Redes de Petri Hierárquicas

ULRICH SCHIEL

Universidade Federal da Paraíba
Departamento de Sistemas e Computação
C.P. 10106 - 58100 Campina Grande/PB

RESUMO

Redes de Petri representam uma ferramenta poderosa para a especificação de sistemas de software com alto grau de paralelismo e complexas condições de interdependência entre seus componentes. Um ponto fraco destas redes é a impossibilidade de uma estruturação hierárquica da descrição de um sistema. Apresentamos aqui Redes de Petri Hierárquicas (HPN) que permitem uma decomposição tanto das unidades de dados como das transições em Sub-Redes.

Palavras-Chave: Redes de Petri, Redes Hierárquicas, Projeto Estruturado

I. INTRODUÇÃO

A utilização de Redes de Petri para o desenvolvimento de sistemas de informação tem várias vantagens: são formais (uma condição necessária para uma implementação correta); seu formalismo gráfico é fácil de aprender e entender; adaptam-se particularmente bem a sistemas com alto grau de paralelismo e interdependência de processos. Já existem trabalhos sobre a combinação de Redes de Petri com orientação a objetos e modelagem do conhecimento (BP-86, KS-88, SM-90). Há entretanto, um problema com o uso destas redes para a descrição de sistemas de grande porte: elas não permitem uma estruturação hierárquica. Sistemas de informação complexos necessitam de estruturação tanto do processo de engenharia como da descrição final do sistema.

Para possibilitar um projeto estruturado de sistemas com Redes de Petri, foram sugeridas várias formas de redes hierárquicas (TM-87, La-86, KS-88). O trabalho mais abrangente neste sentido é o de G. Lausen (La-86), que define uma Rede Hierárquica do Fluxo de Informação, na qual processos são refinados em subredes e unidades de informação podem ser decompostas em unidades de um nível inferior. Sobre objetos abstratos de um grafo chamado Rede da Estrutura Conceitual, podem ser aplicadas operações de especialização, desagregação e dissolução.

As sugestões existentes resolvem o problema de um projeto estruturado de sistemas de informação, mas como, a cada refinamento, o elemento de nível superior desaparece ou permanece indistintamente com os outros, continua-se tendo uma grande rede desestruturada no final do projeto.

Sugerimos aqui uma regra de refinamento com a qual a descrição final não é uma única rede, mas realmente uma hierarquia de redes. Tanto unidades de informação como transições têm procedimentos semelhantes de refinamento; mostramos ainda o equivalente 'plano' de cada refinamento, mantendo-se válidos todos os resultados teóricos sobre Redes de Petri.

II. REDES DE PETRI

Apresentamos neste capítulo os principais conceitos de Redes de Petri, necessários para a introdução das redes hierárquicas. Maiores detalhes podem ser vistos, p.ex., em <Gi.-81>. As hierarquias a serem introduzidas no próximo capítulo, aplicam-se às Redes de Petri de alto nível denominadas Redes Predicado/Transição, mas poderão facilmente ser adaptadas aos outros tipos de redes.

DEFINIÇÃO 1: Uma *Rede de Petri Identificada*, ou simplesmente uma *Rede*, é uma quádrupla

$$R = \langle S, T; F; L \rangle$$

tal que

$S \cap T = \emptyset$
$F \subseteq S \times T \cup T \times S$
L é uma função que associa um nome a cada elemento de $S \cup T$.

Os elementos de S, denominados genericamente *S-elementos*, serão interpretados como *condições*, *lugares* ou elementos que satisfazem um *predicado*, dependendo do tipo de rede. Os elementos de T, ou *T-elementos*, são *eventos* ou *transições*. F representa a relação de *fluxo* de S-elementos. O conjunto $E = S \cup T$ é o conjunto dos *elementos* da rede. Dada uma transição t, distinguimos as entradas e a saídas de t:

$$t^- = \{s \in S \ / \ \langle s,t \rangle \in F\}$$
$$t^+ = \{s \in S \ / \ \langle t,s \rangle \in F\}$$

Uma rede R' é uma *sub-rede* de uma rede R, se todos seus componentes são subconjuntos dos componentes de R. Há um tipo de sub-rede que nos interessará particularmente, é a vizinhança de um elemento da rede. A *vizinhança* de um T-elemento p de uma rede R é a subrede de R formada por

$S_R = \{d \in S_R \ / \ \langle d,p \rangle \in F_R\}$
$T_R = \{p\}$
F_R = todas arestas de F_R ligadas a p

Analogamente definem-se vizinhanças de S-elementos.

O objetivo das Redes de Petri é modelar sistemas em geral. Os possíveis estados de um sistema são representados por marcações distintas dos S-elementos de uma rede.

DEFINIÇÃO 2: Uma **marcação** M de uma Rede de Petri $R = \langle S,T;F;L\rangle$ é uma função

$$M : S \to N$$

que associa a cada S-elemento um número de zero ou mais marcas.

Mudanças de estado são possíveis observando-se as regras de transição para Redes de Petri. Uma rede marcada dotada de uma regra de transição, é chamada de **Sistema de Petri**. Dependendo do contradomínio N da marcação, teremos diferentes tipos de Sistemas de Petri:

1) Se N= 0,1 temos um **Sistema de Condições e Eventos (C/E)**
2) Se N= 0,1,... temos um **Sistema de Lugares e Transições (P/T)**
3) Se existe um universo de objetos U e cada S-elemento representa um predicado que toma valores neste universo, uma marcação seria uma particular constelação de valores. Neste caso teremos N = $P(U)$, o conjunto das partes de U e o sistema é chamado de **Sistema de Predicados e Transições (PrT)**.

A cada sistema está associada uma particular regra de acionamento de T-elementos. Basicamente uma transição está *habilitada* a ser acionada se suas entradas estão marcadas e as saídas estão livres. O *acionamento* de uma transição retira uma marca de cada entrada e coloca uma em cada saída. Nos sistemas PrT a cada transição está associado um predicado de transição, que deve estar satisfeito para que ela se torne habilitada e possa ser acionada.

Resumindo, um Sistema de Petri é formado por uma Rede de Petri, uma marcação (chamada *marcação inicial*) e uma regra de transição, com a qual podemos obter novas marcações (estados) do sistema.

DEFINIÇÃO 3: Dadas duas redes $R1=\langle E1;F1;L1\rangle$ e $R2=\langle E2;F2;L2\rangle$, uma função $f:R1 \to R2$ é um ***morfismo de redes*** quando vale

$$(n1,n2) \in F1 \Rightarrow (\langle f(n1),f(n2)\rangle \in F2 \text{ ou } f(n1) = f(n2))$$

ou seja, um morfismo mantém a relação de fluxo, exceto certas aglutinações de vários elementos do domínio em um só elemento.

O que nos interessa é exatamente o inverso, ou seja, a transformação de um elemento em uma subrede. Portanto, a inversa f^{-1} de um morfismo de redes sobrejetivo é chamado um **refinamento**. Dadas duas redes R1 e R2, se existe um morfismo f de R1 *sobre* R2, dizemos que R1 é um refinamento de R2.

III. REDES HIERÁRQUICAS

Na modelagem de dados para esquemas conceituais de bancos de dados, distinguem-se três critérios que permitem estruturar os dados em vários níveis de abstração para que, a cada nível, sejam considerados somente os dados realmente de interesse a esse nível. São denominadas *generalização*, *agregação* e *agrupamento*. A estas fazemos corresponder as inversas *especialização*, *decomposição* e *dissolução*, que são os conceitos que nos interessam mais diretamente para o refinamento de redes. Resumidamente, a especialização consiste em reconsiderar um objeto acrescentando-lhe mais detalhes, a decomposição considera as partes de um objeto composto e, para objetos que representam grupos de indivíduos, a dissolução consiste em focalizar os elementos do grupo.

Quando vamos analisar a composição de um processo, programa, rotina ou instrução complexa, também encontramos três estruturas básicas: a seqüência, o desvio condicional (case) e a repetição (loop). A estas estruturas corresponderiam respectivamente a decomposição, a especialização e a dissolução.

Esta analogia entre estruturas de dados e de processos será mantida nas regras de refinamentos de elementos de uma Rede de Petri a serem introduzidas. A cada S-elemento e a cada T-elemento podem ser aplicadas três regras de refinamento, transformando o elemento em uma sub-rede.

Em uma Rede de Petri um S-elemento é denotado por uma oval ○ e um T-elemento por um retângulo □ . Na figura 1 esquematizamos um S-elemento *D* cujas marcas são geradas por uma transição *A* e consumidas por uma transição *B*, além de um T-elemento *P* com parâmetros de entrada A e parâmetros de saída B.

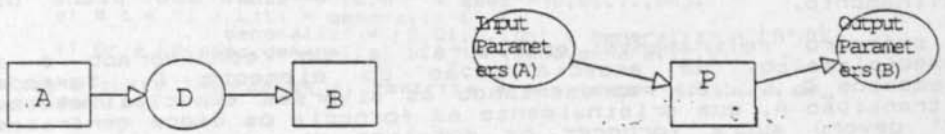


Figura 1 : Constelação inicial antes do refinamento

Basicamente, o refinamento de *D* consiste na definição de transições para o interior de *D* e o refinamento de *P* é feito por predicados interiores a *P*. Estes interiores poderão conter toda uma Sub-Rede de Petri com uma interface bem definida para elementos do nível superior. Este procedimento pode ser repetido tantas vezes quanto necessário. Queremos ressaltar que mesmo os elementos refinados, quando vistos no nível superior, continuam sendo considerados simples elementos de uma Rede sem se tomar conhecimento do comportamento interno destes elementos.

Antes de apresentar as regras específicas de refinamento, é necessário uma observação sobre o comportamento de uma transição *P* refinada. Normalmente o acionamento de uma transição em Redes de Petri é uma ação instantânea que produz os dados de saída. A transição refinada irá primeiro acionar a sub-rede e, só ao final, as saídas serão marcadas. Para refletir este fato, ela deve ser dividida em duas transições, com uma condição de espera entre elas (Fig.2).

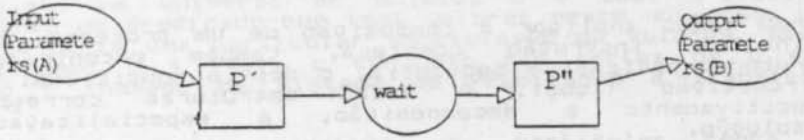


Figura 2 : Transição dividida

No presente trabalho daremos uma ilustração gráfica e uma definição formal do refinamento de especialização de um S-elemento. Para os outros casos, nos limitaremos à ilustração gráfica, deixando a formalização para um trabalho posterior.

Para cada refinamento mostramos graficamente a rede interna ao elemento refinado e, para fins de transformação da rede hierárquica em uma rede plana, que pode ser utilizada para análises teóricas, também apresentamos uma versão plana do refinamento.

O primeiro refinamento estrutural a ser considerado é a especialização. Na especialização do elemento *D*, teremos elementos $D_1,\ldots,D_n$ representando as diversas especializações. A transição A, que originalmente só fornecia os dados genéricos *'a'* deverá agora fornecer os detalhes de um ou mais dados específicos *'b'*. Estes dados são entregues a um S-elemento intermediário D' e uma transição *spec* os distribue para os elementos $D_1..D_n$. Inversamente, se D quer consumir os dados *'a'* deverá consumir todos os complementos *'b'* relacionados com *'a'*. Para tanto, a transição *generaliz* recolhe estes dados do

elemento intermediário D" para serem consumidos por D. A transição C simboliza a possibilidade de definição de novos elementos na subrede.

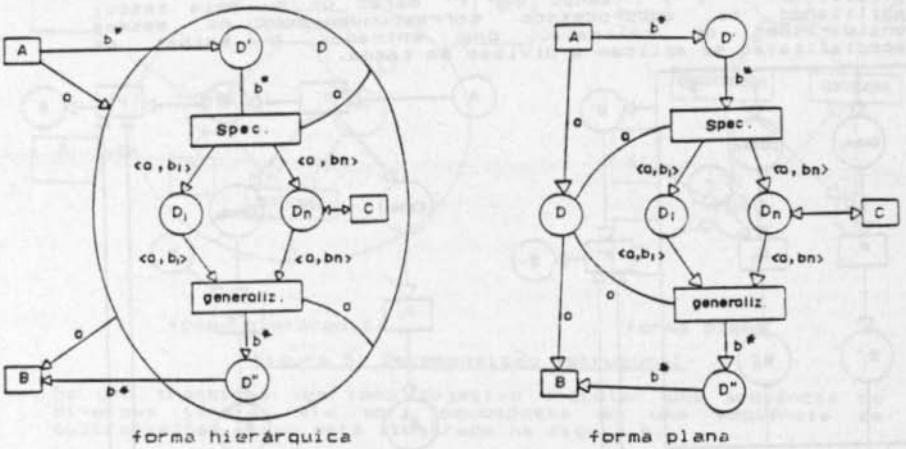


Figura 3: Especialização estrutural

Por exemplo, D poderia ser uma classe de PESSOAS, D1 seriam ESTUDANTES e D2 PROFESSORES. A transição A fornece dados de novas pessoas em 'c' e b1 = b1 U b2 seriam os dados específicos de pessoa como estudante e como professor, respectivamente. Para uma pessoa que não é nem estudante nem professor, teríamos b1 = Ø.

Formalmente, definimos o inverso da especialização, a generalização, como um morfismo que satisfaz certas condições.

DEFINIÇÃO 4: Um morfismo de redes f : R1 -> R2 é uma generalização, quando

a) f(R1) é uma vizinhança de um S-elemento D do R2;
b) ⟨A,D⟩ ∈ F2 => $f^{-1}(A) = A$ & $A^{-} = \{D, D'\}$ em R1
c) (D,B) ∈ F2 => $f^{-1}(B) = B$ & $B^{-} = \{D, D''\}$ em R1
d) ∃ t ∈ T1 / L(t) = spec &
   $spec^{-} = \{D, D'\}$ & $spec^{+} = \{D, D1, \ldots, Dn\}$;
e) ∃ t ∈ T1 / L(t) = generaliz &
   $generaliz^{-} = \{D, D1, \ldots, Dn\}$ & $generaliz^{+} = \{D'', D\}$
f) $D^{-} = \{A, spec, generaliz\}$ & $D^{+} = \{spec, generaliz, B\}$.

É claro que a inversa de f, restrita a sua imagem f(R1), é um refinamento.

O correspondente procedural da especialização é a divisão de uma transição em vários casos. A figura 4 mostra como P foi dividido em P' e P", sendo que P' marca um ou mais casos, habilitando os subprocessos correspondentes. As mesmas considerações de religação das entradas e saídas da especialização se aplicam a divisão em casos.

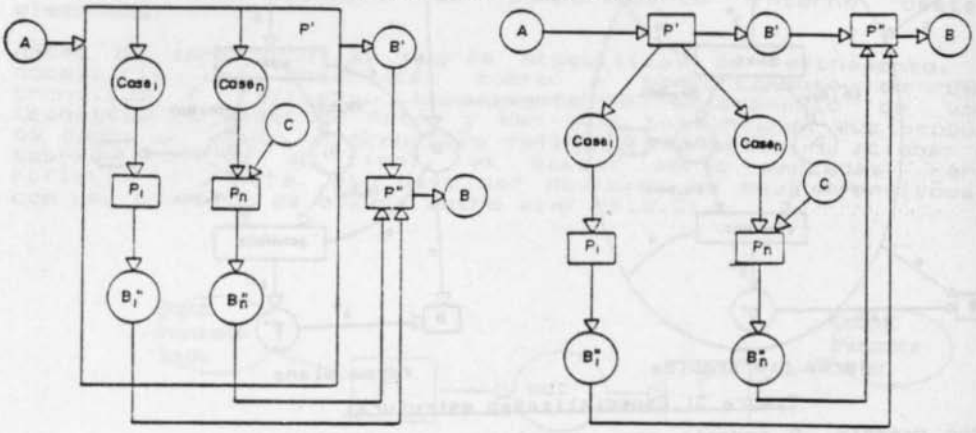
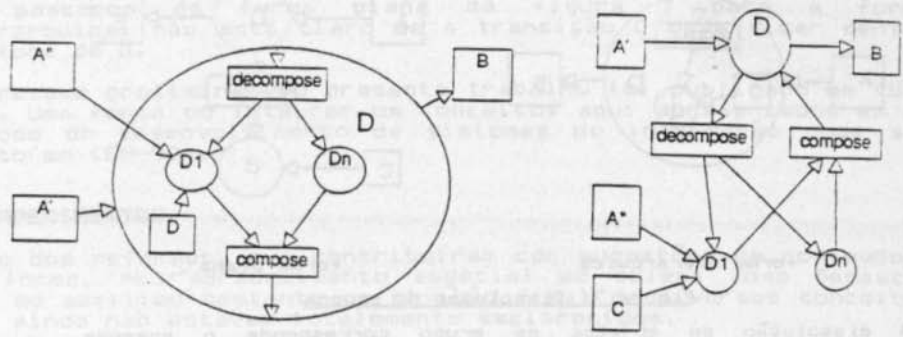


Figura 4: Divisão procedural em casos

Antes de continuar nos outros tipos de refinamento, vamos estabelecer a regra que vai evitar uma ligação indiscriminada de elementos de todos os níveis entre si, desfazendo assim, a própria hierarquia. A regra vai garantir que, no esquema final elementos de um nível *i* somente poderão estar ligados a elementos dos níveis *i*, *i-1* e *i+1*.

RESTRIÇÃO DE RELIGAÇÃO: No refinamento de um elemento de uma Rede Hierárquica entradas e saídas só podem ser religadas se ainda não foram religadas em um passo anterior de religamento.

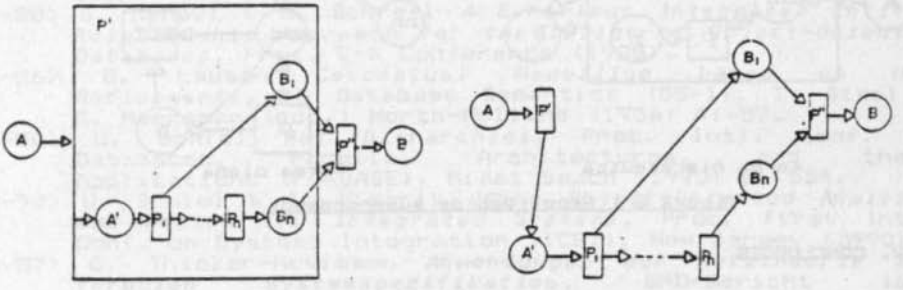
A decomposição de um S-elemento *D* em vários componentes é processada por uma transição *decompose* que gera os componentes, enquanto uma transição *compose* modela a passagem dos componentes para o agregado (figura 5).



forma hierárquica forma plana

Figura 5: Decomposição estrutural

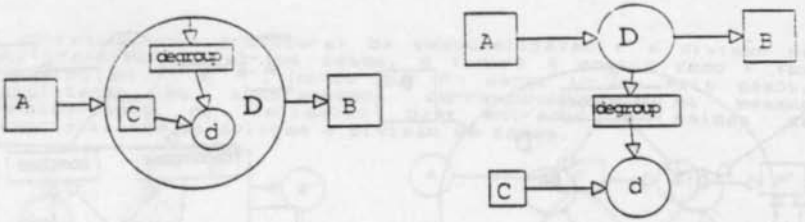
Se uma transição tem como objetivo executar uma seqüência de diversas tarefas ela será decomposta em uma seqüência de subtransações, como está ilustrado na figura 6.



forma hierárquica forma plana

Figura 6: Decomposição procedural em seqüência

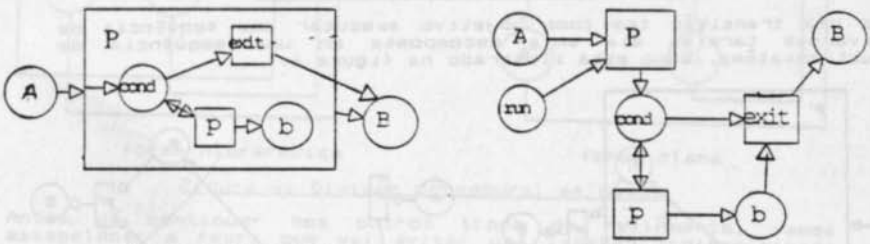
O terceiro e último refinamento ocorre quando cada elemento de *D* representa um grupo ou conjunto de entidades. A transição *degroup* da figura 7 dissolve um grupo e gera em *d* os elementos do grupo dissolvido.



forma hierárquica forma plana

Figura 7: Dissolução de grupos

À dissolução de objetos de grupo corresponde o comando repetitivo de loop. Na próxima figura, a transição *P* está inscrita com uma condição do loop que passa para o subprocesso *p* que, após cada execução, retorna o controle para *cond*. Ao final do loop, a transição *exit* encerra o processo *P*.



forma hierárquica forma plana

Figura 8 : Repetição de subprocessos

IV. CONCLUSÕES

No presente trabalho foram sugeridos três pares de regras de refinamento de Redes de Petri de alto nível. O resultado é uma hierarquia de Redes de Petri dependentes umas das outras, mas cada componente desta hierarquia é, por si só, uma Rede de Petri e pode ser analisado como tal. Chamamos esta hierarquia de Rede de Petri Hierárquica (HPN).

Como a cada refinamento corresponde uma forma plana de rede, fica claro que um HPN pode ser transformada em uma Rede de Petri, e desta forma todos os resultados de Redes de Petri aplicam-se às HPNs. A conversão inversa de uma rede plana em uma HPN não é possível sem informações adicionais. Por exemplo,

na passagem da forma plana da figura 7 para a forma hierárquica, não está claro se a transição *C* deve ficar dentro ou fora de *D*.

Um resumo preliminar do presente trabalho foi publicado em <Sc-90>. Uma forma de integrar os conceitos aqui apresentados em um método de desenvolvimento de sistemas de informação pode ser visto em <SM-90>.

AGRADECIMENTOS

Além dos referees, que contribuiram com sugestões de conteúdo e de forma, meu agradecimento especial ao colega Joao Damasco, que me auxiliou bastante na revisão final de diversos conceitos que ainda nao estavam totalemente esclarecidos.

REFERENCIAS

<BB-86> G. Bruno & A. Balsamo *Petri-Net Object-Oriented Modelling of Distributed Systems*, Proc. OOPSLA (1986).

<GL-81> H.J. Genrich & K. Lautenbach *System Modelling with High-Level Petri Nets*, Theoretical Comp. Science, Vol. 13, (1981) 109-136.

<KS-88> G. Kappel & M. Schrefl *A Behaviour Integrated Entity-Relationship Approach for the Design of Object-Oriented Databases*, Proc. E-R Conference (1988).

<La-86> G. Lausen *Conceptual Modelling based on Net Refinements*, in Database Semantics (DS-1), T. Steel & R. Meersman (eds.) North-Holland (1986) 41-57.

<Sc-90> U. Schiel *Net Hierarchies*, Proc. Intl. Conf. on Databases, Parallel Architectures, and their Applications (PARBASE), Miami Beach (1990) p. 554.

<SM-90> U. Schiel & I. Mistrik *Using Object-Oriented Analysis and Design for Integrated Systems*, Proc. First Intl. Conf. on Systems Integration (ICSI), New Jersey (1990).

<TM-87> G. Thiele-Mevissen *Anwendungen der Netztheorie zur formalen Systemspezifikation*, GMD-Bericht 166, Oldenburg Verlag (1987).