

Representação de Lugares de Fusão em Redes de Petri Coloridas construídas com Componentes Distribuídos

C. G. Furtado Júnior^{1,2}, Luís C. M. de Vasconcelos², J. M. Soares², G. C. Barroso²

¹ Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Departamento de Telemática - Campus Maracanaú

² Universidade Federal do Ceará (UFC)
Departamento de Teleinformática (DETI) – Centro de Tecnologia

cjunior@ifce.edu.br, luis.vasconcelos@sti.ufc.br,
marques@ufc.br, gcb@fisica.ufc.br

Abstract. *In this paper, we propose a solution to the problem of mutually exclusive access to shared areas of a simulation in distributed CPN (DCPN), which enables the paralleling of no conflicting events and without compromising the analytical power associated with traditional RPC.*

Resumo. *Neste trabalho propomos uma solução para o problema do acesso mutuamente exclusivo às regiões compartilhadas de uma simulação em RPC distribuída (RPCD), que possibilite o paralelismo de eventos não conflitantes e sem comprometer o poder analítico associado às tradicionais RPC.*

1. Introdução

Sistemas do mundo real são compostos por diversas partes que apresentam comportamentos autônomos e que, comumente, comunicam-se e cooperam entre si. Para lidar com a complexidade na especificação, análise e desenvolvimento desses sistemas distribuídos (SD), faz-se necessário o uso de modelagem e simulação.

As características dos SD, que incluem concorrência, sincronização, sequência, conflito, dentre outros, podem ser modeladas por Redes de Petri Coloridas (RPC) (Jensen e Kristensen 2009). Entretanto, as diversas ferramentas disponíveis para construção de modelos e simulação de RPC (CPN Tools 2016) representam soluções distribuídas como blocos de um modelo que, embora possam ser modularizados em sub-redes, executam em processos centralizados. Em um SD, cuja complexidade de representação e desenvolvimento é maior que os sistemas centralizados, a simulação centralizada em um processo único pode dificultar a análise e o reuso de partes específicas do modelo, bem como degradar o desempenho nas simulações.

Uma alternativa para lidar com essa limitação é construir simulações com partes independentes, configurando o próprio modelo como um SD, em que diferentes processos, cada um correspondendo a uma rede independente, simulam concorrentemente as partes distintas do sistema modelado. Como cada processo realiza parte do processamento necessário ao modelo global, os resultados podem ser consolidados para permitir uma análise do comportamento global do sistema. Por outro lado, componentes distribuídos do sistema podem ser analisados individualmente, permitindo inferir sobre a sua atuação no sistema. O objetivo deste trabalho é propor uma solução para o problema do acesso mutuamente exclusivo às regiões compartilhadas de uma simulação em RPC distribuída (RPCD), que possibilite o paralelismo de eventos não conflitantes e sem comprometer o poder analítico associado às tradicionais RPC.

Este documento é organizado da seguinte maneira: após a introdução realizada nesta seção, as Redes de Petri Coloridas são apresentadas na Seção 2. A Seção 3 traz uma introdução

às Redes de Petri Coloridas Distribuídas (RPCD). Na Seção 4 são apresentados os problemas inerentes à distribuição das RPC. O Controle de Múltiplo Acesso é apresentado na Seção 5. Por fim, as considerações finais são apresentadas na Seção 6.

2. Redes de Petri Coloridas

Redes de Petri Coloridas (RPC) são uma linguagem gráfica para a construção de modelos de sistemas de eventos discretos e análise de suas propriedades. Estas combinam as capacidades das Redes de Petri (PN) (Murata, 1989) com as capacidades de uma linguagem de programação de alto nível (Jensen e Kristensen 2009).

A estrutura principal de uma RPC consiste basicamente de três estruturas: lugares (representados por círculos ou elipses), transições (representados por retângulos) e arcos (representados por setas direcionais). Lugares são utilizados para especificar um estado ou condição de um componente do sistema, enquanto que as transições representam eventos ou operações. Para interligar os componentes de uma RPC, existem dois tipos de arcos direcionados: arcos de entrada, que ligam uma transição a um lugar, e os arcos de saída, que ligam um lugar a uma transição. A dinâmica da rede apoia-se no uso de fichas, que são depositadas em lugares. As fichas podem representar recursos ou condições, descrevendo o estado da rede em um instante no tempo. Estas são inseridas ou retiradas dos lugares através do disparo de uma transição. Para que uma transição possa ser disparada é necessário que haja uma quantidade de fichas nos lugares de entrada igual ou superior aos pesos dos seus respectivos arcos de entrada. Após o disparo, fichas são inseridas nos lugares de saída da transição de acordo com o peso dos arcos de saída. É através dos disparos das transições que percebemos o fluxo de controle e de informações no sistema. Um estudo mais aprofundado sobre as RPC é fornecido por (Jensen e Kristensen, 2009).

3. Redes de Petri Coloridas Distribuídas

Costa e Gomes (Costa e Gomes, 2009) apresentam um método para a segmentação de Redes de Petri (RP) capaz de dividir um modelo em vários outros submodelos, que se comunicam entre si através de canais síncronos. A decomposição do modelo original é realizada a partir da definição de um conjunto de corte válido, obtido através da aplicação de três regras básicas (Costa and Gomes 2009), o que pode impor restrições na distribuição dos componentes do modelo. Para representação dos componentes distribuídos são utilizadas uma extensão das RP, as Input-Output Place-Transition nets (IOPT) (Gomes et al. 2007), que possuem semântica distinta das tradicionais RP. Além disso, não são tratadas questões como concorrência por recursos e acesso mutuamente exclusivo às regiões compartilhadas.

Knoke e Hommel (Knoke and Hommel 2005) propuseram um mecanismo para simulação otimista de RPC na presença de restrições globais. Nessa modalidade de simulação, os eventos distribuídos são executados independentemente, sem preocupação com a restrição de causalidade. Segundo os autores, uma simulação distribuída cujas transições disparam de forma otimista exige que os eventos concorrentes sejam ordenados por prioridades (Knoke and Hommel 2005), excluindo o problema do conflito distribuído, o que pode comprometer o paralelismo na execução de modelos independentes.

Neste trabalho, informalmente, uma de Rede de Petri Colorida Distribuída é definida pela associação de dois ou mais módulos de RPC (MRPC), que interagem entre si através de trocas de mensagens, e que juntos especificam um sistema único e coerente. A comunicação entre as partes distribuídas se dá por meio do compartilhamento de lugares de fusão em dois ou mais MRPC distintos e que, representando o mesmo objeto no sistema. O lugar de fusão é implementado pelo CPN-Tools para representar um lugar compartilhado pelas sub-redes, possuindo o mesmo estado independente da sub-rede em que seja representado. Em modelos

distribuídos, com processos independentes, o desafio é manter o estado de um lugar de fusão consistente em todos os componentes ao longo da simulação. Devido a essa particularidade, neste trabalho denomina-se um lugar compartilhado por diferentes módulos componentes do modelo distribuído por Lugares de Fusão Distribuídos (LFD).

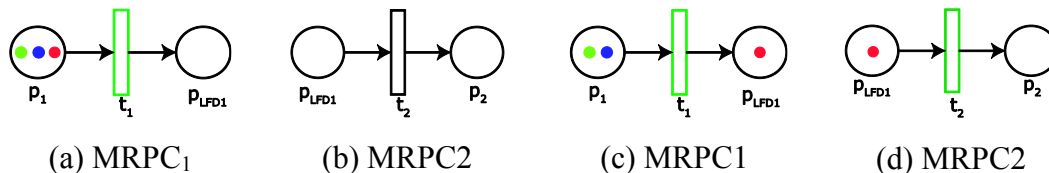


Figura 1. Rede de Petri Colorida Distribuída

A Figura 1(a) e Figura 1(b) apresentam uma RPCD composta por dois MRPC, denominados MRPC₁ e MRPC₂, em seu estado inicial (step = 0). O disparo da transição t_1 no MRPC₁ retira uma ficha do lugar de entrada p_1 e a deposita no P_{LDF1} . Como P_{LDF1} é representado em mais de um módulo da RPCD, para que a consistência seja preservada em uma simulação do sistema modelado, é necessário propagar essa alteração de estado para os demais MRPC que contém P_{LDF1} . Os novos estados dos MRPC da Figura 1(a) e Figura 1 (b), após o disparo da transição t_1 (step = 1), pode ser observado na Figura 1(c) e Figura 1(d), respectivamente. Na seção seguinte caracterizamos o problema da manutenção da consistência de estados e simulação concorrente em uma RPCD.

4. Problemas na simulação concorrente de uma RPCD

A Figura 2 apresenta três partes de um modelo distribuído concebido apenas para fins didáticos. São eles: MRPC₁, MRPC₂ e MRPC₃.

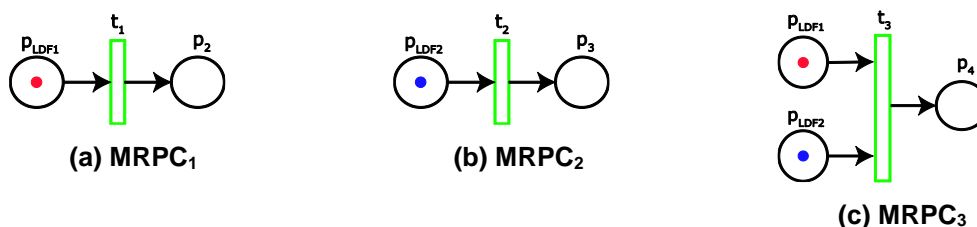


Figura 2. RPCD com acesso concorrente ao LFD P_{LDF1} e LFD P_{LDF2}

Três eventos distribuídos (t_1 , t_2 e t_3) concorrem por acesso ao LFD P_{LDF1} e LFD P_{LDF2} . Diversos são os algoritmos disponíveis na literatura para acesso mutuamente exclusivo às regiões compartilhadas. Dente eles nós citamos o algoritmo com base no servidor central (Coulouris et al. 2011), algoritmo baseado em anel (Coulouris et al. 2011) e o algoritmo utilizando *multicasting* e relógios lógicos de Ricart e Agrawala (Ricart e Agrawala, 1981). Entretanto, a adoção desses algoritmos pode comprometer o paralelismo de eventos não conflitantes ou levar as simulações em RPCD a um estado de bloqueio ou inconsistência.

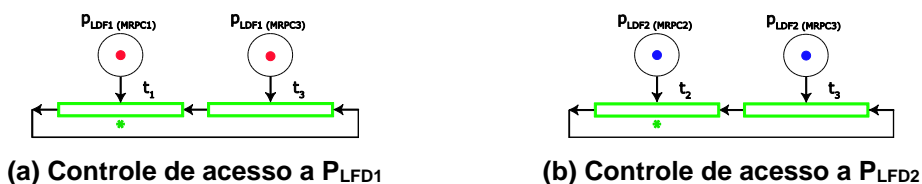


Figura 3. Anéis representando em (a) o controle de acesso ao LFD P_{LDF1} e em (b) o controle de acesso ao LFD P_{LDF2} da RPCD ilustrada na Figura 2.

Para exemplificar uma situação de bloqueio, é utilizada a lógica do Token Ring (Coulouris et al. 2011). Na Figura 3 são apresentadas duas estruturas de controle em anel, uma para acesso a P_{LDF1} e outra para acesso a P_{LDF2} . A manutenção de um anel de permissões para cada conjunto de LFD permite a ocorrência simultânea de eventos paralelos. Os bastões de permissão de acesso são representados por asteriscos próximos às transições t_1 e t_2 , indicando que estes eventos detêm a permissão para o disparo.

Supondo que, por um procedimento aleatório qualquer, os bastões de permissões sejam entregues a t_1 , liberando o acesso ao LFD P_{LDF1} pelo $MRPC_1$, e a t_3 , liberando a esta transição apenas o acesso ao LFD P_{LDF2} , estaria habilitado o disparo da transição t_1 , mas não o disparo de t_3 , que precisa de acesso simultâneo ao LFD P_{LDF1} e LFD P_{LDF2} . O disparo de t_1 retira uma ficha do LFD P_{LDF1} e a deposita no lugar p_2 . O modelo global, nessa situação encontra-se em estado de bloqueio, pois t_3 nunca pode disparar sem acesso simultâneo ao LFD P_{LDF1} , que agora encontra-se com zero fichas. Além disso, t_2 também não pode disparar, pois não detém o bastão de permissões.

5. Controle de Múltiplo Acesso

A solução proposta neste trabalho para contornar o problema apresentado na seção anterior, denominada Controle de Múltiplo Acesso (CMA), é baseada no algoritmo de Ricart e Agrawala (Ricart e Agrawala, 1981), que foi adaptado para excluir situações onde o modelo seria levado a estados bloqueados. O algoritmo compreende duas etapas. A etapa inicial corresponde à análise das relações de conflito e paralelismo existentes entre os eventos do modelo global. Nela é definida uma tabela de relacionamento que deve ser de conhecimento de todos os $MRPC$ que compõem a $RPCD$. A Tabela 1 apresenta um resumo das relações existentes entre os eventos t_1 , t_2 , t_3 do modelo apresentado na Figura 2. Nela, os símbolos \circ e \parallel são utilizados para expressar as relações de conflito e concorrência, respectivamente.

Tabela 1. Relação entre eventos do modelo apresentado na Figura 2

	t_1	t_2	t_3
t_1		\parallel	\circ
t_2	\parallel		\circ
t_3	\parallel	\parallel	

A etapa subsequente determina que, quando um processo deseja ter acesso a um LFD, o mesmo deve enviar uma mensagem *multicast* a todos os $MRPC$. O acesso ao LFD, entretanto, só será liberado quando todos os outros responderem a mensagem (propriedade 1: segurança).

Os processos $MRPC_1$, $MRPC_2$, ..., $MRPC_n$ possuem identificadores numéricos únicos. Mensagens de solicitação de uso da região crítica obedecem a regra de formação $\langle REQ, T, MRPC_i, LFD_j, Tr \rangle$, em que REQ é palavra reservada do cabeçalho que identifica uma mensagem de requisição de uso de um LFD; T é a marca de tempo do emissor; $MRPC_i$ é o identificador único do emissor, LFD_j é o de LFD que o $MRPC_i$ deseja acessar e, por fim, Tr é a transição do $MRPC_i$ que deseja disparar.

Cada $MRPC$ pode estar em um dos três estados: **RELEASED**, quando está fora da região crítica; **WANTED**, quando deseja ter acesso a região crítica ou **HELD**, quando está acessando a região crítica.

Se um $MRPC$ deseja entrar em uma de região crítica e todos os outros $MRPC$ estão no estado **RELEASED**, então todos estes irão responder imediatamente ao solicitante e este obterá o acesso desejado. Caso algum processo esteja no estado **HELD** em relação a um LFD, este não responderá a uma nova solicitação até liberar a região compartilhada, impedindo outro $MRPC$

de ter acesso a esse recurso imediatamente. Caso dois processos solicitem acesso a um LFD simultaneamente, o que tiver menor marca de tempo irá conseguir ter mais rapidamente (N-1) respostas e garantirá acesso ao lugar compartilhado. Caso os dois processos tenham a mesma marca temporal, o de menor identificador numérico ganha a disputa.

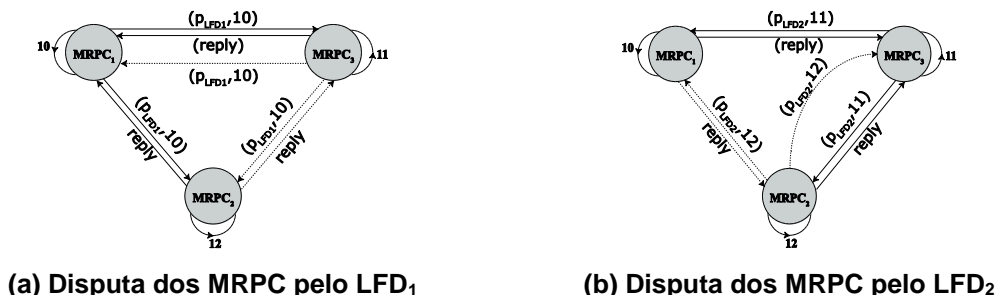


Figura 4. Ordem de acesso ao LFD₁ e LFD₂

Definido o MRPC que tem acesso a um ou mais LFD, a transição associada ao LFD pode disparar. O disparo de uma transição deve ser comunicado para os outros MRPC através de uma mensagem obedecendo a regra de formação $\langle FIRE, MRPC_i, Tr, step, time, count \rangle$, onde FIRE é uma palavra reservada do cabeçalho da mensagem e contém a indicação de disparo de uma transição; MRPC_i identifica o módulo que disparou a transição, Tr representa a transição disparada, step, time e count representam o passo de simulação, o tempo de simulação e um contador, respectivamente.

De posse dessa mensagem e das relações entre eventos exemplificada na Tabela 1, os outros MRPC podem executar de forma concorrente e no mesmo passo de simulação (step) apenas as transições não conflitantes com a transição escolhida para disparar. Caso não existam transições habilitadas e não conflitantes, o MRPC passa do estado HELD para RELEASE, liberando o recurso compartilhado para outro MRPC (propriedade 2: longevidade).

Para ilustrar o funcionamento do algoritmo, considere a situação da Figura 3 onde três eventos t_1 , t_2 e t_3 concorrem por acesso ao P_{LFD1} e P_{LFD2} . A Figura 4 (a) representa a negociação entre MRPC₁ e MRPC₃ para acesso a P_{LFD1} e a Figura 4 (b) representa a negociação entre MRPC₂ e MRPC₃ para acesso a P_{LFD2} .

Na Figura 4 (a), quando MRPC₂, que não deseja acesso a P_{LFD1} , e MRPC₃, cujo tempo é maior que o tempo de MRPC₁, recebem a solicitação de MRPC₁, representadas por linhas contínuas, eles respondem imediatamente liberando o acesso às regiões críticas para esse módulo. O processo MRPC₃ também envia a solicitação de acesso à região compartilhada (linhas pontilhadas) P_{LFD1} para os demais MRPC, entretanto, apenas MRPC₂ responde imediatamente uma vez que não deseja acesso a esse recurso compartilhado. A liberação para acesso a P_{LFD1} por MRPC₃ só é possível quando MRPC₁ sair do estado HELD para o estado RELEASE.

Da mesma forma, na Figura 4 (b), quando MRPC₁, que não deseja acesso a P_{LFD2} , e MRPC₂, cujo tempo é maior que o tempo de MRPC₃, recebem a solicitação de MRPC₃ para acesso a LFD₂, representadas por linhas contínuas, eles respondem imediatamente liberando o acesso à região crítica para esse módulo. O processo MRPC₂ também envia a solicitação de acesso à região compartilhada P_{LFD2} para os demais MRPC, entretanto, apenas MRPC₁ responde imediatamente uma vez que não disputa o acesso a esse recurso compartilhado. A liberação para acesso a P_{LFD2} por MRPC₂ só é possível quando MRPC₃ sair do estado HELD para o estado RELEASE. Assim, temos que o acesso a P_{LFD1} dá-se, primeiramente, pelo MRPC₁ (t_1) e posteriormente pelo MRPC₃. Em relação a P_{LFD2} , o acesso é feito inicialmente por MRPC₃ e, por fim, por MRPC₂.

O disparo da transição t_1 leva o $MRPC_1$ para o estado RELEASE e envia a mensagem $M = \langle FIRE, MRPC_1, [t_1], step, time, count \rangle$ para $MRPC_3$. Isso o leva do estado WANT para o estado HELD. Entretanto, observado a Tabela 1, $MRPC_3$ conclui que não pode disparar t_3 , pois $t_1 \circ t_3$. Então, $MRPC_3$ libera a região compartilhada e envia a mensagem $M = \langle FIRE, MRPC_3, [t_1, nil], step, time, count \rangle$ para $MRPC_2$, que tem seu estado alterado de WANT para HELD. Analisando a tabela de relações, $MRPC_2$ pode disparar t_2 ($t_2 \parallel t_1$). Em outro cenário, onde $MRPC_3$ tenha menor marca temporal que os demais, este fará uso imediato e simultâneo do P_{LFD1} e P_{LFD2} , sendo possível o disparo da transição t_3 .

6. Conclusões

Neste trabalho apresentamos informalmente a definição de Redes de Petri Coloridas Distribuídas e discutimos os problemas associados ao compartilhamento e acesso concorrente aos lugares compartilhados. Propusemos e apresentamos o Controle de Múltiplo Acesso, algoritmo para acesso mutuamente exclusivo aos LFD de uma RPCD que permite a execução concorrente de MRPC com eventos não conflitantes. Foi demonstrado que este atende aos requisitos essenciais de um algoritmo para exclusão mútua: segurança e longevidade. O primeiro requisito implica que dois MRPC não têm acesso simultâneo à mesma região crítica, enquanto o segundo implica ausência de estados de bloqueio e inanição de algum MRPC. Por fim, para o correto entendimento do algoritmo, foi apresentado o passo a passo da evolução do CMA na sua aplicação em uma RPCD com conflito distribuído efetivo.

Diferente das abordagens tradicionais, o CMA não lida com conflitos distribuídos através da associação de prioridades aos eventos. Em se tratando de RPC, essa prática pode comprometer fortemente o poder analítico associado à ferramenta. Embora o CMA tenha sido apresentado no âmbito das RPCD, este pode ser generalizado para os diversos tipos de SD que precisam manter o paralelismo de eventos na presença de conflitos efetivos e sem associação de prioridades.

Agradecimentos

Este trabalho apresenta a metodologia e resultados do projeto de pesquisa 468835/2014-8 do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) denominado “Uma plataforma distribuída para a modelagem colaborativa, simulação de alto desempenho e análise de sistemas híbridos”. Agradecemos o suporte e incentivo do CNPq.

References

- Costa, A. e Gomes, L. (2009). Petri net partitioning using net splitting operation. In *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on*, pages 204–209.
- Coulouris, G., Dollimore, J., Kindberg, T., e Blair, G. (2011). *Distributed Systems: Concepts e Design*. Addison-Wesley Publishing Company, USA, 5th edition.
- CPN Tools (2016). CPN Tools Homepage. Online: available in <http://cpntools.org>. Acesso em Janeiro 2016.
- Jensen, K. e Kristensen, L. M. (2009). *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer Publishing Company, Incorporated, 1st edition.
- Knoke, M. e Hommel, G. (2005). Dealing with global guards in a distributed simulation of colored petri nets. In *Distributed Simulation and Real-Time Applications, 2005. DSRT 2005 Proceedings. Ninth IEEE International Symposium on*, pages 51–58.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.
- Ricart, G. and Agrawala, A. K. (1981). An optimal algorithm for mutual exclusion in computer networks. *Commun. ACM*, 24(1):9–17.