

# A Concepção de Aplicações e o Dimensionamento de Redes Barramento de Campo: Uma Visão Orientada à Engenharia Concorrente

Pedro F. Rosa\*      Francisco Vasques†      Robert Valette

Laboratoire d'Automatique et d'Analyse des Systèmes:  
7. Av. du Colonel Roche 31077 Toulouse Cedex, France  
Tel: (33).61.33.62.63; Fax: (33) 61.33.64.11  
(rosa:vasques:robert)@laas.fr

## Abstract

*This paper presents some properties which must be verified by the project of distributed application in order to make good use of fieldbuses. In general, application and network projects are developed by different persons, such that a good interaction must be reached. With the aim of doing that, Concurrent Engineering have been more and more used.*

*A specification methodology (Internal Behavioral and Conceptual Configuration Models) of real-time distributed application is presented, based on both Concurrent Engineering and Network Transparency Concept, which offers all necessary information to the network project. Three standards (Profibus, FIP, IEC 65-C) have been analysed, from real-time requirements point of view, in order to allow viability test, by taking into account the specification information.*

## Sumário

*Este artigo apresenta algumas propriedades que devem ser garantidas pelo projeto de uma aplicação distribuída para que as redes barramento de campo sejam bem utilizadas. Em geral, os projetos da aplicação e da rede são feitos por pessoas diferentes, sendo que uma boa interação deve ser alcançada. Com esta finalidade, a Engenharia Concorrente tem sido cada vez mais utilizada.*

*Uma metodologia de especificação (Modelos Comportamental Interno e Conceitual de Configuração) de uma aplicação distribuída tempo-real é apresentada, baseada na Engenharia Concorrente e no Conceito de Transparência da Rede, cuja oferece todas as informações necessárias para o projeto da rede. Três padrões (Profibus, FIP e IEC 65-C) são analisados, do ponto de vista dos requisitos tempo-real, para que a viabilidade seja testada, levando em consideração a especificação.*

**Keywords:** FMS, CIM, hard/soft real-time systems, network transparency concept, fieldbus, distributed real-time database, concurrent engineering

## Introdução

Na<sup>1</sup> automação de processos industriais, nota-se uma tendência de embutir tarefas cada vez mais complexas no nível do chão de fábrica, sendo que pelo menos duas razões podem justificar esta tendência:

\*Pedro F. Rosa é associado ao LSI-EPUSP da Universidade de São Paulo e pode ser contactado por e-mail frosi@lsi.usp.br

†Francisco Vasques é associado à FEUP, Universidade do Porto, Portugal e pode ser contactado por e-mail vasques@fe.up.pt

<sup>1</sup>Este trabalho foi parcialmente suportado pelo CNPq (processo N°260213/92-6) e pela JNICT (programa N°BD 1278/91)

- A leitura (escrita) de um número maior de sensores (atuadores) é requerida para se atingir um melhor controle de instalações complexas onde produtividade, qualidade e segurança são mais e mais desejadas;
- A supervisão destas instalações complexas tendem a utilizar o conceito de "automação integrada" (*integrated automation*), isto é, a abrangência tradicional é alargada - além dos algoritmos de controle tempo real também são consideradas novas funcionalidades tais como manutenção, gerenciamento e facilidades para detecção de falhas.

Se de um lado o controle e a monitoração exigem muito mais informações, do outro, do ponto de vista da rede, estas informações devem circular obedecendo certas restrições temporais, para que as "imagens" do processo sendo controlado representem, a menos de um erro limitado e conhecido a priori, o estado real do ambiente. Resumindo existe um compromisso entre a concepção do sistema de controle e o projeto da rede de comunicação.

Neste ponto cabe ressaltar dois conceitos que foram utilizados neste trabalho, o da transparência da rede em sistemas baseados em redes barramento de campo (*fieldbuses*) [Ros 94] e o da Engenharia Concorrente [Kus 93] descritos a seguir.

#### *Transparência da Rede*

Neste nível da fábrica, as aplicações individuais sendo executadas localmente não podem ser consideradas isoladas umas das outras. Elas são, em geral, parte de uma aplicação distribuída global, sendo que as suas interações (das aplicações individuais) acontecem através da rede.

A base de dados tempo-real distribuída (*RTDD - Real-Time Distributed Database*) oferece o enlace lógico, entre as aplicações individuais, o que garantirá as interações e a evolução do sistema.

A idéia fundamental é baseada no modelo "dirigido pelo estado" (*state driven*), cujo modelo permite maior previsibilidade do tráfego na rede. Neste nível, em função de sua particularidade, adotamos dois sub-modelos, um para o tráfego aperiódico e outro para o tráfego periódico.

- A parte aperiódica é tratada de acordo com o modelo "dirigido pelo produtor" (*Producer Driven*) [Aud 92], que implementa um serviço do tipo cliente-servidor, mas que permite múltiplos consumidores para um valor produzido;
- A parte periódica é tratada de acordo com o modelo "dirigido pela rede" (*Network driven*) [Ros 94], que implementa um serviço do tipo "um Produtor  $\Rightarrow$   $n$  Consumidores", onde o produtor atualiza a base de dados tempo-real local e a rede é responsável por atualizar todas as bases de dados tempo-real remotas (imagens) que utilizam a variável produzida.

A utilização da base de dados distribuída como o meio para a interação entre os Processos de Aplicação [Iso 86] é um dos postulados de base do conceito de transparência do meio de interconexão [Ros 94].

#### *Engenharia Concorrente*

A Engenharia Concorrente é geralmente reconhecida como a prática de considerar, e incorporar, diversas características do ciclo de vida do sistema desde as fases iniciais do projeto. Estas características incluem, além das funções primárias, aspectos tais como estética, manufaturabilidade (*manufacturability*), montagem, reciclagem, etc [Kus 93].

A Engenharia Concorrente é um grande desafio gerencial e organizacional: a comunicação entre as diferentes partes envolvidas, já nas revisões iniciais pelo grupo de desenvolvimento do projeto, pode ser complicada enormemente.

No que tange a parte de interconexão de dispositivos de manufatura em ambientes industriais, um problema complexo de projeto é em geral decomposto em problemas menores, sendo a cada um atribuído um processo de engenharia [Kam 92].

Neste ponto destacaremos duas funções sobre as quais aplicaremos a Engenharia Concorrente (ver figura 1):

- O processo de engenharia que fará o levantamento de todas as funções elementares de controle e monitoração de baixo nível, incluindo seus requisitos;

- O projeto do meio de comunicação que garantirá as interações entre as funções elementares, assegurando que seus requisitos serão respeitados.

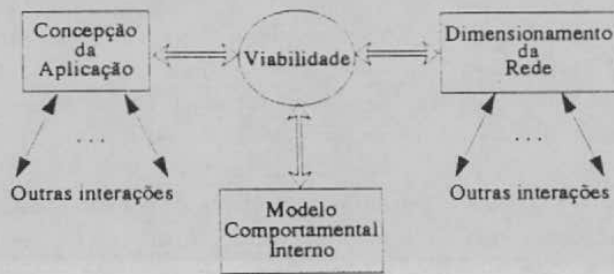


Figura 1: Concepção Sistema x Dimensionamento Rede

Neste artigo nós apresentaremos uma metodologia, baseada na Engenharia Concorrente e no Conceito de Transparência da Rede, para extrair os requisitos de comunicação que estão embutidos na concepção de sistemas tempo-real complexos, sendo que o objetivo principal será investigar as interações entre a concepção de aplicações distribuídas e o projeto/configuração de redes do chão de fábrica.

Na seção 1 descreveremos aspectos da concepção do sistema de tempo-real que serão importantes para o projeto da rede de comunicação. Na seção 2 abordaremos as características relevantes para o tempo-real de diversas redes. Na seção 3 introduziremos os cálculos gerais de viabilidade, levando em consideração os aspectos descritos nas seções anteriores. Na seção 4 mostraremos como ocorrem as interações entre as duas partes, finalizando com a conclusão.

## 1 Concepção da Aplicação e o Projeto da Rede

Nesta seção dois pontos de vista serão levados em consideração a fim de definir com clareza as interações entre os grupos de sistema e da configuração da rede.

Primeiro será definida uma metodologia para o levantamento dos objetos que serão importantes durante o ciclo de vida do sistema, incluindo a configuração da rede. Neste ponto descrever-se-á como será a interface do ponto de vista da concepção do sistema. Esta fase chamaremos de Modelo Comportamental Interno.

Em seguida, definir-se-á como estes objetos (na realidade seus atributos) serão úteis na configuração da rede; mais precisamente como será a interface da parte de dimensionamento da rede. Esta fase chamaremos de Modelo Conceitual de Configuração.

### 1.1 Modelo Comportamental Interno

O Modelo Comportamental Interno consiste em uma descrição "formal" dos algoritmos, dos fluxos funcionais entre os algoritmos, das regras de escalonamento, etc. Neste nível não há qualquer preocupação com o tipo de sistema de comunicação utilizado.

#### 1.1.1 Arquitetura Global

O modelo comportamental interno tem por objetivo descrever o comportamento da aplicação do usuário do modo como ela é vista pelo sistema de comunicação. O modelamento será baseado nos objetos definidos em [Iso 86, Iso 88]; sendo que neste trabalho eles têm os seguintes significados [Ros 94]:

- A aplicação global é dividida em atividades básicas globais de controle e/ou monitoração, sendo que cada uma delas será chamada Tarefa de Processamento de Informação (IPT - *Information Processing Tasks*)[Iso 86, Iso 88];

- Cada IPT é executada utilizando um conjunto de recursos distribuídos, os Dispositivos Virtuais de Manufatura (VMD - *Virtual Manufacturing Devices*);
- Em cada nó do sistema distribuído, o conjunto de recursos disponíveis (*hardware, software*, por exemplo) é chamado de Sistema Aberto Real (ROS - *Real Open System*);
- O ROS é subdividido em um ou mais Processos de Aplicação (AP - *Application Process*), em cada um existindo zero, um ou mais VMDs;
- Cada VMD contém um conjunto de um ou mais Domínios, um arquivo virtual (VF - *Virtual Filestore*), e um conjunto de zero, uma ou mais Entidades de Aplicação (AE - *Application Entity*), tal que cada AE tem sua Interface do Programa de Aplicação (API - *Application Program Interface*). Todas as interações com o VF é feita através da API.
- Cada Domínio<sup>2</sup>, de fato, representará uma parte de uma IPT dentro da aplicação distribuída global; algumas vezes será referenciado como "algoritmo elementar".

O modelo comportamental interno pode ser estabelecido a partir de duas características fundamentais:

- Decomposição do processo em Domínios

A aplicação a ser automatizada deve ser decomposta em domínios (algoritmos elementares), cujos trocam informações entre si. O objeto Domínio corresponde à entidade básica manipulada por este modelo.

Esta decomposição deve definir entidades atômicas que executam determinadas funcionalidades, levando em consideração os aspectos temporais destas funcionalidades, no que tange os tempos de respostas dos processos.

Uma entidade atômica é uma função de base (domínio) de uma aplicação distribuída (IPT), existente dentro de um único equipamento (VMD). A execução de tais domínios é funcionalmente ininterruptível. Um domínio é executado em um dos dois modos:

- Periodicamente, onde além do período deve ainda ser definido o tempo de execução do domínio. Cabe ressaltar que este é o caso típico do modelo dirigido pela rede onde o período será considerado fixo;
- Aperiodicamente, onde é necessário definir o evento de entrada, e, eventualmente o evento de saída. Neste caso a única restrição imposta é que entre duas ocorrências de um mesmo evento exista um tempo mínimo. Aqui utilizaremos o modelo dirigido pelo produtor, sendo que mensagens de alta prioridade poderão ser geradas.

- Declaração do Fluxo de Informação

As funcionalidades do sistema de produção suportadas pelos VMDs estabelecem um fluxo de informação, sujeitas às restrições impostas pela arquitetura distribuída, sendo que estes fluxos de informações devem levar em consideração os dados do processo bem como as ordens de sincronização das IPTs.

Cada um dos domínios, do modelo comportamental interno, é descrito por um objeto que descreve certas características estáticas (nome, período, tempo de resposta, etc).

Os objetos trocados entre os domínios correspondem às variáveis do sistema de comunicação. Cada variável do sistema de comunicação tem uma correspondência biunívoca com um objeto (variável) da camada de aplicação [Ros 92].

É importante frisar que nós suporemos que algumas operações de reagrupamento (*structure*) ou de particionamento (*scatter*) de objetos da aplicação distribuída, em relação aos objetos do sistema de comunicação, foram efetuadas em uma fase anterior à geração da aplicação. Esta temática, nos dias atuais, é abordada pelo chamado "Trabalho Concorrente Suportado por Computador" (CSCW

<sup>2</sup>O modelo [Iso 88] define que um programa é constituído de um ou mais objetos domínios, sendo que neste trabalho consideraremos que um domínio significará um programa em um VMD



- *Computer Supported Concurrent Work*) [Kus 93]. Do ponto de vista deste trabalho esta seria a fase do "Modelo Comportamental Externo".

O conjunto de objetos domínios definidos em acordo a este modelo constitui a parte "estática" do modelo comportamental interno. A parte dinâmica é definida pelas expressões das regras de escalonamento da execução dos domínios [Rod 88].

### 1.1.2 Aspecto Temporal da Execução de Domínios

Basicamente, no âmbito de controle de processos, um domínio (algoritmo) tem três fases distintas em sua execução: uma fase de aquisição de dados, uma fase de processamento dos dados (o algoritmo), e uma fase de externar o(s) resultado(s).

- Fase de aquisição dos dados (IN)  
Esta fase se caracteriza por adquirir (ler) os dados a partir do VF. O início desta fase pode ocorrer periódica ou aperiodicamente (em função de um evento de "entrada"). Em ambos os casos é imperativo que todos os dados adquiridos pelo domínio na fase "IN" sejam temporalmente consistentes [Sta 88, Ram 93, Ros 94], i.e., nenhum objeto será atualizado durante toda esta fase ( $T_{in}$ ).
- Fase de Processamento (PR)  
Esta fase corresponde à execução do domínio propriamente dito. Durante esta fase supõe-se que nenhum objeto será adquirido (atualizado) do (no) VF pelo domínio. Esta fase será caracterizada pela duração  $T_{pr}$ .
- Fase de Resultados (OUT)  
Esta fase se caracteriza por atualizar o VF com o(s) resultado(s) da fase de processamento. O fim desta fase pode ser, eventualmente, sincronizado por um evento de "saída". Quando a fase de aquisição dos dados é síncrona com um evento de entrada, é necessário que todos os objetos atualizados por esta fase sejam temporalmente consistentes, i.e., que os valores utilizados na entrada sejam ainda "visíveis" (válidos) no instante da atualização. Nenhum objeto pode ser difundido durante esta fase ( $T_{out}$ ).

No caso de execuções periódicas, uma outra duração faz-se necessária que é o comprimento do período ( $T_{per}$ ). Este período permite-nos levar em consideração as necessidades da IPT em termos da execução de cada domínio, e, além disso considerar os tempos de respostas do processo físico controlado. Um consideração a mais que se faz necessária é que  $T_{per} \gg T_{in} + T_{pr} + T_{out}$  (ver figura 2).



Figura 2: Fases da execução do Domínio

Em função da consistência temporal entre todos os objetos de entrada e saída, e da decomposição das fases de execução do domínio, poder-se-á deduzir algumas restrições (*constraints*) levando-se em consideração o "período".

Estas restrições basear-se-ão na exclusão mútua que existe entre algumas fases da execução do domínio e as operações de leitura ou atualização dos objetos no VF.

Em princípio podemos destacar as seguintes regras:

- Os objetos de entrada  $E = \{e_i\}$ ,  $i = 1, 2, 3, \dots$ , contidos no VF, não podem ser atualizados durante a fase IN do domínio;

- Os objetos de entrada  $E = \{e_i\}$ , devem ser atualizados entre o final da fase IN do período  $p$  e o início da fase IN do período  $p + 1$ ;
- Os objetos de saída  $S = \{s_k\}$ ,  $k = 1, 2, 3, \dots$ , contidos no VF, não podem ser difundidos durante a fase OUT na execução do domínio;
- Os objetos de saída  $S = \{s_k\}$  podem ser difundidos a partir do VF entre o fim da fase OUT do período  $p$  e o início da fase OUT do período  $p + 1$ ;

Idealmente, os objetos de entrada seriam atualizados imediatamente antes de iniciar a fase IN, e os objetos de saída seriam difundidos imediatamente após terminar a fase OUT. A figura 3 a seguir mostra os instantes válidos de atualização e leitura do VF.



Figura 3: Entrada e Saída: Instantes válidos

No nível do modelo comportamental interno não podemos quantificar diretamente os tempos ( $T_{in}$ ,  $T_{pr}$  e  $T_{out}$ ), para cada domínio, isto porque estes tempos, por serem tempos de execução, serão diferentes em função do dispositivo físico cujo será representado pelo VMD.

É importante notar que neste nível, o usuário deverá preocupar-se principalmente com o aspecto da consistência temporal relativa [Ros 94], sendo por isto que o modelo leva em consideração o fluxo de dados entre os domínios.

O modelo comportamental interno oferece uma visão, orientada a domínios, da IPT sendo especificada. Como já foi dito cada um dos domínios será representado por um objeto que apresentará suas características visíveis do ponto de vista API.

Objeto: Domínio

```

Atr-chave := <Nome do Objeto Domínio> : <Nome do VMD>
Atributos := Período : <Valor de T_per>
           Input : <Valor de T_in>
           Processamento : <Valor de T_pr>
           Output : <Valor de T_out>
Entradas :
  <Nomes dos Objetos Variável {E_i}> :
  <Tipos dos Objetos Variável {E_i}>
Saídas :
  <Nomes dos Objetos Variável {E_i}> :
  <Tipos dos Objetos Variável {E_i}>
Eventos :
  Entrada : <Nome do Evento> | NENHUM
  Saída : <Nome do Evento> | NENHUM

```

Convém ressaltar que quando o conteúdo do atributo "Evento Entrada" é "NENHUM", o conteúdo do atributo "Período" é relevante, sendo, portanto, obrigatório. No caso do valor do atributo "Evento Entrada" ser diferente de "NENHUM", o conteúdo do atributo "Período" definirá o tempo mínimo entre duas ocorrências consecutivas do eventos referenciado.

## 1.2 Modelo Conceitual de Configuração

O modelo conceitual de configuração é um modelo refinado em relação ao modelo comportamental interno, sendo que aqui leva-se em consideração certos aspectos do modelo de serviço que suportará os fluxos funcionais definidos.

Até este momento apresentamos uma visão da aplicação distribuída (IPT) orientada a algoritmos elementares (domínios) implantados dentro dos equipamentos (VMD) do sistema de controle.

O modelo conceitual de configuração oferece uma visão menos agregada da IPT onde a orientação volta-se às variáveis locais (possivelmente "imagem"). Neste trabalho "imagem" de variável significa a representação local de uma variável que foi produzida, por um equipamento do sistema de controle, em algum ponto remoto dentro do sistema de interconexão.

É importante lembrar que uma variável da camada de aplicação (MMS ou MPS) poderá ter um conjunto de imagens (remotas), lembrando ainda que existe um produtor e um ou mais consumidores.

Todas as variáveis da camada de aplicação (imagens ou não) são distribuídas através do conjunto de equipamentos do sistema de controle que estão relacionados, por sua vez, aos Processos de Aplicação.

A construção do modelo conceitual de configuração é feita em duas partes. A parte estática consiste em interpretar algumas características dos domínios, i.e.,

- A partir das variáveis de entrada dos domínios (aquelas adquiridas na fase IN), deduzimos as variáveis locais consumidas do VF;
- A partir das variáveis de saída dos domínios (aquelas atualizadas na fase OUT), deduzimos as variáveis locais produzidas do VF;
- A partir da existência de um evento de entrada, deduzimos uma variável local de sincronização consumida do VF;
- A partir da existência de um evento de saída, deduzimos uma variável local de sincronização produzida do VF.

A parte dinâmica faz a formalização numérica das restrições temporais globais do escalonamento da execução dos domínios. Estas restrições são feitas no nível das variáveis locais (possivelmente imagens) do sistema de comunicação, em diferentes estágios:

- Quantificação das regras ligadas aos instantes característicos de cada domínio;
- Formalização, através das restrições numéricas impostas às regras de acesso/difusão às variáveis de entrada/saída durante as diferentes fases da execução de um domínio.
- Formalização quantitativa (numérica) das restrições globais de escalonamento da execução de domínios entre as quais restrições de escalonamento absoluto e relativo.

### 1.2.1 Parte Estática

Esta parte resulta diretamente da interpretação de objetos definidos dentro da parte estática do modelo comportamental interno. Os objetos construídos a partir desta interpretação definem as variáveis locais no VF. Estes objetos têm a seguinte forma:

```

Objeto: Variável local no VF
Atr-chave : <Nome do Objeto Variável>
Atr-chave : <Nome da Imagem do Objeto Variável>
Atributos : <Produzido | Consumido>
            <Normal | Sincronização>
            <Tipo do Objeto Variável>
            Período : <Valor de Tper>
            <Prioridade>
  
```

Os atributos dos objetos que compõem a parte estática do modelo conceitual de configuração têm uma correspondência direta com as variáveis locais, produzidas ou consumidas, do VF.

Ressaltamos que as variáveis normais de processo declaradas dentro dos domínios correspondem às variáveis com atributo "Normal" no VF, enquanto que aquelas com atributo "Sincronização" correspondem às variáveis associadas a eventos.

### 1.2.2 Parte Dinâmica

A construção desta parte do modelo conceitual de configuração deve associar às variáveis do sistema de comunicação as restrições de acesso ao VF, devidas por um lado às fases IN/OUT de execução do domínio (incluindo as restrições visando consistência temporal relativa), e por outro lado as relações de escalonamento encadeando as execuções dos domínios (regras advindas do modelo comportamental interno).

As notações para o formalismo utilizado para exprimir estas limitações de acesso aos objetos variáveis locais presentes no VF e de execução dos domínios é o seguinte:

- $Ts_{in}^i(D_x)$ : instante inicial da fase "IN" na execução do domínio  $D_x$  no período  $i$ ;
- $Ts_{pr}^i(D_x)$ : instante inicial da fase "Processamento" na execução do domínio  $D_x$  no período  $i$ ;
- $Ts_{out}^i(D_x)$ : instante inicial da fase "OUT" na execução do domínio  $D_x$  no período  $i$ ;
- $Tc_{out}^i(D_x)$ : instante final da fase "OUT" na execução do domínio  $D_x$  no período  $i$ ;
- $At^i(V_k)$ : instante de atualização do objeto variável de entrada  $V_k$  no período  $i$ ;
- $Df^i(V_k)$ : instante de difusão do resultado do objeto variável de saída  $V_k$  no período  $i$ ;
- $D_a \parallel D_b$ : as execuções em paralelo dos domínios  $D_a$  e  $D_b$ ;
- $D_a ! \{V_k\} ! D_b$ : a execução do domínio  $D_b$  é sincronizada com o fim da execução domínio  $D_a$  e, opcionalmente, com a passagem das variáveis  $V_k$ . Para o caso de não haver sincronização com variáveis utilizar-se-á a notação  $D_a ! D_b$ .

As regras de escalonamento entre as execuções dos domínios serão formalizadas como a seguir:

- i)  $D_a \parallel D_b \iff Ts_{in}^i(D_a) = Ts_{in}^i(D_b)$
- ii)  $D_a ! D_b \iff Tc_{out}^i(D_a) \leq Ts_{in}^i(D_b) \leq Ts_{out}^{i+1}(D_a)$
- iii)  $D_a ! \{V_k\} ! D_b \iff Tc_{out}^i(D_a) \leq Ts_{in}^i(D_b) \leq Ts_{out}^{i+1}(D_a)$  e  $\forall k, At^i(V_k) \in [Tc_{out}^i(D_a), Ts_{in}^i(D_b)]$
- iv)  $D_b$  após  $D_a$  min  $k_1$  max  $k_2 \iff Ts_{in}^i(D_a) + K_1 \leq Ts_{in}^i(D_b) \leq Ts_{in}^i(D_a) + K_2$
- v)  $D_b$  antes  $D_a$  min  $k_1$  max  $k_2 \iff Ts_{in}^i(D_a) + K_1 \leq Tc_{out}^i(D_b) \leq Ts_{in}^i(D_a) + K_2$

O fato de não haver atualização(difusão) nas(das) variáveis de entrada(saída)  $V_e(V_s)$  durante os intervalos  $T_{in}(T_{out})$  (fases IN/OUT) na execução de um domínio  $D_a$  é expresso por "i" ("ii"):

- i)  $Ts_{pr}^{i-1}(D_a) \leq At^i(V_e) < Ts_{in}^i(D_a)$
- ii)  $Tc_{out}^{i-1}(D_a) \leq Df^i(V_s) < Ts_{out}^i(D_a)$

Basicamente esta parte se resume à forma de um sistema de restrições:

- Sobre o posicionamento relativo dos instantes característicos da execução de um domínio;
- Sobre os instantes de atualização e difusão relacionando cada uma das variáveis locais com os eventos implementados por cada domínio.

## 2 Características Relevantes de Diversas Redes

Neste nível da fábrica, o nível do controle de processo, a rede deve oferecer um tempo de resposta da ordem de 10ms [Ull 91, Iso 86]. Estas redes têm uma arquitetura reduzida conhecida como "barramento de campo" (*fieldbus*). Este artigo versará sobre três padrões, sendo dois deles muito utilizados nos dias atuais.

Uma apresentação completa das redes Profibus, FIP e da proposta IEC 65-C está fora do escopo deste artigo. Contudo as características peculiares, para uma análise das capacidades de tempo-real, serão resumidamente colocadas nesta seção.



## 2.1 Modelamento das Mensagens

Como foi citado anteriormente, as mensagens requeridas com regularidade (*regular arrival time*) serão chamadas "periódicas" enquanto que aquelas ocasionais, mas com fortes restrições de tempo, serão chamadas "esporádicas". É importante ressaltar que as mensagens esporádicas neste modelo comportar-se-ão de modo pseudo-periódico, pois elas conterão um tempo mínimo entre duas ocorrências de uma mesma mensagem esporádica.

Frequentemente as aplicações tempo-real *hard* coexistem com aplicações não críticas, onde algum atraso é gerenciado sem problemas. Doravante chamaremos as mensagens não críticas de mensagens *background*.

Um sistema ideal transmitiria as mensagens tempo-real *hard* respeitando suas "datas de fim" (*deadline*) e garantiria um tempo médio mínimo de transferência para as mensagens *background* (por exemplo, garantiria que todas as mensagens seriam enviadas dentro do macro-ciclo de ocorrência).

O modelo especifica dois conjuntos de mensagens: um tempo-real *hard* e o outro *background* (incluindo mensagens tempo-real *soft* e mensagens não tempo-real). O conjunto referente ao tempo-real *hard* contém mensagens periódicas e esporádicas [Vas 93]. Nós assumimos que:

- Existe um conjunto  $P$  com  $p$  mensagens periódicas e um conjunto  $S$  com  $s$  mensagens esporádicas, i.e.:  
 $P = \{P_1, P_2, \dots, P_i, \dots, P_p\}$  e  $S = \{S_1, S_2, \dots, S_j, \dots, S_s\}$ ;
- $TP_i$  é o período de cada uma das mensagens  $P_i \in P, i = 1, 2, \dots, p$ ;
- $TS_j$  é o mínimo intervalo entre duas ocorrências das mensagens  $S_j \in S, j = 1, 2, \dots, s$ ;
- $dP_i$  ( $dS_j$ ) é a data de fim para cada uma das mensagens  $P_i$  ( $S_j$ ), i.e., se uma mensagem é requerida no instante  $t$  então sua data de fim é  $t + dP_i$  ( $t + dS_j$ ), onde  $dP_i < TP_i$  ( $dS_j < TS_j$ );
- $CP_i$  ( $CS_j$ ) é o comprimento de cada mensagem  $P_i$  ( $S_j$ ), sendo que aqui poder-se-ia considerar o tempo de transmissão de cada mensagem (que depende da taxa de transmissão do meio escolhido). Por motivos óbvios  $CP_i < dP_i$  ( $CS_j < dS_j$ );
- Existe uma fila para cada tipo de mensagem (periódica, esporádica e *background*).

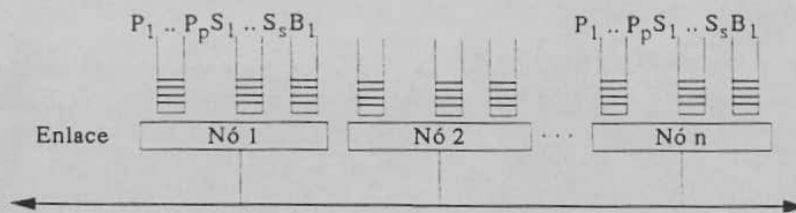


Figura 4: Distribuição das Mensagens

## 2.2 O padrão Profibus

A rede Profibus é do tipo barramento (*bus*) e tem como método de acesso ao meio (MAC) o mecanismo de passagem de ficha (*token bus passing*). A idéia básica é controlar o tempo que a ficha leva para percorrer o anel lógico. Isto é feito na iniciação da rede através da fixação do parâmetro de protocolo chamado  $T_{tr}$  (*token rotation target*). O valor deste parâmetro é definido em função do tempo desejado para uma volta da ficha mais o intervalo disponível durante o qual cada estação pode transmitir suas mensagens [Pro 91].

Na realidade o parâmetro  $T_{tr}$  é calculado levando em consideração o compromisso entre dois fatores:

- Ser bastante grande para que a janela temporal seja suficiente para a transmissão de todas as mensagens requisitadas anteriormente;

- Ser suficientemente pequeno a fim de manter a resolução do comportamento do processo dentro dos patamares definidos pelo tipo da aplicação.

O padrão Profibus sugere um valor mínimo para o parâmetro  $T_{tr}$ . Porém este valor sugerido baseia-se na taxa de transmissão média do sistema e não na data de fim de cada mensagem. Em função disto um cálculo especial [Vas 93], baseado no pior caso de funcionamento, deve ser feito para garantir que as mensagens tempo-real sejam transferidas completamente.

As mensagens esporádicas serão transmitidas com alta prioridade, as mensagens periódicas como mensagens cíclicas e as mensagens sem restrições temporais (*background*) com baixa prioridade. Deste modo cada nó transmitirá primeiro suas mensagens esporádicas, depois as mensagens periódicas e por último as mensagens *background*, se houver tempo disponível (*time window*). De qualquer modo deve-se garantir que cada nó consegue sempre transmitir suas mensagens periódicas e/ou esporádicas.

### 2.3 O padrão FIP

A rede FIP é baseada no protocolo TDMA<sup>3</sup> centralizado, onde uma entidade central, o árbitro do barramento (BA), controla todos os acessos ao meio de transmissão, permitindo assim a troca de mensagens entre as entidades comunicantes. O BA gerencia todas as trocas de mensagens a partir de uma tabela chamada tabela de escrutação. Esta tabela é organizada como um conjunto de micro-ciclos (macro-ciclo), que são percorridos indefinidamente.

O micro-ciclo  $T_{mc}$  é a menor janela de transmissão existente, i.e., no pior caso, em cada micro-ciclo é transmitida pelo menos uma mensagem. O micro-ciclo é função dos períodos dos objetos variáveis e é calculado pelo Máximo Divisor Comum em relação aos períodos:

$$T_{mc} = MDC(TP_i) \quad \forall i \in [1, p]$$

O macro-ciclo  $T_{MC}$  é a distância entre duas ocorrências sucessivas de todos os micro-ciclos, e, portanto, é função dos períodos dos objetos variáveis.  $T_{MC}$  é calculado pelo Mínimo Múltiplo Comum em relação aos períodos:

$$T_{MC} = MMC(TP_i) \quad \forall i \in [1, p]$$

O comprimento da tabela de escrutação é função direta do mínimo múltiplo comum aos períodos, o que significa dizer que dependendo dos valores dos períodos, associados aos objetos variáveis, sua geração pode ser difícil por causa da explosão combinatória.

Por exemplo, imaginemos que trabalhamos com tres variáveis  $v_1, v_2, v_3$  com períodos de valores  $TP_1 = 1, TP_2 = 2, TP_3 = 3$  a representação está na figura 5.

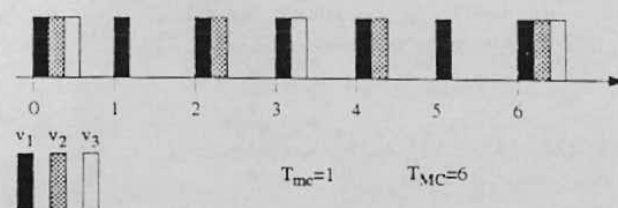


Figura 5: Relação entre macro e micro-ciclo

O árbitro do barramento (BA) executa a transferência de sequências de mensagens do seguinte modo:

- Durante a primeira fase do micro-ciclo (janela periódica), o BA autoriza a troca de uma sequência pré-definida de mensagens. Durante esta fase, estações remotas, se endereçadas, sinalizam ao BA, se for o caso, a existência de mensagens aperiódicas pendentes (varredura implícita)<sup>4</sup>;
- Na parte não utilizada por mensagens periódicas do tempo do micro-ciclo (se houver), o BA induz à transmissão de mensagens aperiódicas requisitadas pendentes (se houverem), segundo dois níveis de prioridade: primeiro as esporádicas e por fim as sem restrições temporais.

<sup>3</sup> Time Division Multiple Access

<sup>4</sup> Varredura explícita também é possível mas não será considerada

## 2.4 A proposta de padrão IEC-65C

A proposta de padrão IEC-65C, também conhecida como ISA SP-50, traz várias facilidades encontradas nos dois padrões citados anteriormente. Ela ainda não passa do plano de intenções, mas por tratar-se de uma proposta pertinente, assim como nos casos anteriores, apresentaremos alguns aspectos relevantes para a avaliação das capacidades tempo-real.

A sub-camada MAC é baseada numa solução de controle TDMA híbrida, que pode ser considerada como centralizada dentro de uma granularidade temporal grande, e como distribuída dentro de uma granularidade temporal pequena [Vas 93].

No caso da granularidade temporal grande um entidade central, o "escalador ativo do enlace" (LAS - *Link Active Scheduler*), controla o acesso ao barramento em um dos dois modos: 1) transferência única de dados entre entidades comunicantes, utilizando uma ficha para transação única (ET - *single transaction token*); 2) janela temporal a ser utilizada por uma estação distante, durante a qual a estação pode enviar mensagens (ES - *timed token with a time window*).

Para o caso da granularidade temporal pequena, a utilização do modo (2), descrito no parágrafo anterior, pode ser vista como um mecanismo de controle de acesso distribuído, pois para janelas temporais consecutivas, o controle de acesso é trocado entre diferentes estações.

A sub-camada de controle lógico do enlace (LLC) oferece um serviço especial que é a "definição de sequência" (*sequence definition*) [Iec 93]. Este serviço permite a concatenação de várias mensagens (periódicas e/ou aperiódicas) o que equivale a vários serviços de transferência de dados.

As operações de disparo são totalmente executadas sob o controle do LAS, de um modo que não é ainda completamente especificado, mas que deve manter a característica de "comportamento previsível". A consulta para envio de mensagens esporádicas pode ser explícita (o LAS consulta diretamente a estação) ou implícita (a estação remota aproveita a oportunidade de transmitir para informar ao LAS). Neste trabalho consideramos apenas o caso implícito.

Como o LAS tem conhecimento sobre todas as sequências definidas dentro da rede, ele poderá escalonar apropriadamente baseado numa tabela pré-definida (*static scheduling*) ou através de um escalonador algorítmico (*dynamic scheduling*).

Esta proposta de padrão oferece uma plataforma ideal para o tratamento do nosso conjunto de mensagens, pois uma definição de sequência inclui parâmetros tais como: período, data de fim, máximo "deslocamento do início" (*release jitter*), etc, que são parâmetros próprios de mensagens tempo-real.

As análises serão executadas para o caso de escalonamento dinâmico, sendo que, como a proposta IEC 65C não define, ainda, a parte dinâmica, trabalhar-se-á com um algoritmo que escolha as prioridades em função das datas de fim das mensagens (EDF - *earliest deadline first*) [Vas 93], acrescentando-se que não haja interrupção (*non preemptive EDF*).

## 3 Cálculos Gerais de Viabilidade

No projeto de sistemas distribuídos tempo-real, os protocolos de comunicação devem apresentar um comportamento determinístico, entre as entidades comunicantes, para que os aspectos temporais das aplicações (*deadline*) sejam respeitados.

Se satisfazer a data de fim de uma mensagem é crítico para o sistema, i.e., o não cumprimento da data de fim pode acarretar consequências desastrosas, então a mensagem é considerada "mensagem tempo-real *hard*". Se a data de fim para uma mensagem é desejável mas o não cumprimento ocasional pode ser tolerado então ela é chamada "mensagem tempo-real *soft*".

### 3.1 Cálculos particulares a cada rede

O fator de utilização do recurso de comunicação para as mensagens periódicas e esporádicas é calculado respectivamente por:

$$fu_p = \sum_{i=1}^p \frac{CP_i}{TP_i} \quad \text{e} \quad fu_s = \sum_{j=1}^s \frac{CS_j}{TS_j}$$

Outros parâmetros relevantes neste nível, e que são extremamente dependentes do protocolo,



são:

- $\delta$  = sobretaxa (*overhead*) do protocolo: é função da quantidade de LPDUs<sup>5</sup> trocadas a fim de gerenciar a rede. Por exemplo, varredura (*polling*), reconhecimento, passagem de ficha, etc;
- $\phi$  = sobretaxa sobre os dados da mensagem: baseia-se na quantidade de PDUs que serão trocadas a fim de enviar uma SDU<sup>6</sup>;
- $\sigma$  = intervalo não utilizado: mede o tempo decorrido entre a requisição para o envio da mensagem e o instante em que o "escalonador" é informado. Este parametro vale somente quando existe um escalonador global.

Como foi citado, existem algumas características que são intimamente ligadas a cada tipo de rede (dependentes do conjunto de protocolos implementados).

### 3.1.1 Cálculo para a Profibus

Para a rede Profibus o fator mais importante é o tempo de translação da ficha, mais precisamente, o parametro  $T_{tr}$  deve ser avaliado para o menor valor possível, mas que mantenha a previsibilidade do sistema.

Chamando-se  $L(t)$  a carga do sistema durante o intervalo  $t$ ,  $m_i$  a duração máxima da sequência de mensagens *background* no nó  $i$  durante o ciclo da ficha  $T_c$ , e  $\delta_T$  o tempo gasto para a passagem da ficha, temos [Vas 93]:

$$L(t) \leq \sum_{i=1}^p CP_i^* \left[ \frac{t}{TP_i} \right] + \sum_{j=1}^s CS_j^* \left[ \frac{t}{TS_j} \right] + \sum_{i=1}^n (m_i + \delta_T) \left[ \frac{t}{T_c} \right]$$

A expressão acima nada mais é que a soma dos fatores de utilização, no pior caso, associado às mensagens periódicas, esporádicas e de *background*, incluindo o tempo gasto pelo mecanismo de passagem da ficha. Como o mecanismo de acesso deste protocolo permite o envio de uma sequência específica de mensagens em cada passagem da ficha, para evitar o acúmulo de mensagens deve-se garantir que  $TP_i \geq t \forall i$  e  $TS_j \geq t \forall j$ .

Assumindo  $L$  como sendo a carga máxima do sistema, que seria equivalente a dizer que utiliza-se o máximo intervalo de tempo entre duas chegadas da ficha, o menor  $T_{tr}$  é:

$$T_{tr} \geq L + \max \left[ \left( \sum_{i=1}^p CP_i^* + \sum_{j=1}^s CS_j^* \right) \mid CP_i^*, CS_j^* \in \text{nok} \right] \forall k \in [1, n]$$

onde

$$L \leq \sum_{i=1}^p CP_i^* + \sum_{j=1}^s CS_j^* + \sum_{i=1}^n (m_i + \delta_T)$$

Para que não haja excesso de uma sequência de mensagens, a condição suficiente é [Vas 93]:

$$TP_i \geq L \forall i \in [1, p] \text{ e } TS_j \geq L \forall j \in [1, s]$$

No que tange o tempo de resposta, para a Profibus, o tempo de resposta para o pior caso (*wert*) é aquele em que uma sequência de mensagens é requisitada imediatamente após a passagem da ficha, isto é,  $wert = L$ .

Como o mecanismo de passagem de ficha garante um comportamento determinístico, a condição suficiente para a escalabilidade das sequências de mensagens impõe que o tempo de resposta para o pior caso seja sempre menor que a data de fim de quaisquer mensagens, ou seja

$$dP_i \geq L \forall i \in [1, p] \text{ e } dS_j \geq L \forall j \in [1, s]$$

Finalmente deve-se garantir que o fator de utilização devido às mensagens periódicas e esporádicas estejam aquém do fator de utilização máximo suportado pela rede. O fator de utilização para as mensagens é obtido de

$$fu = fu_p + fu_s = \sum_{i=1}^p \frac{CP_i}{TP_i} + \sum_{j=1}^s \frac{CS_j}{TS_j} \quad (1)$$

e alcança seu valor máximo [Vas 93] quando  $TP_i = TS_j = L$ , ou ainda

$$fu_{max} = 1 - \left( \frac{(\delta_M + \phi)(p+s) + n\delta_T + \sum_{i=1}^n m_i}{L} \right)$$

onde  $\delta_M$  e  $\delta_T$  representam respectivamente as sobretaxas de protocolo referentes a transferência de sequências de mensagens e passagem de ficha.

<sup>5</sup> Logical Link Protocol Data Unit

<sup>6</sup> Service Data Unit



### 3.1.2 Cálculo para a FIP

Em FIP, o tráfego de mensagens periódicas é completamente predefinido na configuração da tabela de escrutação, então o algoritmo de escalonamento é o reponsável por evitar o excesso de mensagens, bem como garantir "a priori" a previsibilidade e escalabilidade do sistema.

Em função de considerarmos apenas a varredura implícita de mensagens esporádicas temos que avaliar uma grandeza que influi no atendimento de tais mensagens: o intervalo morto, que significa o tempo decorrido desde a requisição de uma mensagem esporádica por uma estação  $k$  até seu atendimento pelo árbitro do barramento (BA).

O máximo intervalo de varredura da estação  $k$  ( $T_k$ ) é o tempo decorrido entre duas transferências de sequência de mensagens periódicas [Vas 93], isto é,

$$T_k = [\min(TP_i) \mid TP_i \in no_k] \quad \forall k \in [1, n] \quad (2)$$

A sequência  $S_i$  tem um intervalo morto ( $\sigma_i$ ) que é o máximo intervalo entre duas operações de varredura na estação  $k$  que inclua  $S_i$ . Ele é avaliado por

$$\forall S_i \in no_k, \sigma_i = \{T_k\} \quad \forall k \in [1, n] \quad (3)$$

Para garantir que não haverá mensagens em excesso durante um certo intervalo  $t$ , deve-se garantir que as mensagens escalonadas tem uma duração acumulada menor que  $t$

$$\sum_{i=1}^p CP_i^* \lceil \frac{t}{TP_i} \rceil + \sum_{j=1}^s CS_j^* \lceil \frac{t}{TS_j} \rceil + \delta \lceil \frac{t}{T_{MC}} \rceil \leq t$$

Para evitar excesso de mensagens na rede e de acordo com a tabela de escrutação o intervalo  $t$  deve ser restringido para algum múltiplo de  $T_{MC}$ , sendo que de acordo com [Vas 93] considerar-se-á o próprio. Isto resulta

$$\sum_{i=1}^p \frac{CP_i^*}{TP_i} T_{MC} + \sum_{j=1}^s CS_j^* \lceil \frac{T_{MC}}{TS_j} \rceil + \delta \leq T_{MC}$$

Em FIP, o tempo de resposta para o pior caso ( $wcrt$ ) no que tange as mensagens esporádicas, que são diretamente relacionadas com os espaços não utilizados pelas mensagens periódicas (não conhecido pelo usuário), pode apenas ser limitado, i.e.,

$$wcrt_{s_j} \leq T_{MC} + \sigma_j$$

A condição suficiente para que as mensagens sejam escalonadas estabelece que onde quer que a sequência de mensagens esteja, o valor retornado pela função "wcrt" deve sempre ser menor que a data de fim das mensagens individuais dentro da sequência, isto é,

a) Se  $dS_r > T_{MC}$

$$dS_r \geq \left( \sum_{i=1}^p CP_i^* \frac{T_{MC}}{TP_i} + \sum_{j=1}^s CS_j^* \lceil \frac{T_{MC}}{TS_j} \rceil + \delta \right) + \delta_r$$

b) Caso contrário ( $dS_r \leq T_{MC}$ ), a condição a) é necessária mas não suficiente. Vale a expressão

$$dS_r \geq \left( \sum_{i=1}^p CP_i^* \lceil \frac{T_{MC} - dS_r}{TP_i} \rceil + \sum_{j=1}^s CS_j^* \lceil \frac{T_{MC}}{TS_j} \rceil + \delta \right) + \delta_r$$

Esta condição b) é válida apenas para os casos onde os deadlines são menores que o macro-ciclo.

Em FIP o fator de utilização é calculado como em (1) sendo que o máximo fator de utilização pode ser obtido por

$$fu_{max} = 1 - \left( \frac{\delta_p}{T_{MC}} + \sum_{i=1}^p \frac{\phi_i + \delta_{M_i}}{TP_i} + \sum_{j=1}^s \frac{\phi_j + \delta_{M_j}}{TS_j} \right)$$

onde  $\delta_p$  e  $\delta_M$  são as sobre-taxas de protocolo devidas respectivamente ao mecanismo de varredura e à transferência de mensagens.

### 3.1.3 Cálculo para a Proposta IEC 65-C

Esta proposta traz algumas características importantes tanto da rede Profibus, assim como da rede FIP, o que permite a partir dela implementar qualquer uma das redes citadas. De qualquer modo trata-se de uma proposta que privilegia o tráfego periódico.

Neste trabalho, a análise de escalonamento para esta proposta [Vas 93] basicamente trabalha a partir de [Jef 91] que é uma primeira análise para a IEC 65-C.

A avaliação do intervalo morto, devido à previsibilidade do manipulador de mensagens periódicas, pode ser calculado para um limite superior, que é o mesmo intervalo em FIP (ver (2) e (3)).

Como já foi citado, nós trabalharemos com um pseudo-período para as mensagens esporádicas.

o que leva-nos, portanto, a trabalhar com um "pseudo-intervalo de chegada" (*pseudo inter-arrival time*) de mensagens esporádicas. Consideremos uma sequência  $S = \{S_1, S_2, \dots, S_s\}$ , de mensagens esporádicas, definida com um pseudo-intervalo mínimo de chegada ao sistema  $TS'_j = TS_j - \sigma_j$ . Isto significa dizer que  $TS'_j$  é o "período" mínimo restante para a sequência  $S_j$ , no instante em que ela é informada ao escalonador da rede. Doravante as expressões sempre referenciarão  $TS'_j$ .

A rede IEC 65-C oferece uma capacidade interessante através do serviço "Definição de Sequência" (*Sequence Definition Service*). Se várias sequências de mensagens são requisitadas simultaneamente no decorrer do tempo, i.e., existem micro-ciclos nos quais estas sequências são requisitadas, então elas podem ser agrupadas e ser requisitadas por este serviço. Vale acrescentar que mesmo mensagens esporádicas podem ser agrupadas, se elas forem requisitadas pelo mesmo evento.

Nos casos onde pode haver agrupamento, há a necessidade de definir uma nova sequência de mensagens que levem em consideração esta facilidade do serviço descrito acima. Consideremos a sequência  $P = \{P_1, \dots, P_i, \dots, P_{i+l}, \dots, P_p\}$  de mensagens periódicas onde  $P_i \dots P_{i+l}$  é um conjunto de sequências que podem ser agrupadas, doravante  $P^* = \{P_1, \dots, P_{p^*}\}$  ( $p^* \leq p$ ). De uma maneira semelhante consideremos  $S^* = \{S_1, \dots, S_{s^*}\}$  ( $s^* \leq s$ ).

Para o instante onde as sequências  $P^*$  e  $S^*$  podem ser enviadas, consideraremos uma nova sequência  $M = P^* \cup S^*$ <sup>7</sup>, organizada por ordem crescente em relação ao período. Deste modo  $M$  é um conjunto de sequências de mensagens tempo-real, agrupadas ou não, que o LAS deve escalonar dinamicamente. A cada elemento de  $M$  estão associadas sobretaxas do protocolo devidas ao mecanismo de passagem da ficha ( $\delta_T$ ) e à operação de varredura ( $\delta_P$ , existente somente nos casos esporádicos).

A condição suficiente para evitar sobrecarga do sistema de comunicação, a partir daquela definida em [Jef 91], foi adaptada para esta proposta como:

$$\sum_{i=1}^m \frac{C_i^* + \delta_T + \delta_{P_i}}{T_i} \leq 1 \quad (3)$$

$$L \geq (C_i^* + \delta_T + \delta_{P_i}) + \sum_{j=1}^{i-1} \left\{ \left\lfloor \frac{L}{T_j} \right\rfloor (C_j^* + \delta_T + \delta_{P_j}) \right\} \quad \forall i \in [1, m] \text{ e } \forall L \in [T_1, T_i] \quad (4)$$

Em outras palavras, a condição (1) garante que em regime permanente a rede não será sobrecarregada, a condição (2) estabelece que toda vez que houver escalonada a transferência de uma mensagem de baixa prioridade no instante  $t$  nenhuma requisição de mensagem de maior prioridade será feita no instante  $t^+$  dentro do intervalo  $L$  ( $L < T_i$ ).

## 4 As interações entre Concepção e Projeto

O projeto como um todo é concebido de tal maneira que as informações pertencentes a cada fase mas importantes naquelas interrelacionadas sejam passadas/recebidas transparentemente.

O modelo comportamental interno apresenta uma visão agregada voltada aos objetos Domínios (algoritmos) cujos atributos e métodos serão analisados e utilizados pelo modelo conceitual de configuração. Este último tem uma visão mais refinada voltada aos objetos Variáveis da camada de aplicação cujos atributos têm origem nos objetos Domínios do modelo comportamental interno.

O modelo conceitual de configuração interage com os cálculos gerais de viabilidade que submeterão os atributos especificados aos algoritmos relacionados à rede de chão de fábrica levando em consideração os fatores comentados na secção anterior (fatores dependentes do tipo de rede).

A partir do tratamento efetuado pelos cálculos de viabilidade ver-se-á se o conjunto de mensagens é escalonável. Em caso negativo, este teste defini claramente a relação dos micro-ciclos e mensagens problemas.

O primeiro passo então será verificar se o conjunto de mensagens problemas admitem a imposição de algum tipo de restrição, i.e., se admitem mútua exclusão e/ou alteração dos períodos mínimos de ocorrência, por exemplo. Neste ponto admite-se inclusive a especificação (remoção) de algum objeto intermediário (existente).

Para o caso onde as restrições aplicam-se a algum objeto Domínio então uma nova incursão pelo modelo comportamental interno faz-se necessária. Em todos os outros casos voltar-se-á ao modelo conceitual de configuração.

<sup>7</sup>O total de sequências em  $M$  é  $m = p^* + s^*$ , sendo que o comprimento da  $k$ -ésima sequência é  $C_k^*$

Caso não sejam restrições a serem impostas aos objetos passa-se para uma fase onde os atributos dos objetos Variáveis serão criticados. Cabe ressaltar que neste caso não apenas as mensagens problemas podem sofrer alterações. A figura 6 a seguir mostra todas as fases de análise.

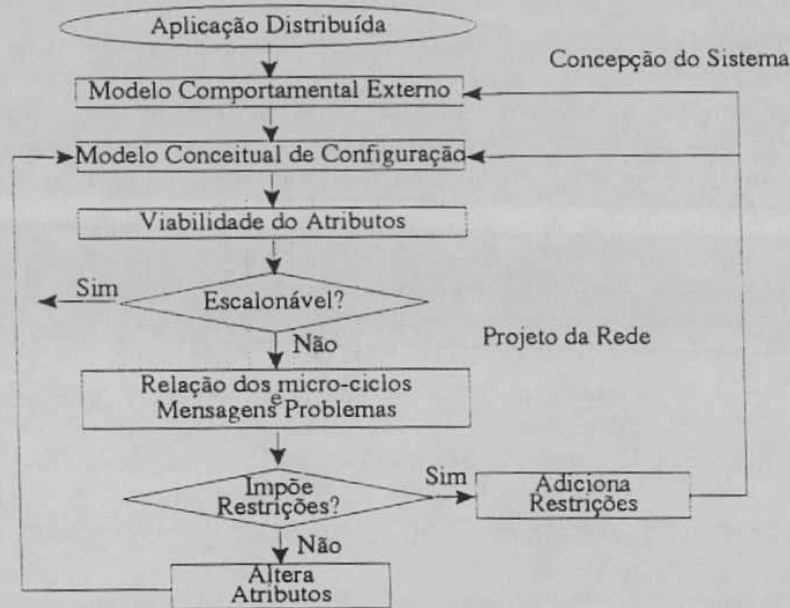


Figura 6: Concepção Sistema x Dimensionamento Rede

Quando falamos em exclusão mútua entre objetos, na realidade estamos impondo restrições temporais de modo que tais objetos serão transmitidos ou executados em intervalos disjuntos. Esta é uma capacidade importante pois ela vai influir diretamente nos aspectos de consistência dos objetos Variáveis. É importante frisar que esta é uma capacidade facilmente manipulável pela proposta IEC 65C.

Outro fator que pode ser explorado é que no caso onde a submissão dos objetos resulta como escalonável, o ambiente pode oferecer os "espaços" livres de tal modo que os projetistas da aplicação podem, por exemplo, aumentar a frequência de monitoração de um certo conjunto de variáveis do processo.

Neste ponto cabe ressaltar que uma coisa muito difícil é definir precisamente quais objetos serão passados através da interface em ambos os sentidos. Em outras palavras, as informações que circulam do grupo da aplicação para o grupo da rede, e vice-versa, devem ser muito bem delineados para que o projeto como um todo alcance os resultados esperados.

Cada um dos módulos apresentados na figura 6 acima são ainda refinados até chegar aos modelos elementares, onde cada elemento é descrito através de uma rede de Petri temporal. De todo modo este tipo de refinamento está fora do âmbito deste trabalho.

## 5 Conclusão

Este trabalho traz um grande benefício inicial que é o de poder-se fazer uma avaliação bem completa antes de qualquer implementação, i.e., é possível a priori fazer vários tipos de avaliações (por exemplo, substituir uma rede de maior desempenho, mas de custo elevado e fator de utilização alto, por duas ou mais redes de baixo desempenho, de menor custo e todas com fator de utilização baixo).

Outro aspecto que pode ser facilmente verificado por este trabalho e que é de extrema importância dentro de ambientes tempo-real é o aspecto da consistência temporal, mais especificamente a consistência temporal relativa. Enquanto a rede é a principal responsável pelas consistências espacial e temporal absoluta, os projetistas da aplicação distribuída tem que preocupar-se com a consistência temporal relativa.



A Engenharia Concorrente mostra-se atraente pois as duas fases principais de interesse apresentadas neste trabalho podem caminhar independentes uma da outra. Neste trabalho nos limitamos a analisar três tipos de redes, mas nada impede que outros tipos não considerados venham a ser incluídos. A única restrição é que as novas inserções comportem-se de acordo com a interface dentro dos padrões estabelecidos a priori pelo ambiente. A sistemática dos modelos comportamental interno e conceitual de configuração manter-se-ão com as preocupações descritas na secção 1.

## Referências

- [Aud 92] Audsley, N.; Burns, A.; Richardson, M.; Wellings, A. Data Consistency in Hard Real-Time Systems. YCS203, Dept. of Computer Science, University of York
- [Fip 91] FIP Bus for exchange of information between transmitters, actuators and programmable controllers. Norme AFNOR C-46603 and C-46605, Data Link Layer and Management Layer standards, 1991
- [Iec 93] Digital Data Communications for Measurement and Control - Field Bus for use in Industrial Control Systems, SC65C 105 Data Link service definition and SC65C 106 Data Link protocol specification draft proposals, 1993
- [Iso 86] A Reference Model for Discrete Parts Manufacturing; ISO TC184/SC5/WG1-N58, ISO
- [Iso 88] Manufacturing Message Specification - Service Definition, ISO DIS 9506/1, 1988
- [Jef 91] Jeffay, K.; Stanat, D.; Martel C. On Non Preemptive Scheduling of Periodic and Sporadic Tasks. in Proc. of IEEE Real Time Systems Symposium, 1991
- [Kam 92] Kameyama, K.-I. Real-Time Constraint Checking in the Design Process. in Concurrent Engineering: Automation, Tools, and Techniques, edited by Andrew Kusiak, John Wiley & Sons, Inc; pp 111-30. New York, 1992
- [Kus 93] Concurrent Engineering: Automation, Tools and Techniques. Edited by Andrew Kusiak, John Wiley & Sons, Inc; New York, 1993
- [Pro 91] Profibus standard (translation of German standards). DIN 19245: Process Field Bus, 1991
- [Ram 93] Ramamritham, K. Real-Time Data bases. Journal of Distributed and Parallel Databases, vol 1, pp 199-226, 1993
- [Rod 88] Rodammer, F; Whith, K. A Recent Survey of Production Scheduling. IEEE Transactions on Systems, Man and Cybernetics, nov-dec, 1988
- [Ros 92] Rosa, P. F.; Stiubiener, S. Utilização da Engenharia de Protocolos na Implementação de um Protocolo da Camada de Aplicação em Ambiente Industrial. X Simpósio Brasileiro de Redes de Computadores, Recife, May 25-27 1992
- [Ros 94] Rosa, P. F.; Vasques, F.; Valette R. The Network Transparency Concept in Fieldbus Based Distributed Systems: aceito no ISIE'94 - IEEE International Symposium on Industrial Electronics, Santiago, Chile, May 25-27 1994
- [Sta 88] Stankovic, J.; Ramamritham, K. Hard Real-Time Systems. IEEE Computer Society Press, 1988
- [Ull 91] Ulloa, G. Fieldbus Application Layer and Real-Time Distributed Systems. in Proc. of IECON'91, pp 1679-1683, 1991
- [Vas 93] Vasques, F.; Juanole G. Fieldbus MAC Mechanisms for Hard Real Time Data Communication Support. LAAS Report, dec 1993