VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA
VI NATIONAL CONGRESS OF MECHANICAL ENGINEERING
18 a 21 de agosto de 2010 – Campina Grande – Paraíba - Brasil
August 18 – 21, 2010 – Campina Grande – Paraíba – Brazil

MODELAGEM E ESPECIFICAÇÃO DE CONTROLE DE SISTEMAS FLEXÍVEIS DE MANUFATURA UTILIZANDO REDES DE PETRI DE ALTO NÍVEL

Alexandre da Silva Ribeiro, Alexandre.ribeiro@fieb.org.br[1]
Herman Lepikson, herman@ufba,br[2]
Eduardo Jose Lima II, eduardo@demec.ufmg.br[3]

[1] SENAI CIMATEC – Centro Integrado de Manufatuca e Tecnologia.
[2]Universidade Federal da Bahia.
[3]Universidade Federal de Minas Gerais .

Resumo: *Baseando-se nos conceitos de Redes de Petri a objeto as especificações de controle para os Sistemas Flexível deManufatura (FMS) podem ser sintetizados de modo a fazer com que o funcionamento obedeça a determinadas restrições de operação. Neste trabalho, é apresentado um método para especificação de controle para FMS utilizando o método formal de Rede de Petri a Objeto e os paradigmas de Programação Orientada a Objeto (POO). É apresentada também uma ferramenta computacional desenvolvida para realização do controle de alto nível da FMS, baseando-se no modelo em Rede de Petri a Objeto. O método proposto, assim como funcionamento da ferramenta computacional, é validado através da modelagem e especificação de controle em um FMS didático composto por nove células que realizam todas as etapas de produção do sistema. São discutidos ainda aspectos de funcionamento do sistema e análise de outros métodos como a utilização de Rede de Petri interpretada.*

Palavras-chave: *Rede de Petri a Objeto, Especifição de controle, Programação Orientada a Objeto(POO) , Modelagem.*

1. INTRODUÇÃO

Sistemas a eventos discretos (SEDs) são aqueles em que as mudanças de estados se dão pela ocorrência de eventos instantâneos, como por exemplo, o início ou o final de operação de uma máquina. Assim como os eventos, os estados de tais sistemas também são definidos por variáveis discretas. (LIMA II, 2002)

Muitos SEDs, como é o caso notadamente dos Sistemas de Manufatura, são compostos por subsistemas cujo funcionamento conjunto deve obedecer a uma série de restrições de coordenação, como, por exemplo, impedir que seja iniciada a operação de uma máquina antes de a peça bruta ser posicionada. A especificação de controle visa justamente a coordenar o funcionamento do SED de modo a assegurar o respeito a tais restrições. Este controle possui, assim, uma natureza permissiva, agindo apenas no sentido de inibir a ocorrência de eventos que levem à violação das restrições.

Uma das abordagens muito utilizadas para a modelagem de SEDs são as Redes de Petri, já que produzem modelos mais compactos e de mais fácil análise gráfica que outras abordagens. Contudo, em sistemas de manufatura mais complexos, os recursos utilizados para a transformação do produto manufaturado, bem como as próprias características do produto, ao longo do processo, podem levar a modelos em Redes de Petri de grande complexidade e de difícil análise, o que torna atraente a utilização de outros métodos, como os modelos em Redes de Petri de Alto Nível. Esse tipo de modelagem mantém o poder gráfico de modelagem e análise das Redes de Petri, adicionando a possibilidade de uma definição mais detalhada dos tipos de processos e características do produto manufaturado.

Este trabalho visa apresentar um método de modelagem formal utilizando rede de Petri de alto nível e a implementação da especificação do controle com base nesta modelagem juntamente com os paradigmas de Programação Orientada a Objeto (POO) para um sistema flexível de manufatura didático denominado de planta CIM.

2. SISTEMAS FLEXÍVEIS DE MANUFATURA

Um sistema flexível de manufatura incorpora sistemas automatizados de manipulação de materiais, robôs, máquinas ferramentas CNC, inspeção e tecnologia de grupo em um único sistema de produção cuja integração está sob controle de uma rede hierárquica de computadores. (FINE, 1989).

Em um sistema flexível de manufatura a unidade de controle é o componente responsável pela execução da estratégia de controle especificada. A interação da unidade de controle com os outros componentes ocorre através de uma rede de comunicação de dados, onde as tarefas que cada um deve realizar são enviadas em forma de mensagens/sinais de comandos, e as condições das tarefas de cada componente são recebidas em forma de mensagens de estado. Assim as tarefas destes componentes são comandadas e monitoradas pela unidade de controle com o objetivo de se cumprir um plano de produção previamente estabelecido. (POLITANO, 1993).

Nesse contexto, um elemento essencial de integração entre os sistemas é o computador. Por meio de redes, um computador central, rodando um software SCADA (Supervisory Control And Data Acquisition) é capaz de se comunicar com o processador de cada elemento do processo produtivo (CLPs, CNCs, robôs, elementos transportadores etc.), de modo a coordenar todas as operações com o objetivo de produzir peças de acordo com especificações previamente estabelecidas. (DANEEL e SALTER, 1999).

Comunicando-se com os equipamentos e controladores locais de cada elemento do FMS, o SCADA é capaz de colher dados, alterar valores de referência (set-points) dos processos, comandar a troca de programas e iniciar e interromper tarefas. Um sistema SCADA pode possuir três funções básicas, de acordo com o grau de complexidade do sistema:

- **Supervisão**: o sistema SCADA apenas monitora o processo, colhendo e processando dados e gerando gráficos, relatórios etc. e exibindo-os ao usuário.
- **Operação**: nesse caso, o SCADA funciona como um painel de controle para interação com o operador. É possível ligar e desligar equipamentos, mudar o modo de operação desses, alterar parâmetros de controle de um processo.
- **Controle**: o sistema atua diretamente sobre os sistemas por meio de sinais de entrada e saída, sem depender de um nível intermediário composto de CLPs. Dessa forma, o processamento é realizado remotamente pelo SCADA.

A Figura 1 mostra a hierarquia de um sistema de manufatura gerenciado por um sistema SCADA.

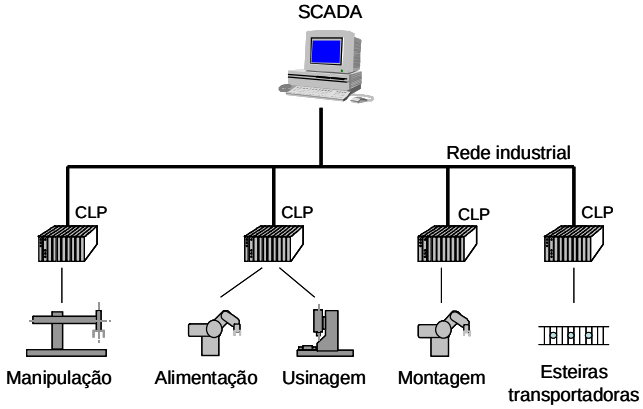

Figura 1 Sistema de manufatura gerenciado por um SCADA.
Fonte: Ribeiro e Lima II, 2005

3. SISTEMAS FLEXÍVEIS DE MANUFATURA

O laboratório denominado de Planta CIM é um FMS didático que foi montado a fim de proporcionar um ensino inovador no estado da Bahia, na área de mecatrônica. A Planta CIM, apesar de ser didática, dispõe de uma variedade de equipamentos industriais. Entre os equipamentos e sistemas instalados, estão: Robôs manipuladores industriais de sistema de transporte, sistema de inspeção por visão computadorizada e por contato, célula de tratamento superficial de peça, centros de usinagem que dispõe de torno e fresa industriais e didáticos, célula de transporte por veículo motorizado (Automatic Vehicle Guide-AVG), célula de armazenamento e recuperação automática de peças (AS/RS), célula de montagem com dispositivos de prensa.

A Planta CIM, no que se refere à produção, tem como produto final um objetivo denominado de “Porta Objeto”, ou seja, um objeto composto por quatro peças que tem a finalidade de guardar objetos de escritório como canetas, lápis, clipes etc. A Figura 2 ilustra este objeto.

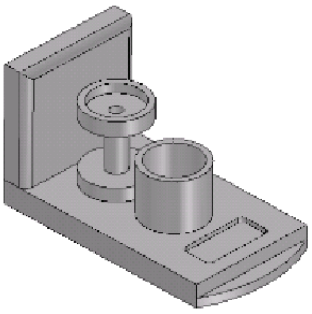

Figura 2 Porta Objeto

A Figura 3 mostra a arquitetura da planta CIM.

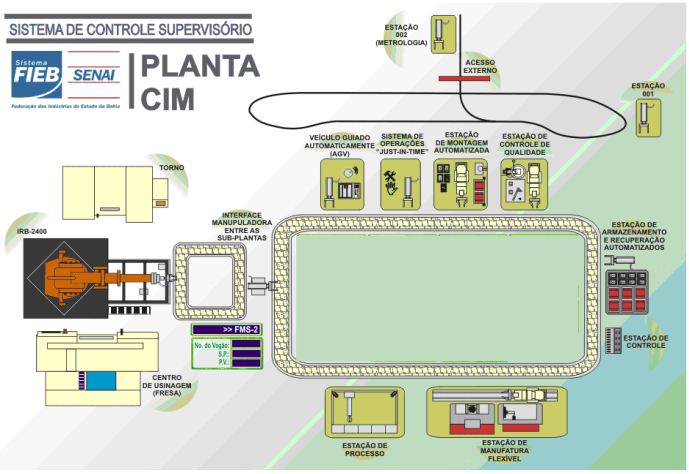

Figura 3 Arquitetura FMS Didático

A arquitetura de controle é composta por um CLP do fabricante SIEMENS, responsável pelo controle da planta e de toda a comunicação de sinais entre o controlador principal com as demais estações que fazem parte do sistema. Este controle é feito de forma centralizada, onde este controlador faz identificação das peças que chegam às outras células,as tarefas que devem ser executadas por cada célula e as regras de produção da planta. O CLP principal estabelece duas formas de comunicação entre os dispositivos. São elas: Comunicação através de sinais digitais de E/S(entradas/saídas) com as células que não possuem CLPs como:Sistema Flexível de Usinagem I e II, controle de qualidade, montagem, Just In Time (JIT) e a Veiculo Automaticamente Guiado (AGV).

As peças, para serem transportadas entre as estações, são dispostas sobre paletes que possuem pinos metálicos para a fixação destas. Qualquer palete é capaz de armazenar qualquer peça. Estes paletes, por sua vez, para se deslocarem,são colocados sobre transportadores unitários que no caso destes sistema são denominados vagões que estão dispostos sobre uma esteira transportadora.

4. REDES DE PETRI DE ALTO NÍVEL

As tentativas de utilização das redes de Petri, em problemas práticos reais, principalmente problemas computacionais, em que se trabalha com uma grande massa de dados, mostraram duas desvantagens principais (VILLANI, 2004). Em primeiro lugar, não existe o conceito de dados e os modelos tomam-se excessivamente grandes porque toda a manipulação de dados tem que ser representada através da estrutura da rede (isto é, através de lugares e transições). Em segundo lugar, não existe noção de hierarquia e, portanto, não é possível construir um modelo de grande porte através de um conjunto separado de sub-modelos com interfaces bem definidas.

Para resolver estes problemas, foram propostas as redes de Petri de alto nível, onde procura-se incorporar estruturas de dados e de composição hierárquica ao modelo original das redes de Petri (VILLANI, 2004). Entre as redes de Petri de alto nível, encontram-se as redes de Petri Coloridas (JENSEN, 1997), as redes Predicado-Transição (GENRICH,1987) e as redes de Petri a objetos (LAKOS, 1995).

Um sistema de manufatura caracteriza-se principalmente pela capacidade de fabricar vários tipos de produtos (processos de fabricação) simultaneamente e também por permitir rapidamente a modificação do plano de fabricação, seja quanto ao número de peças de cada processo, seja do próprio processo (CARDOSO e VALETTE, 1997). Existe, portanto, um componente de dados importante (tipo de peça a ser fabricada, estado atual da peça etc.), em que a dinâmica tem papel fundamental (seqüência de operações possíveis sobre um processo de fabricação, por exemplo). Dessa forma, torna-se atraente a utilização de Redes de Petri de alto nível para a modelagem de tais sistemas.

4.1 Rede de Petri a Objeto

As redes de a objetos são definidas sob a forma de redes de Petri ordinárias (redes subjacentes) munidas de inscrições. A análise é feita, primeiramente, sobre essas redes, da mesma forma como no caso das redes interpretadas.Os invariantes da rede de Petri subjacente fornecem resultados sobre a conservação de números de objetos independentemente de suas classes. (CARDOSO e VALETTE, 1997).

A Figura 4 mostra uma Rede de Petri a objetos que modela usuários (US) que fazem o processamento de peças (PC), utilizando máquinas (MAQ). As transições ini e fim representam, respectivamente, o início e final de operação.

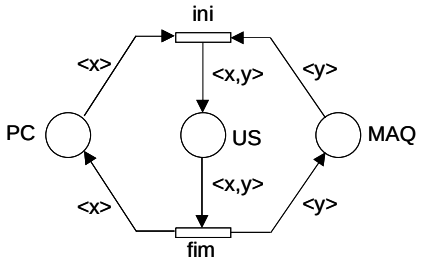

Figura 4 Rede de Petri a objetos.
Fonte: VARDOSO e VALENTE, 1997

Considerando o conjunto de classes Class = {peça, máquina} e que x são variáveis do tipo peça e y são variáveis do tipo máquina, que os atributos de peça incluem o tipo de operação a ser realizada e que os atributos de máquina incluem os tipos possíveis de operações, uma condição para o disparo de ini seria:

$$Atc(ini) = x.\text{operação} \in y.\text{operações}.$$

A marcação inicial da rede poderia ser representada por:

$$M_0 = \begin{bmatrix} < pc_1 > + < pc_2 > + < pc_3 > \\ 0 \\ < maq_1 > + < maq_2 > + < maq_3 > \end{bmatrix},$$

em que os objetos pc1, pc2 e pc3 são da classe peça e os objetos maq1, maq2 e maq3 são da classe máquina.

5. ABORDAGEM DE MODELAGEM E ESPECIFICAÇÃO DE CONTROLE PARA FMS UTILIZANDO REDE DE PETRI OBJETO

O método aplicado para o desenvolvimento sistemático do modelo do sistema e a sua especificação de controle deve auxiliar e orientar, de forma adequada, o projeto do controlador do sistema flexível de manufatura. O método para a modelagem utilizando rede de Petri a objeto e especificação do controle de um sistema flexível de manufatura estão descritos nos ítens abaixo. Para existe artigo é exemplificada a aplicação utilizando a esteira transportadora e os componentes da célula FMS II.

5.1 Definição das classes

Foram definidas as classes para cada componente do sistema para exemplificar a tabela 1 mostra a definição de classe para o componente peça.

Tabela 1 Definição da Classe Robô

Robô	**Nome da classe**
Propriedades	
nomeRobo: literal	Descrição do robô
modeloRobo: literal	Modelo do robô
lugar: inteira	Lugar da rede de Petri onde o objeto está localizado

Métodos	
void setNome (literal nome)	Método que modifica o atributo nomeRobo.
literal getNome ( )	Método que retorna o valor do atributo nomeRobo.
void setModelo (literal modelo)	Método que modifica o atributo modeloRobo.
literal getModelo ( )	Método que retorna o valor do atributo modeloRobo.
void setLugar (inteiro lugar)	Método que modifica o atributo lugar.
inteiro getLugar ( )	Método que retorna o valor do atributo lugar.
void inicoCicloRobo ( )	Método que informa à planta onde o robô deve começar sua operação.
void fimCicloRobo ( )	Método que informa ao controlador da rede que o robô finalizou o trabalho.

5.2 Diagrama de classe e Modelagem do controlador do Sistema Flexível de Manufatura denominado laboratório Planta CIM

A Figura 5 representa o diagrama de classe da célula sistema flexível de usinagem, onde é representada a relação de agregação inicialmente entre as classes vagão e palete (o que descreve o palete que está sobre o vagão para ser processado na estação), a agregação entre as classes robô e palete (que configura o transporte do palete com a peça para o buffer), a agregação entre as classes buffer e palete (que representa a chegada do palete no buffer), a agregação entre as classes robô e peça (que representa o transporte da peça para a máquina) e, por fim, a agregação entre as classes peça e máquina. O diagrama também mostra todos os atributos e métodos definidos para as classes.

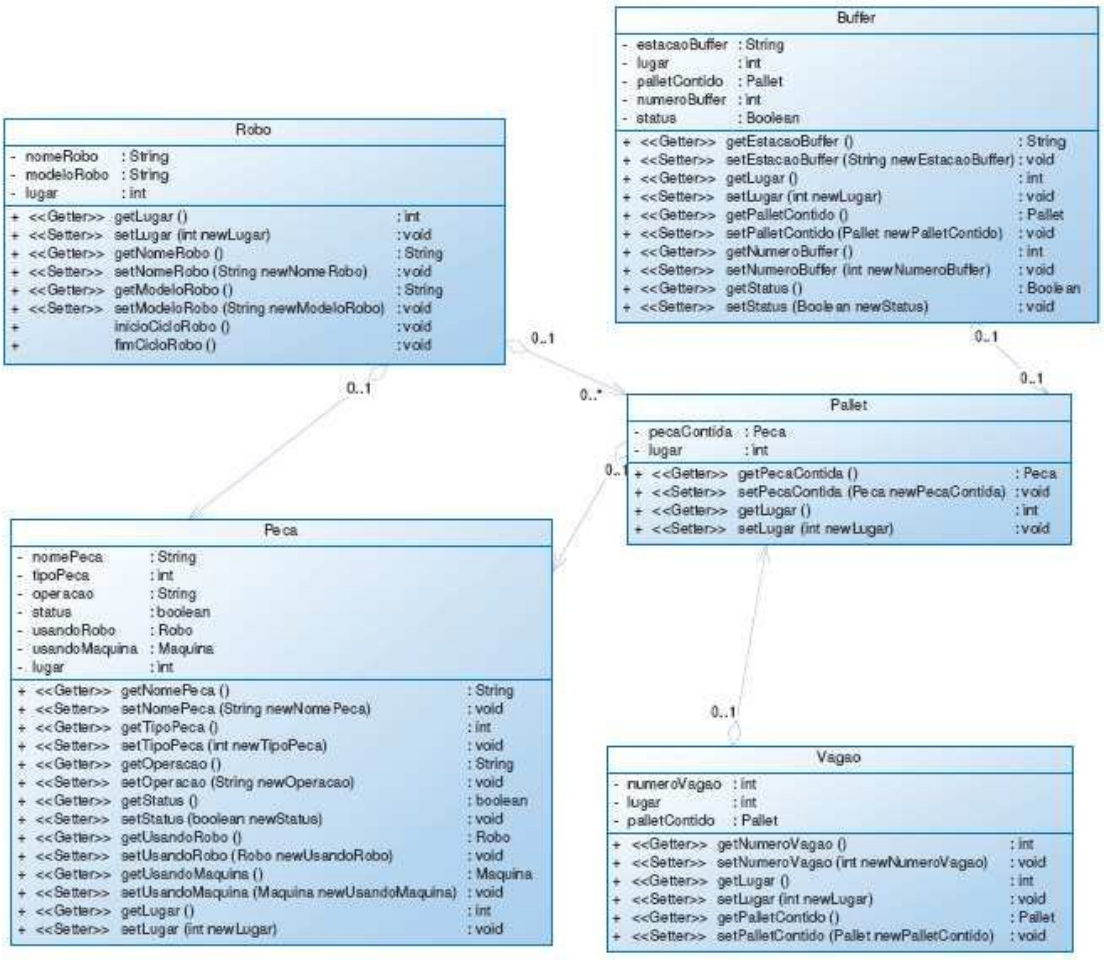

Figura 5 Diagrama de classe sistema flexível de maquina

Na Figura 6 é apresentado o modelo do controlador destas células, onde a transição não controlável t20 é uma transição que faz interseção com o modelo da esteira transportadora que é disparada quando chega um vagão com peça, para ser trabalhada na referida estação. O arco que habilita esta transição está representado de forma pontilhada para indicar que está conectada a um lugar que faz parte do modelo da esteira.

A transição controlável t21 é disparada quando existe um palete no lugar p20, um objeto robô existe no lugar p31, um objeto buffer existe no lugar p30. O palete, então, é transportada pelo robô e o método setRoboUsado (getNomeRobo()) é executado no objeto peça, através do objeto palete, que contêm uma instância de peça e o método inicoCicloRobo( ) é executado no objeto robô.

A transição não controlável t22 é disparada quando o robô terminar o transporte, executando o método fimCicloRobo( ) no objeto robô.

A transição controlável t23 dispara quando existe um objeto robô no lugar p31 e um objeto máquina no lugar p32, que execute a operação indicada no atributo operações da classe peça. Sendo assim, a peça é transportada pelo robô para a máquina e o método inicioCicloRobo( ) é executado na classe robô, o método setRoboUsado(getNomeRobo( ), na classe peça e o método preparaMaquina( ), na classe máquina.

A transição não controlável t24 é disparada quando o robô finaliza o transporte da peça para a máquina. Assim, o método fimCicloRobo( ) é executado na classe robô e o método inicioCicloMaquina( ), na classe máquina.

A transição não controlável t25 é disparada, quando a máquina chega ao final de seu ciclo de trabalho, logo é executado o método fimCicloMáquina( ) na classe máquina.

A transição controlável t26 é disparada quando existe um objeto robô no lugar p31, o método inicioCicloRobo( ) é executado na classe robô e o método setRoboUsado(getNomeRobo( )) é executado na classe peça.

A transição não controlável t27 é disparada quando o robô finaliza o transporte da peça para o palete. Logo, o método fimCicloRobo( ) é executado na classe robô e o recurso máquina é disponibilizado.

A transição controlável t28 é disparada quando existe, no lugar p31, um objeto robô disponível para transportar o palete para fora da célula. Quando chegar um vagão vazio na esteira, o arco que mostra esta ligação está pontilhado para indicar que é uma ligação com o modelo da esteira. Sendo assim, o método inicioCicloRobo( ) é executado na classe robô e o método setRoboUsado(getNomeRobo( )), na classe peça.

A transição não controlável t29 é disparada quando é finalizado o transporte do palete com a peça para o vagão palete para o vagão fora da célula, onde o método fimCicloRobo( ) é executado e o método setPaleteContido(palete) com as informações do palete e da peça que está sobre o mesmo. Esta operação irá se repetir para a devolução dos três paletes vazios.

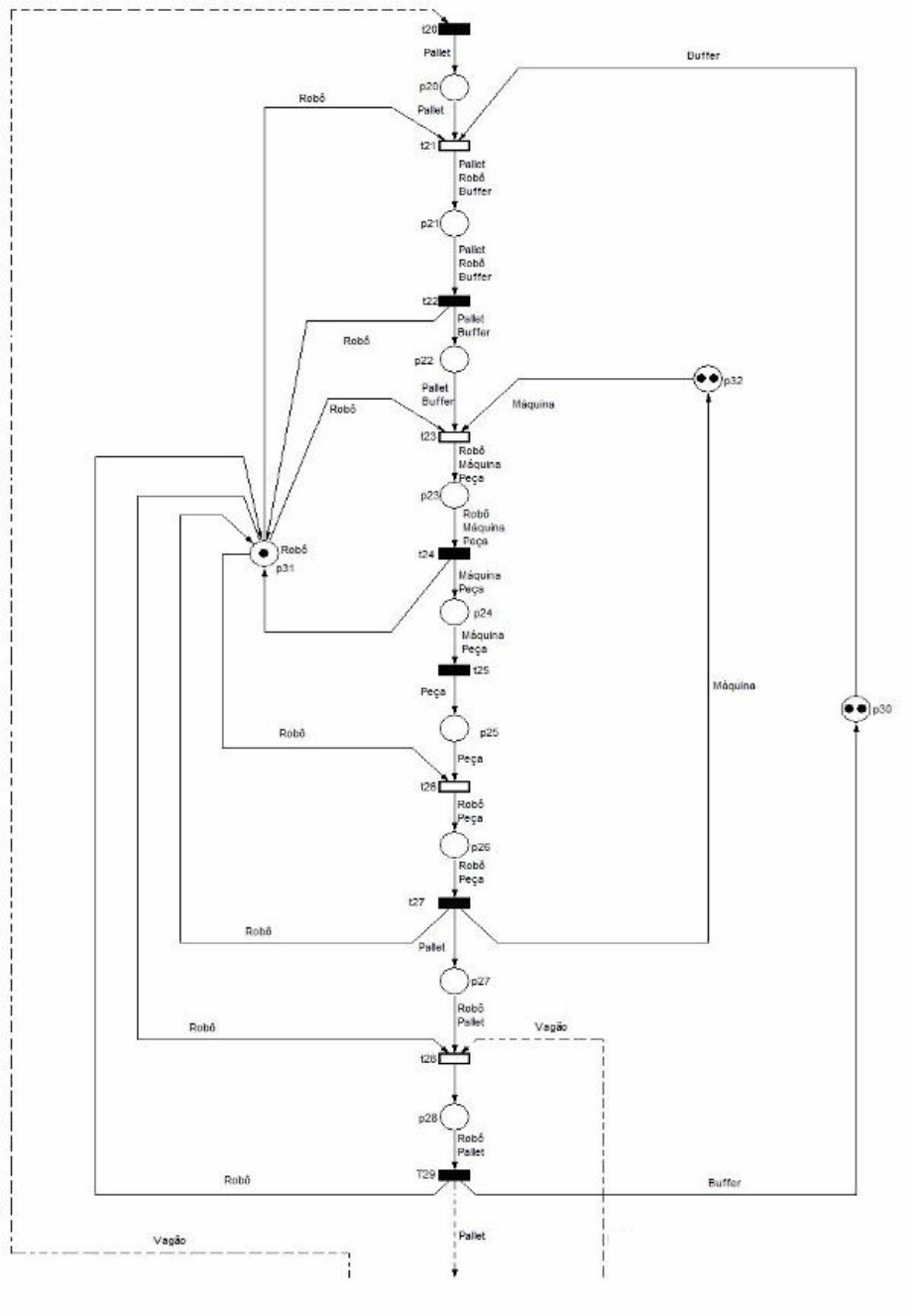

Figura 6 Modelo do controlador da célula de montagem

A Figura 8 mostra o modelo do controlador da esteira transportadora, dispositivo responsável pelo transporte dos vagões para as estações. Os lugares p90, p92, p94, p96, p98, p100, p102 e p104 são os lugares que representam a chegada do vagão na posição referente a uma das estações. Estes lugares fazem interface com os modelos das células através dos arcos que foram representados de forma pontilhadas no modelo do controlador de cada célula.

As transições controláveis t80, t82, t84, t86, t89, t91, t93, t95 são disparadas quando é iniciado o transporte de um vagão para a próxima estação, sendo executado o método inicioTransporte( ), na classe vagão.

As transições não controláveis t81, t83, t85, t87, t90, t92, t94 e t96 são disparadas quando o transporte do vagão chega ao final para uma referida estação. Desta forma, é executado um método fimTransporte( ) na classe vagão.

Os lugares p91, p93,p95, p97, p99, p101, p103 e p105 representam o espaço da esteira entre as células e nestes lugares pode haver mais de uma ficha. A retirada das fichas destes lugares é determinado pelos cinco sensores que estão posicionados à frente de cada célula. Estes identificam que vagão possui a peça que deve ser processada na referida célula, fazendo com que a transição não controlável subseqüente ao lugar seja disparada.

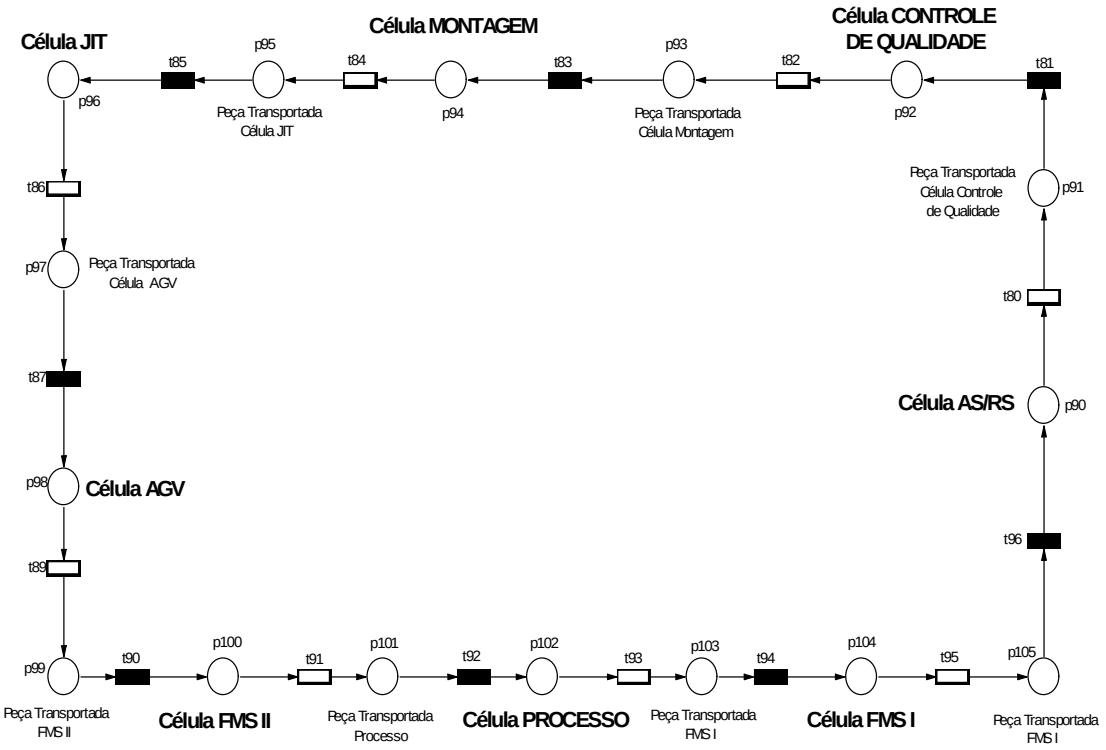

Figura 7 Modelo do controlador da Esteira.

5.3 Método de Controle

É proposto, aqui, um método de implementação de controle para FMS, utilizando-se o método formal de modelagem do controlador utilizando rede de Petri a Objeto e os a utilizar os paradigmas de POO. Desta forma, a implementação da especificação do controle do sistema dividido nos seguintes níveis:

- **Executor**: é responsável por fazer a atualização das marcações da rede de Petri objeto e enviar os sinais de inicio de ciclo dos componente que fazem parte do sistema para o controlador da planta.
- **Jogador**: é responsável de verificação as transições controláveis que se apresentam habilitadas Como também informar para o nível executor as transições habilitadas do sistema.
- **Gateway**: é responsável em fazer a conexão entre o PC onde esta implementado o modelo do controlador de alto nível e o controlador da planta. Como também verificar os sinais do controlador da planta que tiveram os seus valores modificados e informar ao nível jogador o vetor de transições não controláveis que estão habilitadas.
- **Controlador da Planta:** é responsável pelo controle de baixo nível da planta, ou seja, todas as etapas necessárias para a realização das operações.

Na Figura 8, é mostrada esta divisão e o fluxo de informações.

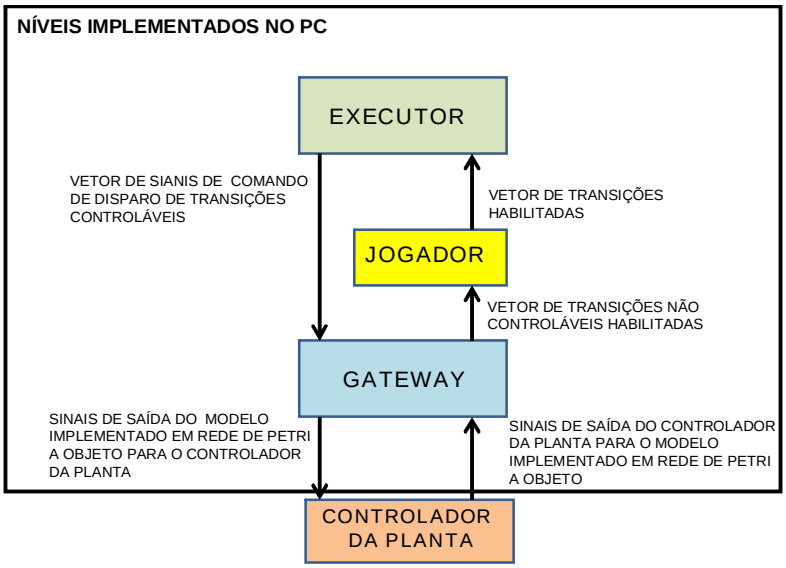

Figura 8 Estrutura do sistema de controle proposto

5.4 Implementação

Os níveis a serem implementados no PC, que estará conectado ao controlador de baixo nível da planta é o executor, o jogador e o gateway.

O ambiente de programação utilizado para a implementação do controlador de alto nível será o Delphi, por ser uma ambiente que permite implementar os paradigmas de POO.

A implementação da solução consiste em um sistema computacional desenvolvido em 3 camadas, sendo uma camada de conexão a qual é responsável entre a conexão da aplicação e o controlador de dispositivo da planta que é um CLP do fabricante SIMENS S7-300, a outra camada será a camada de negócio que é responsável pelas regras de evolução da rede de Petri a objeto que representa o modelo do controlador de alto nível da planta e a camada de apresentação que é responsável pela interface entre o sistema computacional implementado e o usuário. A Figura 9 mostra a arquitetura de camadas para o sistema implementado na planta CIM.

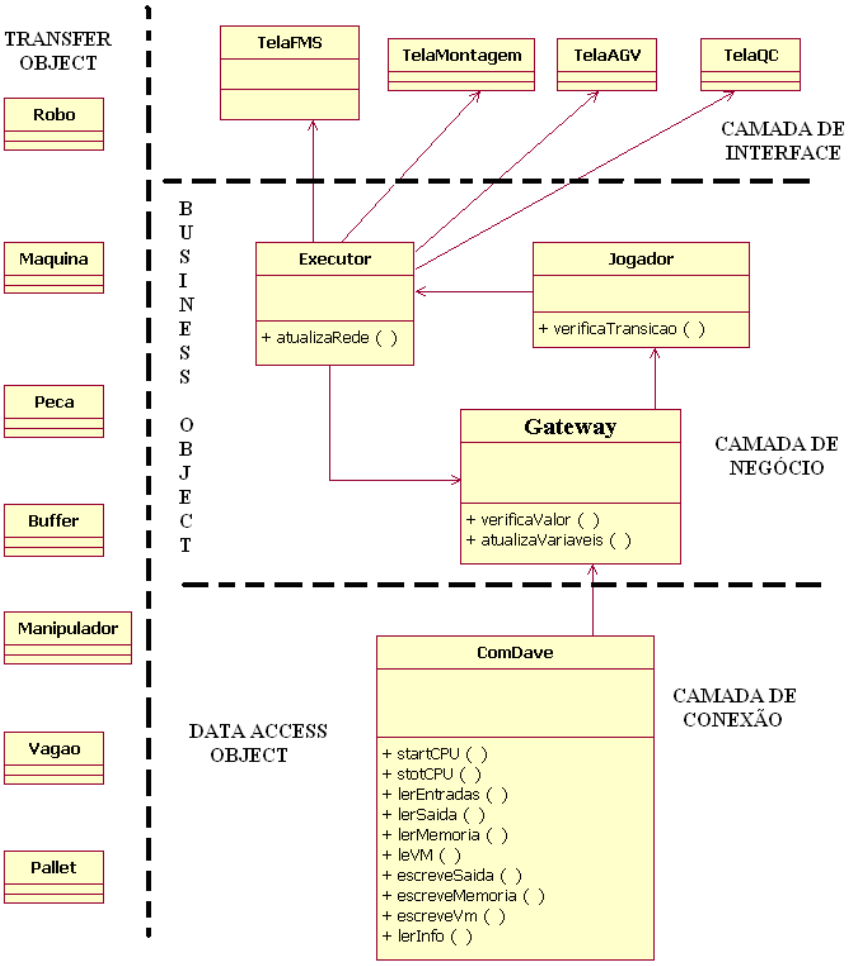

Figura 9 Arquitetura implantada no sistema da planta CIM

5.4.1 Camada de conexão

Para a implementação desta camada, foi utilizado o padrão de projeto denominado de Data Access Object (DAO) que abstrai e encapsula todo acesso à conexão com o CLP da planta, como também gerencia esta conexão. (ALUR,2004).

O DAO implementa mecanismo de acesso necessário para trabalhar com fontes de conexão com CLPs independente do seu tipo e da forma física da conexão utilizada. Para a implementação utilizada na planta CIM, a camada de conexão é constituída pela classe comNoDave, que utiliza uma biblioteca NoDave que é um software livre.

Nesta classe está implementado o código que permite a comunicação com o CLP S7-300. Através desta classe, é possível realizar a leitura dos sinais de entradas e escrita nos sinais de saídas físicas do controlador como também dos sinais de memória da CPU do mesmo.

5.4.2 Camada de Negócio

Nesta camada, é implementada a especificação de controle do modelo do controlador de alto nível que foi desenvolvido, utilizando o formalismo de rede de Petri a objeto, para solução de implementação para esta camada foi utilizado o padrão de projeto Business Object(BO), este padrão encapsula e gerencia as relações e interações das regras de negócios que têm como características o modelo de domínio conceitual, relacionamentos e estado dos objetos que é uma característica de predominância na modelagem, utilizando rede de Petri a objeto. Para a implementação foram criadas 3 classes são elas:

Gateway – esta classe é responsável por verificar se os sinais do CLP, que interagem com o modelo do controlador de alto nível, sofreram alguma modificação, como também informar à camada de conexão os sinais do programa do CLP que devem ser atualizados. Esta classe possui, como um dos seus atributos, um objeto do tipo comNoDave para, através dos seus métodos, fazer acesso às informações do CLP e possui dois métodos para implementar a lógica de verificação dos valores do sinais.

Jogador – Esta classe é responsável por receber os sinais que foram modificados no CLP da planta, através da classe gateway, fazer uma verificação no modelo do controlador de alto nível sobre quais transições estão no estado de habilitadas conforme as regras de controle especificadas no modelo da planta. Essa classe possui como um dos seus atributos um objeto do tipo da classe gateway e implementa um método chamado verificaTransição que recebe como parâmetro a lista de sinais que foram modificados no controlador de baixo nível da planta e com a interação com as classe que representam os objetos existentes foram descritas na tabela de definição de classe. Tais objetos possuem atributos e métodos que contêm a situação de disponibilidade destes recursos. A solução utilizada para disposição destas classes no software de controle de alto nível é a Transfer Object (TO), onde as classes estão disponíveis de formal transversal, ou seja, poderão ser acessadas por qualquer uma das três camadas do sistema.

Executor- esta classe é responsável por atualizar todos os estados do modelo do controlador de alto nível, prover informações para as classe da camada de interface e informar a classe gateway quais sinais do CLP devem ser atualizados obedecendo as restrições do modelo. Esta classe tem como atributos um objeto do tipo da classe gateway e um objeto do tipo de cada classe que implementa as telas do sistema. O método que esta classe implementa chama-se atualizaRede, este método recebe como parâmetro um vetor de transições habilitadas no sistema , faz a atualização das variáveis de memórias que deverão ser atualizadas no CLP, ativando o método atualizaVariaveis( ) que é um método que faz parte da classe gateway e o atualizaRede para atualizar a tela com a evolução da rede.

5.4.3 Camada de Interface

Na camada de interface são implementados os códigos responsáveis pelas interações gráficas do sistemas. Tais interações representam a evolução do modelo desenvolvido. O sistema possui uma tela para cada célula do FMS, sendo que a tela da respectiva célula possui a instância dos objetos que compõem a referida célula, para que permita a configuração do sistema, tendo como exemplo a tela da célula sistema flexível de usinagem I, onde existem objetos do tipo robô, máquina, buffer, palete e peça. Através desta tela é possível configurar todos os atributos que caracterizam estes objetos. A Figura 10 mostra a tela da célula flexível de usinagem II.

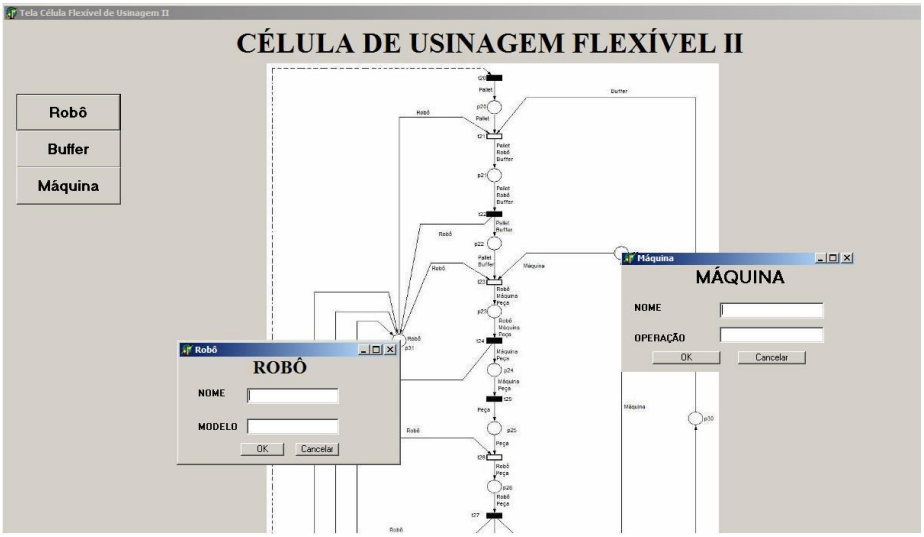

Figura 10 Tela da célula sistema de usinagem flexível II

6. CONCLUSÃO

Pôde-se comprovar que a abordagem baseada na Rede de Petri a objeto se apresenta como uma solução que facilita os processos de especificação e implementação do controle, bem como, a análise de sistemas flexíveis de manufatura. Esta abordagem tem como vantagem, a possibilidade de manter a flexibilidade do sistema associando as

características dos componentes que estão envolvidos como, por exemplo: máquinas e peças, no modelo que representa o sistema real.

O seu funcionamento obedece às restrições de controle estabelecidas em projeto e facilita a implementação, utilizando programação orientada a objeto que possui também a característica de implementar em uma ferramenta computacional instância de objetos, com as características dos componentes que compõem o sistema real.

Neste artigo foi apresentado um método para especificação de controle em sistema flexível de manufatura e os resultados da implementação prática desta técnica foram aplicados a uma célula de um FMS didático denominado de Planta CIM.

O funcionamento do sistema ocorreu conforme as restrições implementadas no software de controle. Foi possível observar que o sistema evoluiu sem a ocorrência de deadlocks, foi minimizada a ociosidade dos recursos envolvidos, através das regras de controle implementadas, como foi proposto no modelo apresentado para o FMS didático. Isto permitiu validar a aplicação prática da técnica de implementação de especificação de controle, utilizando o formalismo de rede de Petri a objeto integrado aos paradigmas de Programação Orientada a Objeto (POO). Com esta integração, foi possível criar no software que implementa a especificação de controle instâncias dos componentes que compõem o FMS com as mesmas características que foram denominadas de atributos e modificá-las através dos métodos de cada um deles durante a evolução do sistema real.

Um ponto importante a salientar é que a utilização dos paradigmas de POO, para a implementação da especificação de controle, facilitou a implementação da estratégia do software de controle que utilizou o sistema em três camadas, o que permite separar a aplicação em camada de conexão, a camada de negócio onde estão implementadas as regras do modelo do controlador do sistema e a camada de interface.

7. REFERÊNCIAS

ALUR,D. Core J2ee Patterns Melhores Praticas E Estrategias De Design. Rio de Janeiro: Campus, 2004.576p.

CARDOSO, J. e VALETTE, R. Redes de Petri. Ed. da UFSC, Florianópolis.1997

DANEEL, A. e SALTER, W. What is SCADA? International Conference on Accelerator and Large Experimental Physics Control Systems, Trieste, Italy.1999

FINE, C. H. Strategic manufaturing - dinamic new directions for the 1990s, Dow-JonesIrwin, Homewood, Illiois, USA, 312 Pp., 1989.

GENRICH. H. Predicate/transition nets. Lecture notes in Computer Science (Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986. Part I), v.254., p.207-247.1987

JENSEN, K. Coloured Petri nets: basic concepts, analysys methods and practical use. 2.ed. Springer-Verlag, Berlin.1997.

LAKOS, C. From Coloured Petri Nets to Object Petri Nets, Conference on the Application and Theory of Petri Nets, Torino, Italy.1995.

LIMA II, E.J. Uma Metodologia para a implantação atreavés de CLPs de controle supervisório de células de manufatura utilizando redes de Petri. Mestrado apresentado no programa de pós- graduação em engenharia elétrica da escola Politécnica da UFBa, 2002.

POLITANO, P. R. Especificação e Implementação de uma unidade de controle para célulaflexível de manufatura. São Carlos, Agosto, 1993, 130 Pp., Dissertação (Mestrado),Escola de Engenharia de São Carlos - EESC, Universidade de São Paulo - USP.

VILLANI, E. Modelagem e Análise de Sistemas Supervisórios Híbridos. Tese (Doutorado) – Universidade de São Paulo, São Paulo. 2004.

8. DIREITOS AUTORAIS

O autor é o único responsável pelo conteúdo do material impresso incluído no seu trabalho.

MODELLING AND CONTROL ESPECIFICATION FLEXIBLE MANUFACTURING SYSTEMS USING HIGH LEVEL PETRI NETS

Abstract: *Based on Object Petri Nets, control specifications for Flexible Manufacturing Systems (FMS) can be obtained, making the system functioning to obey determined operation constrains. In this work, it is presented a method to specify control rules to FMSs using Object Petri Nets formal methods and Object Oriented Programming (OOP) paradigm. It is also presented a computational tool developed to control the high level operations of the FMS based on the Object Petri Net model. The proposed method and the computational tool are validated through the modeling and control specification of a didactic FMS composed by nine production cells that carry all the production steps on a part. It also discusses functioning aspects of the system and analyze other methods as Interpreted Petri Nets.*

Keywords: Object Petri Nets, Control specification, Object Oriented Programming, Modeling.

1. REPONSIBILITY NOTICE

The Authors are the only responsible for the printed material included in this paper.