeto desta classe representa uma ficha ordinária na rede de Petri (lugares relacionados à disponibilidade de recursos).

Foi criada, ainda, uma associação no Elipse E3 entre a propriedade *Visible* de cada círculo das telas com a propriedade *Existe* do respectivo lugar. No momento em que o script de atualização da marcação da rede atualiza o valor da propriedade *Existe*, automaticamente o programa atualiza a tela tornando visível ou não o círculo que representa a ficha.

7.3. Scripts

As instruções contendo as regras de controle foram feitas usando scripts no Elipse E3. Para o controle de cada estação foram utilizados três scripts, o primeiro rodando a cada um segundo no supervísório, o segundo e o terceiro scripts rodando a cada vez que for alterado o valor das tags de comunicação com o CLP.

O primeiro script recebeu o nome de “1seg”. Este *script* faz a verificação da condição para os disparos das transições, sendo executado a cada um segundo na tag de transição controlável da estação. Quando uma condição for satisfeita, uma mensagem para o CLP é enviada, informando qual tarefa deve ser iniciada. Para que não houvesse mais

de uma transição disparada ao mesmo tempo em uma mesma estação, foi criada uma tag interna que recebe uma string informado qual transição cujo disparo foi comandado. Caso não houvesse transição disparada, essa string estaria vazia e isso possibilitaria o disparo de uma outra transição. A tag interna também foi criada para se evitar conflito nos disparos das transições e, mesmo se houver conflito, a ordem em que as transições aparecem no script determina a sua prioridade de disparo.

O segundo script do programa se chama "OnChangeTC" e faz a atualização da marcação da rede de Petri após efetivado pelo CLP o disparo de uma transição controlável. Ele é executado na tag de comunicação de transições controláveis da estação toda vez que o valor da tag for alterado. O CLP, para indicar que recebeu o comando do supervisor e iniciou a tarefa, atualiza o valor da tag de transição controlável para 0. Ao se satisfazer esta condição, o programa atualiza a marcação da rede e depois os atributos das fichas. No final de cada condição a tag interna é zerada para possibilitar o início de outra transição pelo script 1seg.

O terceiro script roda na tag de transição não controlável da estação e faz a atualização da marcação da rede no final de cada tarefa. Este script recebe o nome de "OnChangeTNC", sendo executado toda vez que o valor da tag for alterado. A condição é satisfeita quando um bit da tag for verdadeiro. Assim, o programa realiza a atualização da marcação da rede e dos atributos da ficha. Por fim, o programa faz o bit da tag igual a falso informando ao CLP que já terminou de atualizar a marcação.

8. RESULTADO

Os resultados dos testes do programa desenvolvido neste projeto a partir da comunicação do supervisor com o CLP são apresentados nas Figs. (8) e (9). Estas figuras representam a tela do supervisor com a estação FMS. As seqüências de disparos mostradas nas figuras são para a chegada de um vagão com um cilindro do tipo 1 na estação.

Na Fig. (8) o vagão com a peça está na entrada da estação. Na Fig. (9) o robô inicia o processo de pegar o pallet com a peça e colocá-lo no buffer. Nesta figura pode-se observar que o recurso do torno (CT) não está mais disponível, pois já começou o processo de usinagem da peça e o recurso do robô (CR) também não está disponível porque o robô esta executando uma operação.

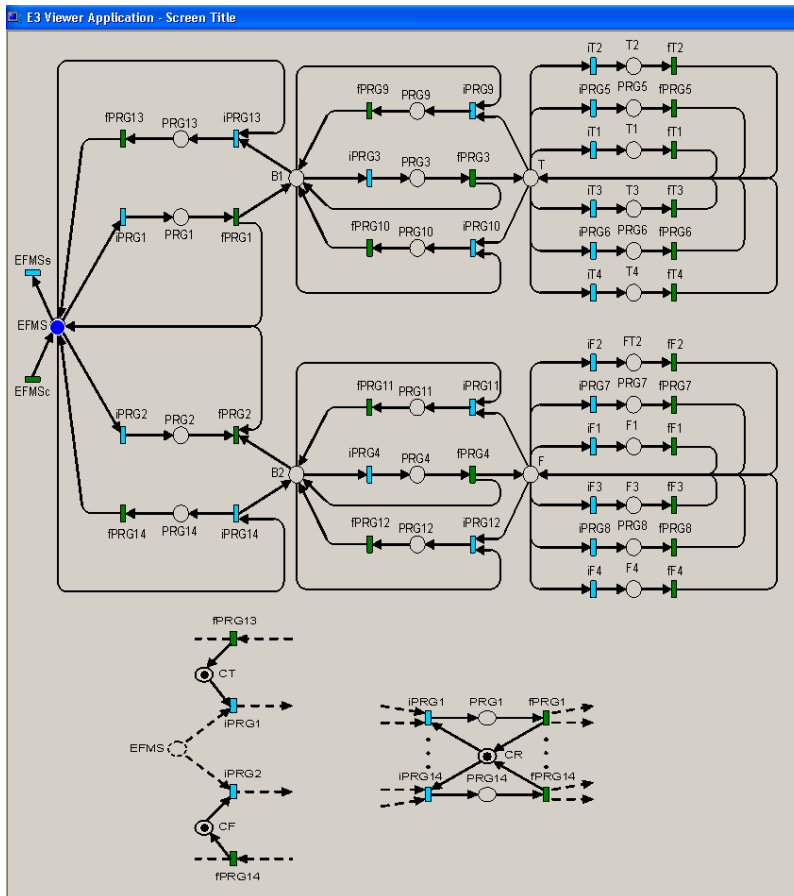


Figura 8. Marcação da rede após chegada de uma peça bruta.

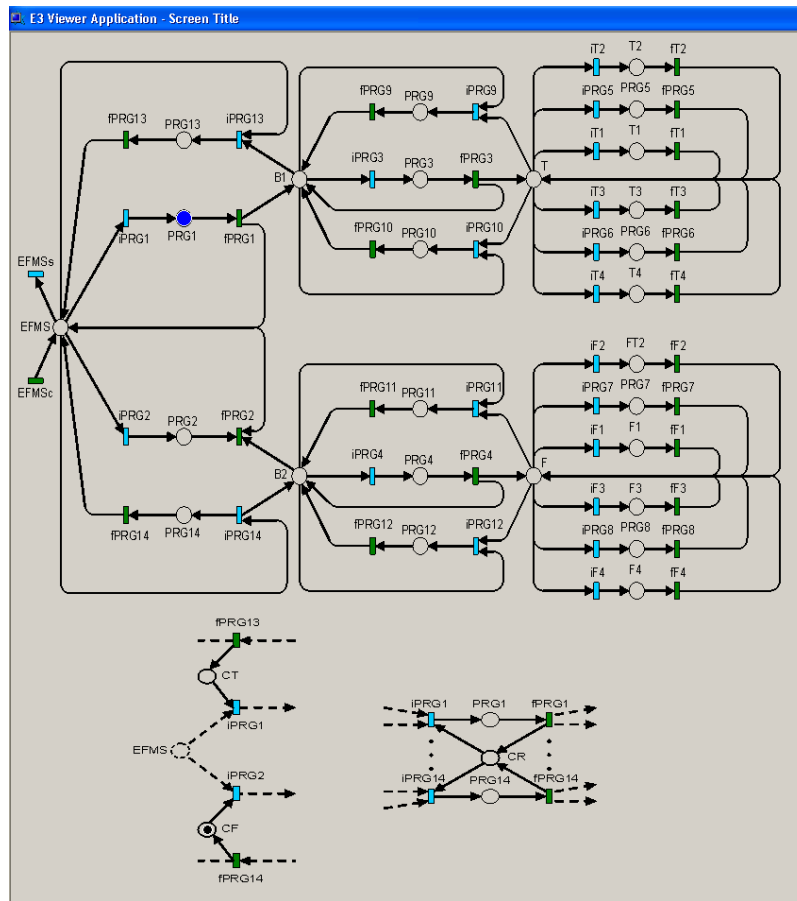


Figura 9. Marcação da rede após início do programa PRG1.

O teste do sistema de controle foi realizado utilizando um CLP com a programação dos sinais de baixo nível da planta e uma simulação da planta sendo executada em outro PC.

9. CONCLUSÕES

Neste projeto foram estudados os modelos do controlador do sistema de manufatura flexível da planta CIM do SENAI-CIMATEC. Com base nestes modelos, foi apresentada uma metodologia para implementação em um sistema SCADA da técnica de controle supervísório utilizando redes de Petri coloridas.

O uso de redes de Petri coloridas diminuiu consideravelmente a complexidade dos modelos das estações. O emprego de redes de Petri ordinárias teria tornado os modelos das estações muito maiores e, com isso, sua implementação ficaria bem mais complexa. Assim, foi possível verificar a grande vantagem de se usar a rede colorida, uma vez que nela as fichas possuem várias cores (atributos) e, portanto, possibilitam a obtenção de um modelo muito mais compacto, diminuindo a complexidade de sua implementação no supervísório.

O supervísório implementado funcionou perfeitamente em conjunto com o CLP, enviando comandos e aguardando término de operações.

10. AGRADECIMENTOS

Os autores agradecem à FAPEMIG pelo apoio financeiro ao desenvolvimento do trabalho e ao SENAI-CIMATEC por disponibilizar as informações sobre o sistema estudado.

11. REFERÊNCIAS

- Cardoso, J. E Valette, R., 1997. “Redes de Petri”. Ed. da UFSC, Florianópolis.
 Fabian, M. e Hellgren, A., “PLC-based Implementation of Supervisory Control for Discrete-Event Systems”, Proceedings of the 37th IEEE Conference on Decision and Control, Tampa, EUA, 1998, pp. 3305–3310;

- Gaidzinski, V. H., 2003. "A Tecnologia da Informação no Chão de Fábrica: As Novas Ferramentas e a Gestão Integrada da Informação". Florianópolis, 153p. Dissertação (Mestrado) – Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção.
- Lima II, E.J., 2007. "Implementação em Linguagem Ladder de Controladores Baseados em Modelos em Redes de Petri". Anais do 2º Seminário de Sistemas Industriais e Automação, Belo Horizonte - MG.
- Lima II, E. J.; Dórea, C. E. T., 2007. "Implementação Através de CLPs de Controle Supervisório Baseado em Redes de Petri para uma Célula de Manufatura". Anais do XIV Congresso Brasileiro de Automática, 2002, Natal-RN.
- Moody, J.O. AND Antsaklis, P.J., 1998. "Supervisory Control of Discrete Event Systems Using Petri Nets", Kluwer Academic Publ., Boston.
- Murata, T., 1989. "Petri Nets: Properties, analysis and applications". Proceedings of the IEEE, 77(4), pp. 541–580.
- Oliveira, G.L.S., e Lima II, 2008. "Modelagem de Controlador para um Sistema Flexível de Manufatura Utilizando Redes de Petri Coloridas", XVII Congresso Brasileiro de Automática, Juiz de Fora-MG.
- Queiroz, M.H., Santos, E.A.P. e Cury, J.E.R., 2001. "Síntese Modular do Controle Supervisório em Diagrama Escada para uma Célula de Manufatura", Anais do V Simpósio Brasileiro de Automação Inteligente, Canela, RS.

12. DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído neste trabalho.

IMPLEMENTATION OF SUPERVISORY CONTROL IN SCADA SYSTEMS BASED ON HIGH LEVEL PETRI NETS

Lucas Nasciutti Prudente, lukaznp@yahoo.com.br¹
Horácio Medina da Rocha, horaciomedina@gmail.com¹
Eduardo José Lima II, eduardo@demec.ufmg.br²

¹Engenharia de Controle e Automação, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 - Pampulha, CEP 31270-901, Belo Horizonte, Minas Gerais, Brasil

²Departamento de Engenharia Mecânica, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 - Pampulha, CEP 31270-901, Belo Horizonte, Minas Gerais, Brasil

Abstract: *In computer integrated manufacturing systems, the implementation of SCADA (Supervisory Control And Data Acquisition) systems and the programming of PLC (programmable logic computers) are made based on the experience of the developer and his knowledge of the manufacturing process. This implementation isn't normally made by following a formal methodology and this causes the occurrence of errors and inefficiency and idleness of resources. Add to this the fact that flexible manufacturing systems deal with a large number of variables, such as raw materials types, tasks to be performed, types of finished products and so on. One efficient technique for modeling flexible manufacturing systems is the Petri nets. The high level Petri nets are used for modeling a manufacturing system of high complexity because they allow a better representation of all available resources as well as a better organization of the production process and a good display of the process variables informations. In this work it's presented the implementation and application of supervisory control techniques using high level Petri nets in a SCADA software for a given plant, with the objective to generate control rules and to verify its applicability in a real system. The implementation is divided in two parts: high level control and low level control. The first part runs in the supervisory software and implements the token player of the coloured Petri Net. The second part runs in the PLC and commands the plant signals based on commands sent by the supervisor. The model has been implemented in the SCADA software through scripts respecting all the constraints and operation priorities of the plant and applying object-oriented programming functionalities.*

Keywords: *Coloured Petri nets, supervisory control, SCADA*