



AVALIAÇÃO DE DESEMPENHO DE ARQUITETURAS DE COMANDO

Luís Clovis L. Viana

João Maurício Rosário

Laboratório de Automação Integrada e Robótica

Universidade Estadual de Campinas, Departamento de Projeto Mecânico

Cx. P. 6122 – 13083-970 – Campinas, SP, Brasil

Jean-Marc Faure

Laboratoire d'Ingénierie Intégrée des Systèmes Industriels

CESTI-ISMCM

Maison des Technologies – Place Georges Pompidou – Quartier Mayol

F83000 – Toulon, France

Resumo. *O projeto de Sistemas Automatizados de Produção requer, entre outras atividades, a especificação de uma arquitetura de comando. Principalmente em sistemas com estrutura de controle-comando bastante distribuída, esta especificação é determinante para todo o restante do ciclo de vida do sistema em questão. Nesses casos, o projetista vê-se face à difícil tarefa de, numa fase inicial de projeto, escolher uma arquitetura de comando capaz de atender a pré-requisitos funcionais como tempos de resposta do sistema. Este trabalho apresenta uma metodologia para validar um modelo de arquitetura de comando distribuída através da construção de modelos de Análise Estruturada e de sua posterior simulação sob a forma de Redes de Petri Coloridas e Temporizadas. Como estudo de caso, esta metodologia será aplicada a uma plataforma didática em automação.*

Palavras-chave: *Comando distribuído, Simulação, Redes de Petri Coloridas e Temporizadas, Características temporais*

1. INTRODUÇÃO

O ciclo de vida de um Sistema Automatizado de Produção (SAP) (Verdin, 1991) começa com uma *fase de especificação e projeto*. Esta fase consiste nas seguintes atividades:

- análise das necessidades do sistema, resultando na definição de seu *Caderno de Tarefas*;
- projeto preliminar do sistema, com a especificação de suas propriedades gerais;
- a especificação e projeto da *parte de comando*;
- a especificação e projeto da *parte operativa*;
- diversas etapas de validação, pois cada atividade de projeto gera modelos que devem ser validados antes de passar à *fase de realização do sistema* e ao restante de seu ciclo de vida.

O **projeto da arquitetura de comando** do SAP consiste em definir a repartição das funções de comando do sistema entre os diversos equipamentos de controle-comando disponíveis.

Esta atividade pode ser posicionada entre as atividades de *especificação e projeto da parte de comando*, ou seja, no meio da fase de projeto e especificação. No entanto, este posicionamento é indicado para sistemas de arquitetura de comando simples, muitas vezes reduzida a um só equipamento de controle.

Atualmente, visando melhores níveis de flexibilidade e reatividade no meio industrial, os sistemas de controle-comando de um SAP são freqüentemente organizados de forma distribuída: as partes operativas comandadas por diversos Controladores Lógicos Programáveis (CLP), estes, por sua vez, muitas vezes dirigidos por um sistema de supervisão; as trocas de dados entre as diversas estruturas físicas de comando sendo realizadas por meio de uma rede local.

Assim, para SAPs com estruturas de comando fortemente distribuídas, a escolha de uma arquitetura de comando adequada torna-se crucial para todas as atividades seguintes de seus ciclos de vida. O projetista deve, então, considerar o projeto da arquitetura de comando suficientemente cedo durante a fase de projeto e especificação do sistema em questão: mais precisamente, durante o **projeto preliminar do sistema**.

Este trabalho se propõe a adaptar uma metodologia de projeto de arquitetura de comando, extraída dos trabalhos de Meunier e Denis (1997), para uma aplicação concreta de comando de um SAP, no caso um dos postos de trabalho de uma plataforma didática em automação (Rosário *et al*, 1995).

2. O PROJETO DA ARQUITETURA DE COMANDO

O objetivo principal da escolha de uma arquitetura de comando para um dado sistema é determinar que componentes serão utilizados e como as diversas tarefas e pré-requisitos funcionais da aplicação em questão serão repartidas no conjunto de equipamentos disponíveis. Visando considerar tanto o aspecto funcional como o aspecto material da parte de comando, a arquitetura de comando de um sistema distribuído pode ser modelado segundo três pontos de vista distintos (Meunier et Denis, 1997): *arquitetura funcional*; *arquitetura material* e *arquitetura operacional*.

2.1 Arquitetura funcional

A arquitetura funcional representa o conjunto dos tratamentos de informações proveniente dos pré-requisitos funcionais de comando. Seus modelos são geralmente construídos como se a estrutura física do sistema fosse constituída de um só equipamento, ou seja, não se leva em conta sua distribuição física. Ela deve assegurar as seguintes funções:

- representar todas as funções (*funções de tratamento de dados e funções de controle*) realizadas pelo programa (software) de comando assim como todos os dados que devem ser estocados ou circular entre estas;
- determinar o comportamento das *funções de controle*, isto é, as funções que geram os eventos necessários à ativação de outras funções;
- determinar o tempo de vida dos diferentes dados, ou seja, se os dados são consumidos na transferência entre duas funções (função produtora e consumidora de dados), ou se é preciso prever um ponto de estocagem para a perenidade da informação.

Diferentes métodos, tais como SA-RT, *diagramas entidade-relação*, entre outros (Jaulent, 1990), podem ser utilizados para descrever integralmente a arquitetura funcional. Tradicionalmente, opta-se pela abordagem conhecida como *Análise Estruturada* (Yourdon,

1989) para expressar a arquitetura funcional através de um conjunto coerente de modelos *funcionais* (que descrevem as funções do sistema de comando), *comportamentais* (que descrevem a evolução e ativação destas funções) e *de informação* (que descrevem a relação entre os dados).

A Figura 1 mostra um exemplo de modelo de arquitetura funcional. Neste exemplo, como no decorrer deste trabalho, são utilizadas variações bem aceitas no campo da automação industrial do *diagrama de fluxo de dados* (funcional) e do *diagrama estado-transição* (comportamental): respectivamente, o método SADT (*Structured Analysis and Design Technique*) (Ross, 1977) e o GRAFCET (IEC, 1988) (*Graphe Fonctionnel de Comande Etape-Transition*).

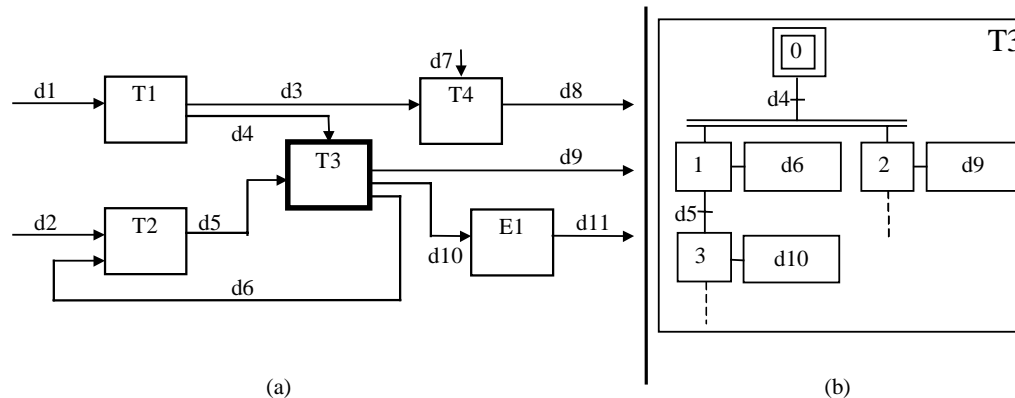


Figura 1 - Modelo esquemático de arquitetura funcional: (a) fluxo de dados entre funções; (b) GRAFCET da função de tratamento T3.

2.2 Arquitetura material

A *arquitetura material* (Fig. 2) é uma representação dos meios físicos disponíveis para responder a um problema específico. Estes meios são equipamentos de controle-comando e de informática industrial.

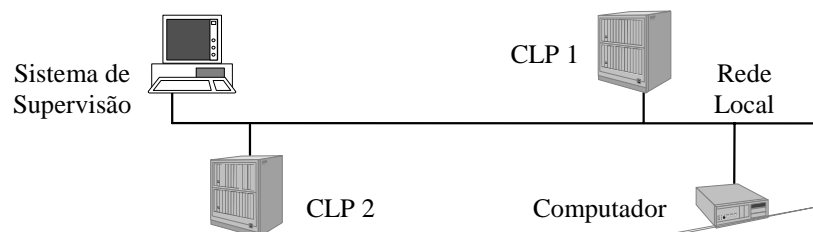


Figura 2 - Exemplo de arquitetura material.

Apesar da diversidade dos equipamentos utilizáveis em um SAP, os diferentes componentes da arquitetura material podem ser agrupados em categorias genéricas:

- processadores (controladores lógicos programáveis (CLP), computadores industriais diversos, sistemas de supervisão, etc.);
- interfaces homem-máquina (consoles de comando, terminais de diálogo, sistemas de supervisão, etc.)
- meios de comunicação (ligações ponto-a-ponto, redes, etc.)

2.3 Arquitetura operacional

A **arquitetura operacional** representa a repartição dos tratamentos e dos dados do sistema de comando nos equipamentos disponíveis através da associação das entidades da arquitetura funcional com os componentes da arquitetura material. A Figura 3 mostra um exemplo de arquitetura operacional derivado dos exemplos precedentes (Fig. 1 e 2).

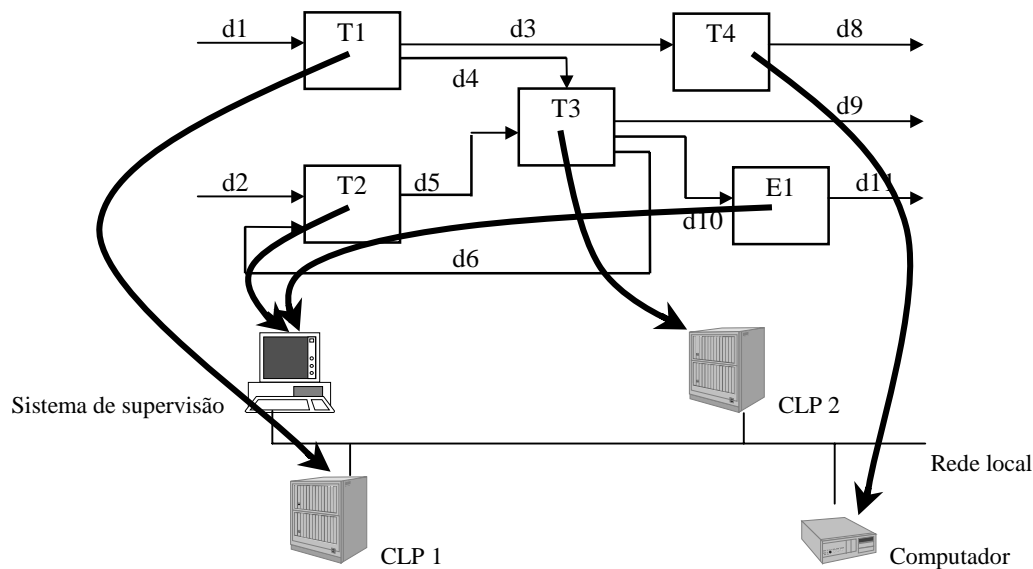


Figura 3 - Exemplo de arquitetura operacional.

Para cada um dos tratamentos (T_i), determina-se qual processador deve assegurar sua execução. Cada função de estoque de dados (E_i) pode ser designada ou para um equipamento de tratamento (para sua área de memória), ou para um equipamento específico de armazenamento de dados. Os fluxos de dados circulam entre os diferentes tratamentos e estoques através de um meio de comunicação, neste caso uma rede local.

Neste estágio do projeto da arquitetura de comando, diversas *arquitecturas operacionais* podem ser implementadas. Cada uma delas apresenta comportamentos distintos, notadamente no que diz respeito a tempos de resposta do sistema. O projetista precisa validar o comportamento dinâmico dos modelos de arquitetura operacional para em seguida poder avaliá-los quanto a seu desempenho temporal e, finalmente, definir a arquitetura de comando para o SAP em questão.

3. VALIDAÇÃO DO COMPORTAMENTO TEMPORAL ATRAVÉS DE REDES DE PETRI COLORIDAS E TEMPORIZADAS

Para validar o comportamento temporal das diferentes arquiteturas operacionais deve-se construir um modelo dinâmico através do acoplamento dos modelos dos diferentes tratamentos e dos diferentes componentes. Estes modelos devem ser genéricos para poderem ser livremente acoplados entre si segundo as diferentes arquiteturas propostas.

As *Redes de Petri (RdP) Coloridas e Temporizadas* são particularmente adequadas à construção de tal modelo dinâmico. A *coloração* permite criar modelos genéricos, pois uma sub-rede pode ter uma estrutura (*lugares, transições e arcos*) única, independentemente de seu posicionamento na arquitetura: apenas as diferentes cores das marcas (representando, por exemplo, diferentes tipos de dados) indicam seu papel em relação às outras sub-redes. Já a *temporização* é indispensável para avaliar o desempenho do modelo dinâmico.

Para simular e avaliar os modelos de arquitetura operacional, optou-se pelo programa *Design/CPN*, software mantido pelo grupo *CPN* da Universidade de Aarhus, Dinamarca.

Design/CPN é um editor e simulador de RdP Coloridas e Temporizadas. Além disso, o programa permite a construção de modelos estruturados através dos seguintes mecanismos (Reisig, 1985) (Jensen & Rozenberg, 1991):

- *transições* podem conter em seu interior sub-redes, o que permite uma modelagem hierárquica do tipo *top-down* ou *bottom-up*;
- *lugares* de sub-redes diferentes podem ser fundidos entre si como se fossem um só, mesmo estando em regiões diferentes do modelo, o que permite que sub-redes distintas sejam ligadas entre si formando uma rede única.

As sub-redes modelizando as **funções**, provenientes da arquitetura funcional, são acopladas às sub-redes dos **equipamentos**. A Figura 4 mostra uma situação derivada dos exemplos anteriores (Fig.1,2 e 3), em que o tratamento T3 é projetado no CLP 2.

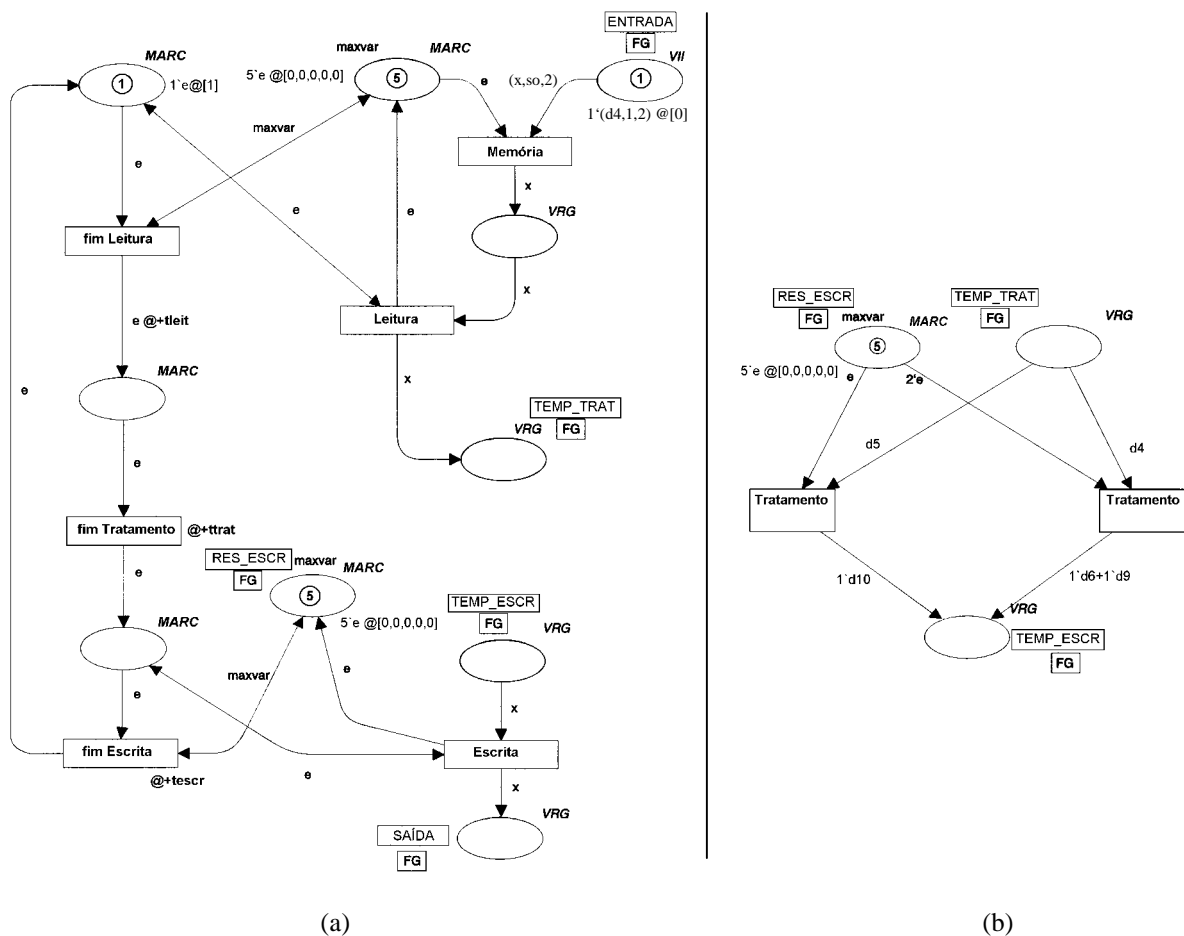


Figura 4 - Modelo dinâmico da projeção de uma função em um equipamento da arquitetura material: (a) modelo do CLP 2; (b) modelo do tratamento T3.

A Figura 4 utiliza a sintaxe de Design/CPN. Mesmo se estas sub-redes estão localizadas em regiões distintas do modelo, alguns de seus lugares estão conectados logicamente através do mecanismo de *fusão de lugares*. Estes lugares são assinalados com a inscrição **FG** (significando *Fusão Global*) e estão ligados a suas "imagens" (por exemplo, os lugares assinalados com **FG** **RES_ESCR** constituem o **mesmo lugar** e portanto contêm as mesmas cinco marcas).

Em Design/CPN as *marcas* carregam consigo um selo de tempo, indicando o instante em que chegaram a seu último *lugar*. Por meio destes selos, durante a simulação os tempos dos

fluxos de dados podem ser coletados ou mesmo plotados para a avaliação de uma arquitetura de comando adequada.

Basicamente, a seqüência de três lugares e três transições (*fim Leitura*, *fim Tratamento* e *fim Escrita*) na parte esquerda da Figura 4a modelam o ciclo padrão de um CLP. Uma marca *e* percorre ciclicamente estes lugares, e o tempo de permanência desta marca em cada uma das partes do ciclo do CLP é determinado sucessivamente por *tleit* (tempo de leitura), *ttrat* (tempo de tratamento) e *tescr* (tempo de escrita). Na parte direita da Figura 4a ocorre o fluxo das marcas representando os dados do modelo, através dos lugares de coloração VII e VRG (em Design/CPN, os lugares têm colorações próprias, indicando a coloração das marcas que eles podem conter). A Figura 4b modela o comportamento lógico do tratamento T3 (de acordo com o GRAFCET mostrado anteriormente na Figura 1): uma entrada *d4* resulta em saídas *d6* e *d9*; enquanto uma entrada *d5* resulta em uma saída *d10*.

4. ESTUDO DE CASO: POSTO DE MONTAGEM DE PRODUTOS

Como aplicação desta metodologia, foi utilizado um dos postos de trabalho da plataforma didática em automação (Rosário *et al*, 1995) que é o resultado da colaboração entre o Laboratório de Automação Integrada e Robótica da UNICAMP e o *Laboratoire d'Ingénierie Intégrée des Systèmes Industriels* do CESTI-ISMCM de Toulon, França. A plataforma constitui um SAP pedagógico servindo de suporte experimental para pesquisas na área de automação.

4.1 Caderno de Tarefas

A plataforma realiza a montagem automática de peças *LEGO*[®]. Ela é formada por quatro *postos de trabalho* e um *sistema de transferência*, que interagem para montar corretamente uma determinada seqüência de cubos sobre uma placa maior. Cada placa contendo uma seqüência diferente de cubos é considerada um **tipo de produto** distinto (Fig. 6a).

O *Posto de Montagem Central* deve montar até dois cubos sobre a posição **central** da placa, de acordo com um código afixado a ela (código de barras ou etiqueta magnética), o qual especifica o tipo de produto a ser realizado pela plataforma. O funcionamento do Posto de Montagem Central pode ser descrito basicamente pela seguinte seqüência de operações (Fig. 6b):

- a placa chega através da esteira do sistema de transferência e seu código é lido;
- ela é em seguida levantada por *Elevador* e levada até o interior do posto pelo atuador *Gaveta Inferior*;
- o conjunto de atuadores da parte superior encarrega-se de manipular os cubos estocados no posto (*Gaveta Superior* e *Posicionar Cubo*) e de montá-los (*Colocar Cubo*);
- *Gaveta Inferior* coloca a placa já montada sobre *Elevador*, que por sua vez a desce novamente até o sistema de transferência.

4.2 Arquitetura de comando

Arquitetura funcional. As principais funções de comando do Posto de Montagem Central foram modeladas por meio do método SADT. Um modelo destas funções consiste em (Fig. 7a):

- “**processar código**”, isto é, identificá-lo e transmitir a informação do tipo de produto às outras funções do comando;

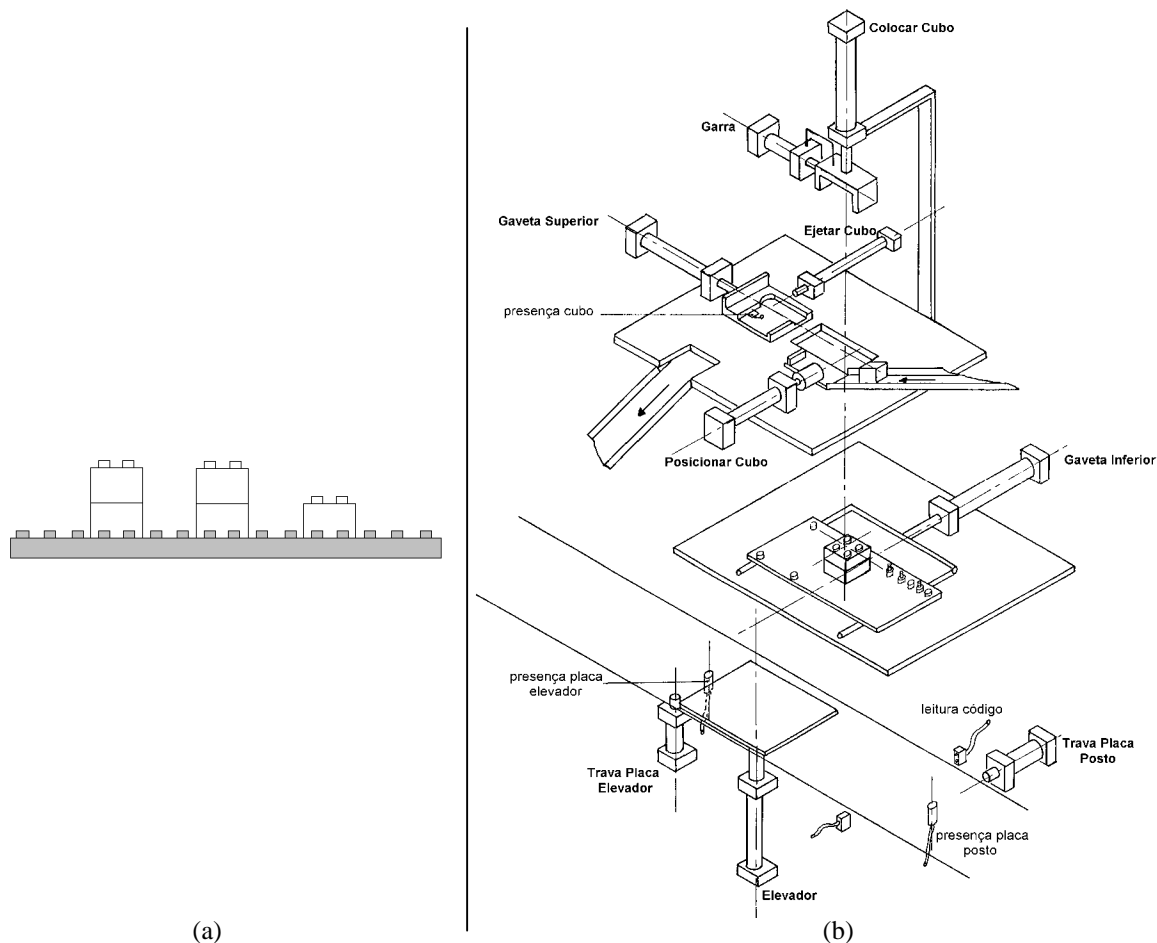


Figura 6 - (a) Exemplo de produto da plataforma; (b) Estrutura do Posto de Montagem Central.

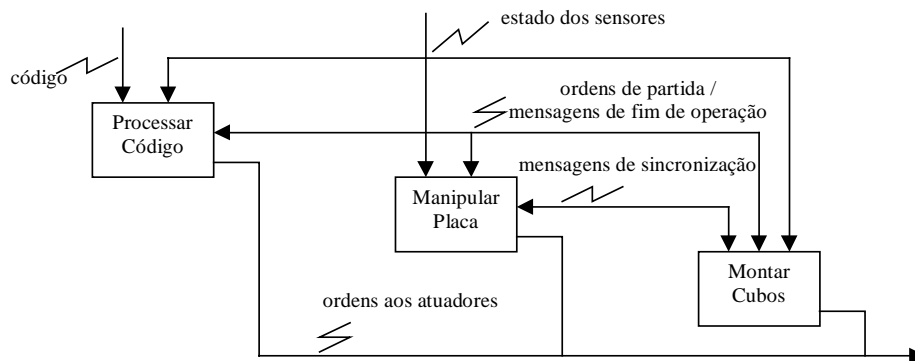
- **“manipular placa”**, isto é, comandar as operações alternadas dos atuadores *Elevador* e *Gaveta Inferior*, ou seja, o recebimento da placa e sua evacuação depois de montado o produto;
- **“montar cubos”**, isto é, comandar os atuadores e sensores responsáveis pela montagem dos cubos.

Estas duas últimas funções são interdependentes e trocam entre si mensagens de sincronização que asseguram a coordenação entre o posicionamento da placa dentro do posto e a colocação dos cubos sobre esta.

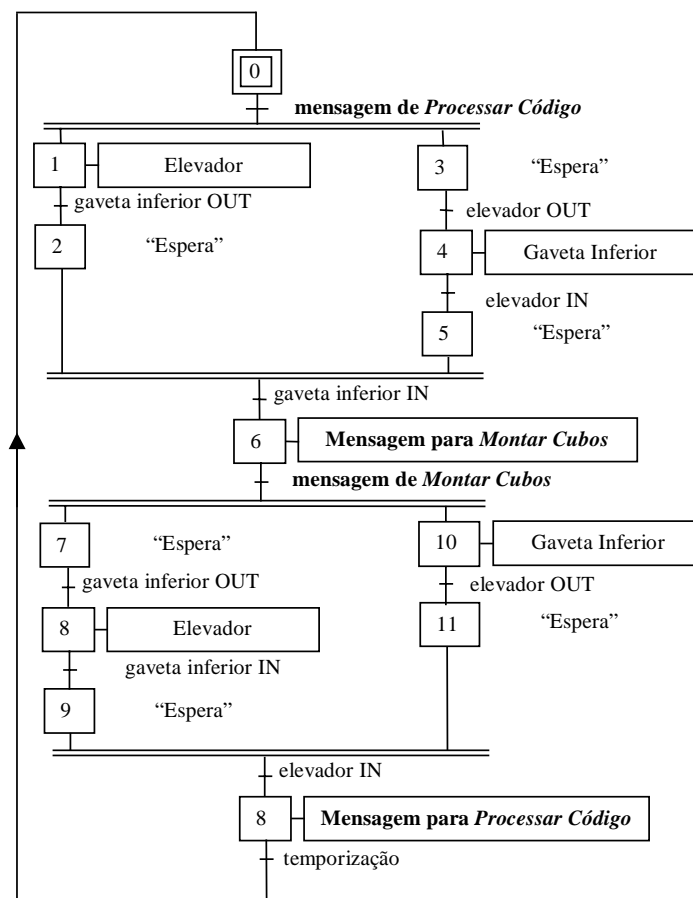
O comportamento dessas funções, sobretudo a seqüência dos fluxos de dados entre elas é expresso pelos respectivos GRAFCETs (Fig. 7b).

Arquiteturas materiais. Duas configurações de arquitetura material são propostas para executar o comando do posto:

- a primeira consiste em **três CLPs** ligados por rede *UNITELWAY*, cujo protocolo tem controle de acesso do tipo *mestre/escravo*;
- a segunda arquitetura material, por sua vez, em **dois CLPs e um módulo programável** dedicado para leitura/gravação de etiquetas magnéticas, ligados entre si por rede *UNITELWAY*; o módulo programável suporta conexão nesta rede apenas como escravo.



(a)



(b)

Figura 7 - Arquitetura funcional do Posto de Montagem Central: (a) *actigrama* SADT; (b) GRAFCET da função *Manipular Placa*.

Arquiteturas operacionais. São os cenários a serem simulados e avaliados, e derivam da projeção das funções sobre as duas arquiteturas materiais anteriores. Assim, temos a “**Arquitetura A**” como sendo:

- função *Processar Código* no CLP mestre da rede;
- função *Manipular Placa* e *Montar Cubos* em cada um dos CLPs escravos.

E, como “**Arquitetura B**”:

- *Processar Código* no módulo programável (escravo);
- *Manipular Placa* no CLP mestre;
- *Montar Cubos* no CLP escravo.

4.3 Indicadores de desempenho

Como visto no modelo de arquitetura funcional (Fig. 7), existem fluxos de dados entre as funções da parte de comando. Ou seja, há comunicação entre elas por meio de mensagens de ativação e/ou sincronização. Assim, as medidas de desempenho a serem coletadas durante a simulação são os tempos **entre causa e efeito em duas funções diferentes**. Por exemplo, o tempo entre a identificação da chegada da placa ao elevador (em *Processar Código*) e a ação de subir o elevador (comandada por *Manipular Placa*), estas estando separadas pela emissão e recebimento de uma mensagem (Fig. 8).

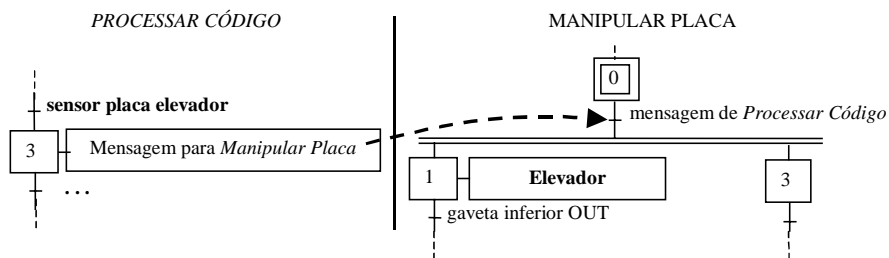


Figura 8 - Tempo de reação entre uma comunicação.

4.4 Simulação e resultados

Design/CPN também proporciona a construção de gráficos. A Figura 9a mostra uma transição (em preto) que foi adicionada ao modelo de CLP visto anteriormente (Fig. 4a). A esta transição foram associadas algumas linhas de código (indicadas por **C**) para inicializar as variáveis *tdebut2* e *tdebut4*, que por sua vez foram utilizadas para a construção dos gráficos da Figura 9b. Sempre que esta transição é transposta, o código associado é executado.

A Figura 9b mostra alguns dos indicadores de desempenho para as duas arquiteturas de comando propostas: o tempo decorrido entre o sinal do sensor *gaveta inferior IN* (da função Manipular Placa) e o acionamento do cilindro *Colocar Cubo* (da função Montar Cubos); e o tempo decorrido entre a detecção de elevador IN (da função Manipular Placa) e a sinalização de Fim de Ciclo (da função Processar Código). Nos dois casos, como a rede *UNITELWAY* é

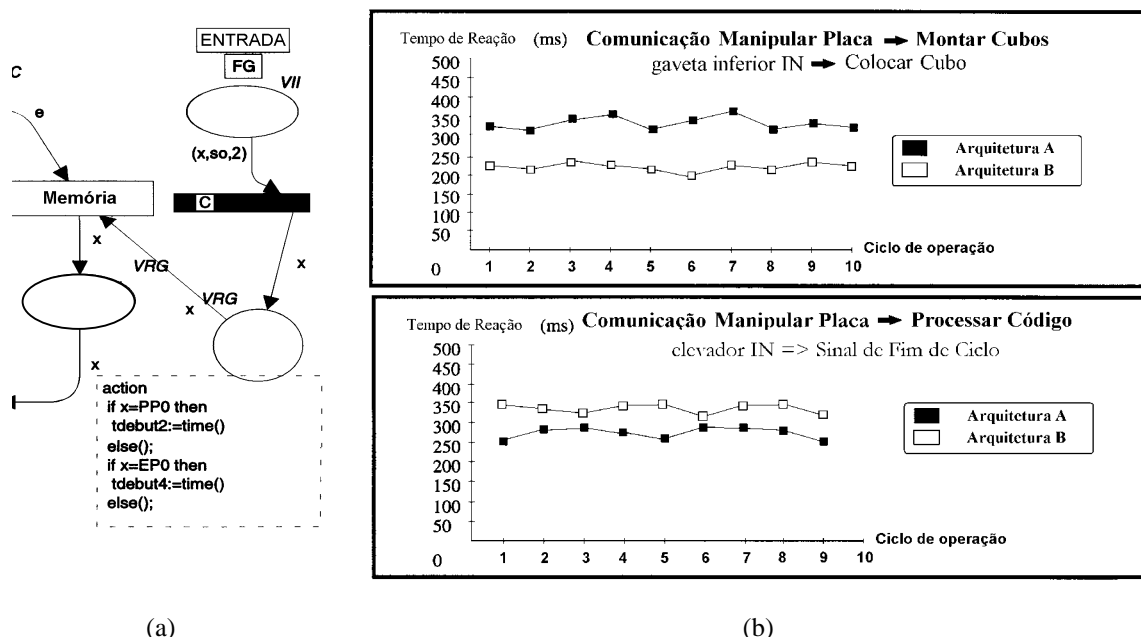


Figura 9 – (a) Código de programação associado ao "tiro" de uma transição; (b) desempenho de duas arquiteturas de comando.

uma rede com protocolo com *polling* cíclico, a hierarquia entre os diferentes equipamentos é determinante para o desempenho do sistema. No primeiro caso, a Arquitetura B apresenta um menor tempo de reação porque nela a função Montar Cubos reside no CLP mestre, tendo prioridade de acesso à comunicação via rede. Inversamente, a Arquitetura A é mais favorável ao segundo indicador de desempenho (tempo entre *elevador IN* e *Sinal de Fim de Ciclo*) pois nesse caso Processar Código reside no equipamento mestre.

5. CONCLUSÕES

As RdP Coloridas e Temporizadas fornecem um meio poderoso e abrangente para modelar sistemas dinâmicos. Aliadas à metodologia apresentada, torna-se possível abordar o projeto de uma arquitetura de comando distribuído de forma sistemática e estruturada. Comparado à validação analítica ou à construção de protótipos, este processo permite, sem dúvida, que se estabeleçam diretrizes de projeto a um custo baixo e sem o comprometimento das fases posteriores do ciclo de vida do sistema.

REFERÊNCIAS

- Design/CPN Tutorial for X-Windows, Meta Software Corporation, Cambridge, MA.
- IEC, 1988, Preparation of Function Charts for Control Systems, International Electrotechnical Commission, International Standard, 848, Genève.
- Jaulent P., Génie Logiciel: Les Méthodes, Armand Colin, Paris, 1990.
- Jensen K. & Rozenberg G., 1991, High-Level Petri nets. Theory and Application, Springer-Verlag, Berlin.
- Meunier, P & Denis, B., 1997, Validation du Comportement Dynamique des Architectures de Conduite de Systèmes de Production par Simulation, Modélisation et Simulation des Systèmes de Production et de Logistique, Junho 5-6, Rouen, França, pp. 229-238.
- Reisig W., 1985, Petri Nets: an Introduction, Springer-Verlag, New York.
- Rosário, J. M. *et al*, 1996, Platform for Teaching and Research on Industrial Automation, Proceedings of the Third International Electronic Engineering Conference, Agosto, Trujillo, Peru.
- Ross, D. & Schoman, K., 1977, Structured Analysis for Requirements Definition, IEEE Transactions on Software Engineering, vol. SE-3, n. 1, pp. 6-15.
- Verdin, L., 1991, De la Spécification à l'Exploitation: Le Cycle de Vie des SAP, Actes du GAMI: Génie Automatique et Production Industrielle, vol. 2, Março, Saint-Ouen, France.
- Yourdon E., 1989, Modern Structured Analysis, Prentice-Hall.

PERFORMANCE EVALUATION OF CONTROL ARCHITECTURES

Abstract. *The design phase of Automatic Manufacturing Systems comprises many different tasks and among them stands the design of a control architecture. Actually for highly distributed control systems the choice for a specific architecture has great impact on the overall system's life cycle. In such cases the designer has to propose early enough in the system's design phase a control architecture which must fulfill functional requirements such as response time. This work presents a framework for validating distributed control architectures by modelling the system's distributed features with tools from Structured Analysis and further 'translating' its dynamic behavior into Timed Colored Petri Nets models for simulation. A case study is provided as an application in a didactic automatic platform.*

Keywords: *Distributed control, Simulation, Timed Colored Petri Nets, Response time*