

# AMBIENTE DE CONCEPÇÃO DE SOFTWARE PARA SISTEMAS DISTRIBUÍDOS, BASEADO EM REDE DE PETRI.

Carlos Alberto Maziero \*  
Jean-Marie Farines \*\*

## SUMÁRIO

Este artigo visa mostrar o desenvolvimento de um Ambiente de Concepção de Software para Sistemas Distribuídos, baseado na escolha da rede de Petri como formalismo de base para facilitar a integração das diversas ferramentas e simplificar seu uso.

## ABSTRACT

This paper shows the design and implementation of a Distributed Systems Software Development Environment, which uses Petri nets as formal basis to make easier the integration of several tools and to simplify their use.

\* Mestrando em Controle e Automação. Software em Tempo Real, Sistemas Distribuídos.  
\*\* Dr. Ing. INPT, France. Software em Tempo Real, Eng. de Software, Sistemas Distribuídos.

Laboratório de Controle e Microinformática "Marcos Cardoso Filho". Deptº de Eng. Elétrica, Universidade Federal de Santa Catarina. Campus Universitário da Trindade, Fpolis - SC, CEP 88049. Fone (0482) 33-9202.

## 1. Introdução.

A necessidade de reduzir custos e de aumentar a confiabilidade impõem como meta para a indústria a sua automatização, assentada na integração das diversas operações do processo industrial. Esta integração é garantida pela existência de Sistemas Informáticos Distribuídos que permitem trazer melhorias substanciais em termos do desempenho global, do compartilhamento dos recursos, da disponibilidade e da confiabilidade dos serviços fornecidos.

Aspectos tais como arquitetura de sistemas distribuídos (protocolos de comunicação, sistemas operacionais distribuídos), algorítmica distribuída (em particular para aplicações em tempo real e com requisitos de confiabilidade) e concepção de software distribuído são objeto de uma grande atividade de pesquisa e desenvolvimento a nível mundial.

Além disso, a crescente automação do processos de produção, baseada no uso de máquinas com comando numérico, controladores lógicos programáveis, robôs e outros equipamentos, tornou necessário o uso de uma filosofia de desenvolvimento adequada, ou seja, uma metodologia que seja capaz de modelar corretamente estes dispositivos e suas interações, sem dependência das tecnologias envolvidas.

Esta modelagem nem sempre é fácil. Problemas de sincronização, exclusão mútua, disponibilidade de recursos e outros, associados a mecanismos automáticos, são frequentes.

Entre as ferramentas surge a rede de Petri (rdP) como capaz de modelar fácil e corretamente paralelismo, concorrência, exclusão, partilhamento de recursos e outras situações semelhantes. O formalismo utilizado permite analisar os modelos obtidos, possibilitando detectar problemas oriundos de uma modelagem incorreta ou equivocada [1].

Outrossim, a rdP pode ser usada em várias etapas do desenvolvimento de um sistema, como a especificação, o projeto ou até mesmo a implementação, podendo ser analisada (estrutural e funcionalmente), simulada, emulada ou traduzida para uma linguagem de implementação.

Na integração da manufatura a abordagem por rdP é utilizada principalmente nos níveis de controle local e coordenação do funcionamento global da fábrica, não sendo utilizada entretanto nas funções de planejamento e gerência.

## 2. O Analisador / Simulador de Redes de Petri ARP.

### 2.1. Características Gerais.

Foi desenvolvido no Laboratório de Controle e Microinformática (LCMI-EEL-UFSC) um programa de análise de redes de Petri clássicas (ou seja, contendo somente a informação estrutural). Este programa fará parte do ambiente de desenvolvimento, permitindo ao usuário levantar resultados bastante significativos a respeito da estrutura do modelo que está estudando. O ARP conta atualmente com os seguintes módulos:

- edição de redes (editor de textos próprio).
- enumeração de marcações acessíveis, onde são levantados todos os estados acessíveis pela rede.
- procura de sub-redes do tipo máquina de estados (ou invariantes de lugar), que correspondem a componentes conservativas.
- procura de sub-redes do tipo grafo de eventos (ou invariantes de transição), que correspondem a sequências cíclicas de funcionamento.
- redução de redes, permitindo uma simplificação das redes.
- simulação, permitindo observar e decidir a respeito da evolução da rede.

Atualmente estão sendo desenvolvidos, para fazer parte do programa, os módulos de Projeção para verificação através de equivalência de linguagens e o de Cálculo Algébrico de Invariantes lugar/transição. O primeiro é particularmente interessante na análise de protocolos de comunicação, permitindo destacar do grafo de marcações da rede os pontos viáveis de sincronização paralela e gerção. Este módulo su programação poderá ser eventualmente utilizado após uma redução da rede. Já o módulo de cálculo de

invariantes permite uma maior generalidade na análise de redes, permitindo a análise de redes com peso de arco diferente de um.

Uma versão alternativa do ARP foi desenvolvida para a análise de redes de Petri com temporização. Esta versão permite efetuar a cobertura dos estados alcançáveis por este tipo de rede e sobre o grafo gerado estudar o comportamento temporal da rede (tempos de ciclo, etc). Esta versão utiliza a mesma linguagem aceita pelo ARP.

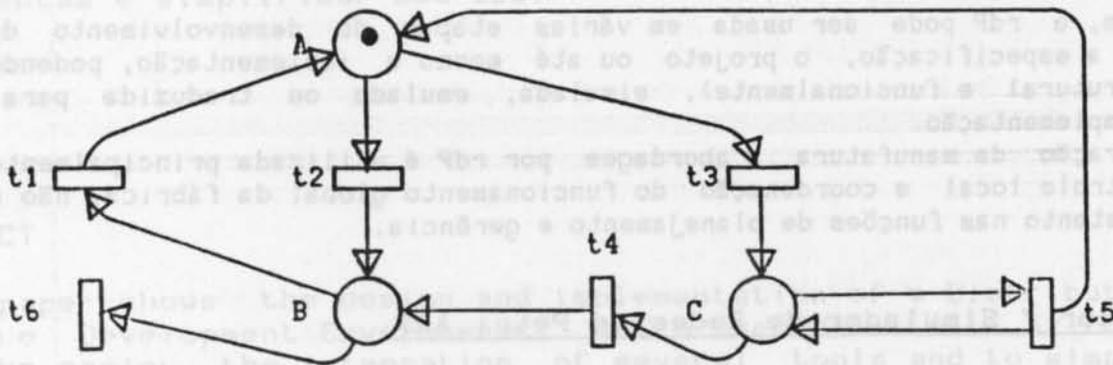
A interface homem-máquina do ARP apresenta características que facilitam a interação com o usuário, através de janelas e textos que podem ser removidos quando necessário, evitando a saturação da tela com um grande número de informações. A evolução do processamento é sempre indicada para facilitar o acompanhamento.

Uma opção bastante interessante permitida pelo programa consiste em efetuar uma análise "indireta" sobre as redes, ou seja: reduzir a rede, analisar a rede reduzida (através de qualquer método) e refletir os resultados encontrados para a rede original (não reduzida). Este processo pode acelerar em muito a análise de redes complexas, sem significativa perda de informação.

As redes são descritas por uma linguagem simples, de sintaxe semelhante à da linguagem Pascal, permitindo a identificação de lugares e transições por qualquer nome (limitado a 20 caracteres). O editor de textos permite a digitação ou alteração das redes, que são armazenadas na forma de arquivos ASCII normais, que podem ser usados por outros editores.

## 2.2. Exemplos de Resultados do programa ARP.

A seguir serão apresentados exemplos de edição e resultados tais como se apresentam nas diversas janelas do ARP. A rede usada no exemplo é mostrada a seguir:



### Edição

Rede Exemplo\_de\_Uso ; ( para exemplificar o programa )

Rodos

A : Lugar ( 1 ) ; ( marcação inicial )

B,C : Lugar ;

t1,t2,t3,

t4,t5,t6 : Transicao ;

Estrutura ( entradas e saidas das transicoes )

t1 : ( B ) , ( A ) ;

t2 : ( A ) , ( B ) ;

t3 : ( A ) , ( C ) ;

t4 : ( C ) , ( B ) ;

t5 : ( C ) , ( A,C ) ;

t6 : ( B ) , ( ) ;

Fim.

## Enumeração de Marcações

Grafo de acessibilidade da rede:

M0 : (t2: M1) (t3: M3)  
 M1 : (t1: M0) (t6: M2)  
 M2 :  
 M3 : (t4: M1) (t5: M4)  
 M4 : (t2: M5) (t3: M4) (t4: M5) (t5: M4)  
 M5 : (t1: M5) (t2: M5) (t3: M5) (t4: M5) (t5: M5) (t6: M5)

## Enumeração de Marcações

Marcacoes acessíveis pela rede: (W → infinito)

M0 : (A)  
 M1 : (B)  
 M2 : ()  
 M3 : (C)  
 M4 : (W\* A, W\* C)  
 M5 : (W\* A, W\* B, W\* C)

## Enumeração de Marcações

Propriedades verificadas:

A rede em análise não é limitada.

Lugares Nulos (M = 0) : ()  
 Lugares Binários : ()  
 Lugares k-Limitados : ()  
 Lugares Não Limitados : (todos(as))

Sequências de disparo que levam a crescimento de fichas:

Seq. (M0 → M4) : t3, t5

A rede em análise não é estritamente conservativa.

A rede em análise não é viva.

Tr. vivas : ()  
 Tr. quase-vivas : (todos(as))  
 Tr. não disparadas : ()

A rede em análise não é reiniciável.

Marcacoes reiniciáveis : M0, M1, M3  
 Transicoes limitantes : (t5, t6)  
 Marcacoes terminais:  
 M2 : ()

Sequências de disparo que levam a bloqueios:

Seq. (M0 → M2) : t2, t6

## Decomposição em Sub-Redes

Inibicoes usadas para esta decomposicao:

Transicoes Obrigatorias : ()  
 Transicoes Inibidas : ()  
 Lugares Obrigatorios : ()  
 Lugares Inibidos : ()

Não existem máquinas de estados na rede.

### Decomposição em Sub-Redes

Inibicoes usadas para esta decomposicao:

Transicoes Obrigatorias :()  
 Transicoes Inibidas :()  
 Lugares Obrigatorios :()  
 Lugares Inibidos :()

Grafos de Eventos encontrados:

GE1 P = (A, B)  
 T = (t1, t2)  
 GE2 P = (A, B, C)  
 T = (t1, t3, t4)  
 GE3 P = (A, B, C)  
 T = (t2, t5, t6)

Participaram de todas as sub-redes encontradas:

P = (A, B)  
 T = ()

Nao participaram de nenhuma sub-rede encontrada:

P = ()  
 T = ()

### 2.3. A versão Atual.

A versão atual do programa ARP conta com cerca de 340 Kbytes de código fonte em Pascal, e roda em compatíveis IBM-PC sob o sistema operacional DOS 3.0 ou superior.

Algumas limitações do ARP são:

- redes de dimensão limitada a 200 lugares e 200 transições.
- grafo de marcações acessíveis limitado a 2000 marcações.
- identificadores da linguagem com até 20 caracteres.
- no máximo 400 sub-redes podem ser encontradas.
- a procura de sub-redes só se efetua em redes que não possuam arcos de peso superior à unidade.

As limitações acima mencionadas são basicamente devido à memória disponível no equipamento, com exceção da última que é consequência do algoritmo de procura empregado.

O programa ARP tem sido intensivamente usado, tanto a nível de ensino de pós-graduação, nas disciplinas que se baseiam em redes de Petri, quanto a nível de pesquisa, onde se destaca o uso para modelagem e análise de protocolos de comunicação e células flexíveis de fabricação.

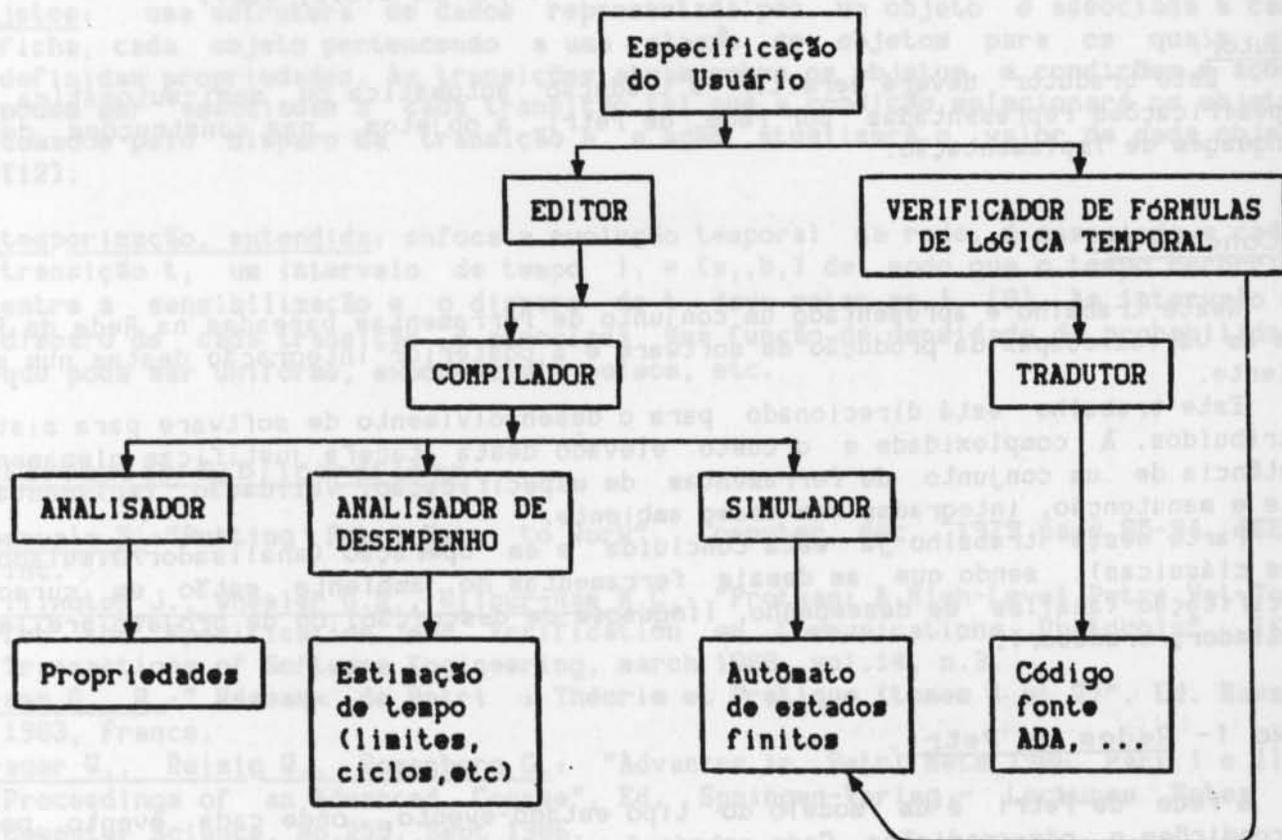
### 3. O Ambiente de Concepção de Software Baseado em Rede de Petri.

O trabalho que está sendo desenvolvido hoje no LCM-EE-UFSC visa a realização de um ambiente de desenvolvimento de software para sistemas distribuídos, baseado em rede de Petri. Este ambiente deve permitir ao usuário realizar as várias etapas da produção de software, desde a especificação inicial até o programa acabado, passando por sucessivas etapas de validação e verificação.

A classe de rede escolhida é a rede de Petri a objeto [12] que permite representar, além da estrutura de controle, a estrutura de dados na forma de objetos e classe de objetos; uma extensão para a introdução das noções de tempo e de densidade de probabilidade será necessária para a análise de desempenho.

As diversas funções do ambiente estarão integradas através da linguagem de descrição das redes. Está sendo desenvolvida uma linguagem para a descrição de redes de Petri que será utilizada por todas as ferramentas do ambiente.

O ambiente previsto deverá integrar as seguintes ferramentas, conforme apresentado na figura a seguir:



### 3.1. Descrição das ferramentas.

#### Editor:

O editor suportará uma linguagem de descrição dos modelos de rede de Petri a objeto, na forma de declarações por regras de produção; este editor deverá permitir a construção de representações por refinamentos sucessivos e a introdução de blocos pré-existentes.

Numa versão posterior, é previsto um editor gráfico, com as mesmas características.

#### Analisador:

O analisador permitirá obter as propriedades gerais (limitação, vivacidade, reinicialização, etc.) e específicas (conservatividade, repetição, etc.) do modelo. Para tal, ele deverá propiciar a construção do grafo de marcações (autômato de estados finitos), a procura dos invariantes de lugar e de transição (utilizando, no caso da rede de Petri a objeto, a projeção sobre uma variável, transformando os coeficientes variáveis da matriz de incidência em constantes), a redução da rede e/ou do grafo de marcações acessíveis, a análise de linguagens aceitas pelo autômato de estados finitos.

#### Simulador:

Este simulador deverá possibilitar a interação com o usuário através de uma evolução passo-a-passo e de uma opção de retorno ("back-tracking"); uma opção automática permitindo análises estatísticas deverá ser também disponível. Uma facilidade oferecida pelo ambiente deverá ser a geração de "cenários" que facilitará a depuração do modelo.

Analizador de desempenho:

A partir de uma representação por rede de Petri com temporização estendida (que associa a cada ocorrência de eventos um intervalo de tempo com uma densidade de probabilidade associada), esta ferramenta permitirá construir um grafo de estados com probabilidades associadas; da análise deste grafo, poderá se concluir a respeito do desempenho do sistema modelado (tempos médios, tempos de ciclo, etc).

Tradutor:

Este tradutor deverá permitir a tradução automática ou semi-automática das especificações representadas por rede de Petri a objetos, nas construções de uma linguagem de implementação.

4. Conclusão.

Neste trabalho é apresentado um conjunto de ferramentas baseadas na Rede de Petri para as várias etapas da produção de software e a posterior integração destas num mesmo ambiente.

Este trabalho está direcionado para o desenvolvimento de software para sistemas distribuídos. A complexidade e o custo elevado desta tarefa justificam plenamente a existência de um conjunto de ferramentas de especificação, validação, implementação, teste e manutenção, integradas num mesmo ambiente.

Parte deste trabalho já está concluída e em operação (analizador/simulador de redes clássicas), sendo que as demais ferramentas do ambiente estão em curso de especificação (análise de desempenho, linguagem de descrição) ou de projeto preliminar (simulador, tradutor).

Anexo I- Redes de Petri.

A rede de Petri é um modelo do tipo estado-evento, onde cada evento possui pré-condições e pós-condições. Cada estado é caracterizado pela distribuição de marcas (fichas) na rede. Formalmente, uma rdP é formada por uma quintupla  $(P, T, I, O, M_0)$ , onde:

$P$  : conjunto de lugares  $(p_1, p_2, \dots, p_m)$ .

$T$  : conjunto de transições  $(t_1, t_2, \dots, t_n)$ .

$I$  : função de entrada das transições (matriz  $n \times m$ ), onde  $I_{ij}$  indica a relação entre a  $i$ -ésima transição e o  $j$ -ésimo lugar).

$O$  : função saída das transições (matriz  $n \times m$ , similar a  $I$ ).

$M_0$  : vetor marcação inicial ( $m$  campos, cada um indicando o número de fichas no lugar respectivo da rede).

Deve-se observar que  $P \cap T = \emptyset$ , e os campos de  $I$ ,  $O$  e  $M_0$  são inteiros não negativos [3,10].

A rede de Petri clássica funciona baseada no movimento das fichas (marcas) dentro da mesma, causado pelo disparo das transições. O disparo das transições sensibilizadas provoca a transferência das fichas dos lugares de entrada para os de saída, alterando portanto o estado da rede. Uma transição  $t_i$  está sensibilizada se a relação

$$M(p_j) \geq I(t_i, p_j) \quad \forall p_j \in P$$

é verificada. O disparo da transição  $t_i$ , a partir da marcação  $M_k$  gera uma nova marcação  $M_{k+1}$  descrita por:

$$M_{k+1}(p_j) = M_k(p_j) + O(t_i, p_j) - I(t_i, p_j) \quad \forall p_j \in P.$$

## Anexo II- Extensões utilizadas.

Existem diversas extensões das rdP. Cada uma dessas extensões enfoca uma determinada característica do sistema, aumentando assim o poder de expressão do modelo. As extensões com as quais trabalhará o ambiente descrito serão as seguintes:

- a objetos: uma estrutura de dados representada por um objeto é associada a cada ficha, cada objeto pertencendo a uma classe de objetos para os quais são definidas propriedades. As transições atuam sobre os objetos, e condições e ações podem ser associadas a cada transição tal que a condição selecionará os objetos tomados pelo disparo da transição e a ação atualizará o valor de cada objeto [12].
- com temporização, extendida: enfoca a evolução temporal da rede. É associado a cada transição  $t_i$ , um intervalo de tempo  $I_i = [a_i, b_i]$  de modo que o tempo decorrido entre a sensibilização e o disparo de  $t_i$  deve estar em  $I_i$  [9]. Ao intervalo de disparo de cada transição é associada uma função de densidade de probabilidade que pode ser uniforme, exponencial, pulsos, etc.

## 5. Referências Bibliográficas:

- 1 - Agerwala T.: "Putting Petri Nets to Work". Computer, dec. 1979 pags 85-94. IEEE, Inc.
- 2 - Billington J., Wheeler G.R., Wilbur-Ham M.C.: "Protean: A High-Level Petri Net Tool for the Specification and Verification on Communications Protocols". IEEE Transactions of Software Engineering, march 1988, vol.14, n.3.
- 3 - Brans G. W.: "Réseaux de Petri : Théorie et Pratique (tomes 1 et 2)". Ed. Masson 1983, France.
- 4 - Brauer W., Reisig W., Rozenberg G.: "Advances in Petri Nets 1986, Part I e II - Proceedings of an Advanced Course", Ed. Springer-Verlag - Lectures Notes in Computer Science, No.255, Sept 1986.
- 5 - Courvoisier M., Valette R.: "Systemes de Commande en Temps Reel. Description, Analyse et Realisation". Editions SCM, 1980. Paris, France.
- 6 - Diaz M.: "Modelling and Analysis of Communication and Cooperation Protocols Using Petri Nets Based Models", Proceedings of IFIP Protocol Specification, Testing and Verification, may 1982.
- 7 - Esteban P., Valette R., Courvoisier M.: "Simplified Algorithms for Petri Nets Analysis", Proceedings of IECON'86.
- 8 - Esteban P.: "Sur la Recherche d'Algorithmes Simplifiés d'Analyse des Réseaux de Petri". Thèse de Doctorat. LAAS Toulouse, France, 1985.
- 9 - Menasche M.: "Analyse des Réseaux de Petri Temporisés et Application aux Systèmes Distribués". Thèse de Doctorat. Université Paul Sabatier de Toulouse, France, 1982.
- 10- Peterson J.L.: "Petri Net Theory and the Modeling of Systems". Ed. Prentice-Hall 1981.
- 11- Roux J.L., Juanole G.: "Functional and Performance Analysis Using Extended Petri Nets", International Workshop on Petri Nets and Performance Models, aug 1987.
- 12- Sibertin-Blanc C.: "High-Level Petri Nets with Data Structure", 6th European Workshop on Applications and Petri Nets, ESPOD 1985.