



Production

ISSN: 0103-6513

production@editoracubo.com.br

Associação Brasileira de Engenharia de  
Produção  
Brasil

Dallavalle de Pádua, Silvia Inês; Yassushi Inamasu, Ricardo  
Mapeamento do modelo de processos de negócio do EKD em redes de Petri  
Production, vol. 18, núm. 2, mayo-agosto, 2008, pp. 260-274  
Associação Brasileira de Engenharia de Produção  
São Paulo, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=396742033005>

- Como citar este artigo
- Número completo
- Mais artigos
- Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica

Rede de Revistas Científicas da América Latina, Caribe, Espanha e Portugal

Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

# Mapeamento do modelo de processos de negócio do EKD em redes de Petri

Silvia Inês Dallavalle de Pádua FEARP-USP

Ricardo Yassushi Inamasu EMBRAPA

## RESUMO

O *EKD* – Enterprise Knowledge Development – é uma metodologia que fornece uma forma sistemática e controlada de analisar, entender, desenvolver e documentar uma organização. Infelizmente não possui uma sintaxe e semântica bem definidas, dificultando análises mais complexas dos modelos. Como resultado, o modelo de processos de negócio do *EKD* pode ser ambíguo e de difícil análise, principalmente em sistemas mais complexos, não sendo possível verificar a consistência e completude do modelo. Neste trabalho, esses problemas serão tratados sob uma abordagem baseada em *redes de Petri*. Assim, este trabalho tem como objetivo apresentar o mapeamento do modelo de processos de negócios do EKD em redes de Petri. O mapeamento permitirá que o modelo de processos de negócios do EKD seja analisado de forma que seja possível verificar a consistência e completude do modelo.

## PALAVRAS-CHAVE

Modelo de processos de negócios, formalização do modelo de processos de negócio, redes de Petri, modelagem organizacional, EKD.

## *Mapping of the business processes model the EKD in Petri nets*

## ABSTRACT

The *EKD* – Enterprise Knowledge Development – is a methodology that gives a systematic and controlled way to analyze, understand, develop, and document an Enterprise. Unfortunately it doesn't have syntax neither a semantic well defined, which doesn't help on more complex analyses of the models. As a result, the Enterprise Process model of *EKD* can be ambiguous and hard to analyze, especially on more complex systems, and also it is not possible to verify the consistency and entireness of the model. On this paper, these problems will be studied under an approach based on Petri Nets. This work has the objective to develop a mapping of the business process model of *EKD*. Such mapping will make possible the verification of possible building and locking model errors.

## KEY WORDS

Business process model, Petri nets, enterprise modeling, assessment of business process model, *EKD*.

## 1. INTRODUÇÃO

A tecnologia da informação, direcionada para o gerenciamento e melhoria dos processos de negócio, tem ajudado a organização a completar sua visão da empresa e a melhorar sua posição competitiva. A tecnologia da informação não deveria ser utilizada apenas para automação dos processos de negócio existentes, mas também ser uma base para a reformulação desses processos visando a encontrar os objetivos de negócio existentes, como concorrência, competitividade e estratégias.

Existem muitos sistemas que, embora tecnicamente corretos, não satisfazem as reais necessidades do negócio. Sistemas que não satisfazem as necessidades da organização podem impedir o desenvolvimento do negócio. Nesse contexto, de acordo com Rolland et al. (2007), Nurcan e Barrios (2003), Persson (2000), Nurcan e Rolland et al. (2003), Alencar (1999) e Santander (2002), a modelagem organizacional facilita a compreensão do ambiente empresarial e é reconhecida como uma atividade valiosa para o desenvolvimento de sistemas de informação.

O processo de modelagem organizacional deve trazer respostas a essas questões: por que, o que, quem, qual, quando, onde e como. Para tanto, existem diversas técnicas de modelagem na literatura com uma significativa variedade de notações. A abordagem que será utilizada neste trabalho é o EKD – *Enterprise Knowledge Development* – uma metodologia que fornece uma forma sistemática e controlada de analisar, entender, desenvolver e documentar uma organização e seus componentes, usando a Modelagem Organizacional (ROLLAND et al., 2000; BUBENKO et al., 1998; DALLAVALLE, 2001; NURCAN, 1999). Essa metodologia é explicada em detalhes em Pádua et al. (2004) e Pádua (2001).

O EKD também contribui para a tomada de decisão em modernas organizações que são altamente dependentes de tecnologia de informação (NURCAN; BARRIOS, 2003; NURCAN; ROLLAND, 2003). De acordo com Bubenko et al. (1998), os tipos de submodelos do método EKD são: Modelo de Objetivos, Modelo de Regras do Negócio, Modelo de Conceitos, Modelo de Processos do Negócio, Modelo de Atores e Recursos e Modelo de Requisitos e Componentes Técnicos.

O principal problema das abordagens de Modelagem Organizacional, incluindo-se o EKD, é a ausência de técnicas capazes de análises mais complexas. O problema da estrutura informal das técnicas de modelagem organizacional e de processos de negócio tem sido discutido por diversos autores. Entre eles, podem-se mencionar: Dongen et al. (2007);

Lenz et al. (2005); Mevius e Oberweis (2005); Chen-Burger et al. (2002); Chen-Burger (2001); Koubarakis e Plexousakis (2002); Jonkers et al. (2003); Taylor e Tofts (2003); Junginger et al. (2001); Jonkers et al. (2003); Berio e Petit (2003); Chen-Burger e Stader (2003); Dehnert (2003); Pádua (2004); Pádua et al. (2003); Aalst (1999).

A sintaxe e a semântica do modelo de processos de negócio do EKD não são bem definidas formalmente e rigorosamente. Como resultado, o modelo de processos de negócio do EKD pode ser ambíguo e de difícil análise, principalmente em sistemas mais complexos, não sendo possível verificar a consistência e completude do modelo. A ausência de semântica formal dificulta, também, o uso de técnicas mais eficientes de análise. Neste trabalho, esses problemas foram estudados sob uma abordagem baseada em redes de Petri. O formalismo das redes de Petri as torna uma poderosa técnica de modelagem para a representação de processos, permitindo a exibição de: concorrência, paralelismo, sincronização, não-determinismo e exclusão mútua.

O processo de modelagem organizacional deve trazer respostas a essas questões: por que, o que, quem, qual, quando, onde e como.

Muitos trabalhos têm valorizado a estrutura formal das redes de Petri para representação de processos de negócios, entre eles podem-se mencionar Verbeek et al. (2007), Guan et al. (2006), Zhang e Shuzen (2006), Ou-Yang e Lin (2007), Aalst e Hee (2002), Pádua (2004), Pádua et al. (2003), Aalst (1999).

Assim, este trabalho tem como objetivo apresentar o mapeamento do modelo de processos de negócio do EKD em redes de Petri desenvolvido por Pádua (2004).

## 2. REDES DE PETRI E MODELO DE PROCESSOS DE NEGÓCIO

As diversas aplicações das redes de Petri na Engenharia são apresentadas em Pádua et al. (2003). Em Aalst e Hee (2002), Neiger e Churilov (2003), Chrastowski-Wachtel et al. (2003), McDermid (2003), Rinderle et al. (2003), Verbeek et al. (2002), Pádua et al. (2002), Pádua et al. (2004), Janssens et al. (2000), Holt (2000), Desel (2000), Dellarocas e Kleins (2000), Hee e Reijers (2000), Knolmayer et al. (2000) e em Salimifard e Wright (2001), são apresentados o potencial das redes de Petri em modelagem e análise de processos de negócio.

Desde que Zisman (1977) usou redes de Petri para modelar *workflow* pela primeira vez, muitos autores publicaram trabalhos que procuravam, também, integrar os dois assuntos. Entre eles, podem-se mencionar: Chrzastowski-Wachtel et al. (2003); Rinderle et al. (2003); Chen-Burger e Stader (2003); Cindio et al. (1988); Li et al. (1993); Ferscha (1994); Merz et al. (1995); Schömig e Rau (1995); Ellis e Nutt (1996); Nutt (1996); Wikarski (1996); Oberweis (1997); Aalst (1998); Adam et al. (1998); Badouel e Oliver (1998); Dehnert (2003); Aalst e Hee (2002); Eshuis e Dehnert (2003); Grigova (2003); Verbeck et al. (2002). Aalst e Hee (2002) afirmam que existem diversas razões para usar redes de Petri para modelagem de processos de negócio:

1. Semântica formal: um processo de negócio especificado em uma rede de Petri tem definição clara e precisa, por serem a semântica da rede de Petri clássica e suas extensões definidas formalmente.
2. Natureza gráfica: rede de Petri é uma linguagem gráfica intuitiva e de fácil aprendizado. A natureza gráfica também apóia a comunicação com usuários finais.
3. Expressividade: redes de Petri suportam todas as primitivas necessárias para modelar um processo de negócio. Todas as construções das rotas presentes nos sistemas de gerenciamento de processos de negócio atuais podem ser modeladas. Além disso, o fato de os estados serem representados explicitamente permite escolhas explícitas na modelagem de processos.
4. Propriedades: nas últimas três décadas, muitas pessoas têm investigado as propriedades básicas de redes de Petri. A fundamentação matemática firme permite algumas conclusões e verificações importantes.
5. Análise: as redes de Petri são caracterizadas pela disponibilidade de muitas técnicas de análise. Claramente, esse é um ponto positivo a favor do uso de redes de Petri para modelagem de *workflow*. Essas técnicas podem ser usadas para provar propriedades (segurança, *deadlock*, entre outras) e para calcular a métricas de *performance* (tempo de resposta, tempo de espera, taxa de ocupação). Dessa forma, é possível avaliar *workflows* alternativos usando ferramentas de análise baseadas em redes de Petri.
6. Não dependente de fornecedor: rede de Petri é uma ferramenta de modelagem com estrutura para modelagem e análise independente de um vendedor.

### 3. FORMALIZAÇÃO DO MODELO DE PROCESSOS DE NEGÓCIO

Para que seja possível realizar o mapeamento do Modelo de Processos de Negócio em Redes de Petri foi criada, ba-

seada em Aalst (1999), uma definição formal do Modelo de Processos de Negócio do EKD (MPN-EKD). Dessa forma, foi possível descrever os requisitos que um Modelo de Processos de Negócio deve satisfazer para que o mapeamento seja desenvolvido.

Para essa definição formal do Modelo de Processos de Negócio foi criado um conjunto de conectores para o Modelo de Processos de Negócio do EKD. O conjunto de conectores é representado por  $C$  e é composto por  $C_{AND}$ ,  $C_{OR}$ ,  $C_I$ ,  $C_S$ ,  $C_{IP}$  e  $C_{PI}$ . Os conectores  $C_{OR}$  e  $C_{AND}$  foram criados para identificar escolha (exclusiva) e paralelismo para que os casos de paralelismo e escolha não sejam modelados exatamente da mesma forma, criando ambigüidades e dificuldades de compreensão. Os conectores  $C_I$  e  $C_S$  definem conectores do tipo *join* e *split*. Para descrever a natureza do fluxo dos processos e de suas interações existe um conjunto de termos, utilizados em *Workflow Management Coalition* (1996) e em Aalst e Hee (2002), que são apresentados a seguir:

- *AND-Split*: ponto em que, de uma única linha de fluxo, partem duas ou mais linhas que são executadas em paralelo.
- *AND-Join*: ponto em que duas ou mais atividades, executando em paralelo, convergem em uma única linha de fluxo comum.
- *OR-Split*: ponto em que uma única linha de fluxo faz uma decisão entre um número de opções.
- *OR-Join*: ponto no qual uma atividade que possui um número de alternativas direciona-se para uma única opção.
- De acordo com essas definições de AND-Split, AND-Join, OR-Split e OR-Join, as construções da Figura 1 não são permitidas no MPN-EKD formal.

Os conectores  $C_{IP}$  e  $C_{PI}$  demonstram que um conector  $C$  é um caminho de um *inf-set* para um processo ou um caminho de um processo para um *inf-set*.

Os estados inicial e final não são especificados no Modelo de Processos de Negócio do EKD, foi necessário criar esses estados para que a formalização fosse efetivamente realizada. Essa situação será explicada no decorrer deste artigo.

**Definição 1.** Um MPN-EKD é uma quintupla  $(I, P, C, Q, A)$ :

- $I$  é um conjunto finito de *inf-set* (conjunto de informações);
- $P$  é um conjunto finito de processos;
- $C$  é um conjunto finito de conectores lógicos;
- $Q \in C \rightarrow \{AND, OR\}$  é uma função que mapeia cada conector dentro de um tipo de conector;
- $A \subseteq (I \times P) \cup (P \times I) \cup (I \times C) \cup (C \times I) \cup (P \times C) \cup (C \times P)$  é um conjunto de arcos.

Um MPN-EKD é composto por três tipos de elementos: *inf-set* – conjunto de informações (I), processos (P) e conectores (C). O tipo de cada conector é dado pela função Q:  $Q(c)$  é o tipo (AND ou OR) de um conector  $c \in C$ . A relação A especifica um conjunto de arcos conectando processos, conjunto de informações (*inf-set*) e conectores. A definição 1 demonstra que não é permitido ter um arco conectando dois processos ou dois *inf-sets* ou dois conectores.

**Definição 2.** Um caminho direcionado p de um nodo  $n_i$  para um nodo  $n_k$  é uma sequência  $\langle n_1, n_2, \dots, n_k \rangle$ , tal que  $\langle n_i, n_{i+1} \rangle \in A$  para  $1 \leq i \leq k-1$ . p é elementar se, e somente se, para qualquer um dos nodos  $n_i$  e  $n_j$  em p,  $i \neq j \Rightarrow n_i \neq n_j$ .

A definição de caminho direcionado será usada para limitar o conjunto de construções de rotas que podem ser usadas. Essa definição permite a definição de  $C_{IP}$  (conjunto de conectores de um *inf-set* para um processo) e  $C_{PI}$  (conjunto de conectores de um processo para um *inf-set*).  $C_{IP}$  e  $C_{PI}$  divide o conjunto de conectores C. Baseado na função Q, o C é particionado em  $C_{AND}$  e  $C_{OR}$ . Os conjuntos  $C_j$  e  $C_s$  são usados para classificar os conectores em conectores *join* ou *split*.

**Definição 3.** Seja  $MPN-EKD = (I, P, C, Q, A)$  um:

- $N = I \cup P \cup C$  é um conjunto de nodos do MPN-EKD.
- $C_{AND} = \{c \in C \mid Q(c) = AND\}$ .
- $C_{OR} = \{c \in C \mid Q(c) = OR\}$ .
- Para  $n \in N$ :  $\bullet n = \{m \mid (m, n) \in A\}$  é o conjunto de nodos de entrada, e  $n \bullet = \{m \mid (n, m) \in A\}$  é um conjunto de nodos de saída.
- $C_j = \{c \in C \mid |\bullet c| \geq 2\}$  é um conjunto de conectores do tipo *join*.
- $C_s = \{c \in C \mid |c \bullet| \geq 2\}$  é um conjunto de conectores do tipo *split*.

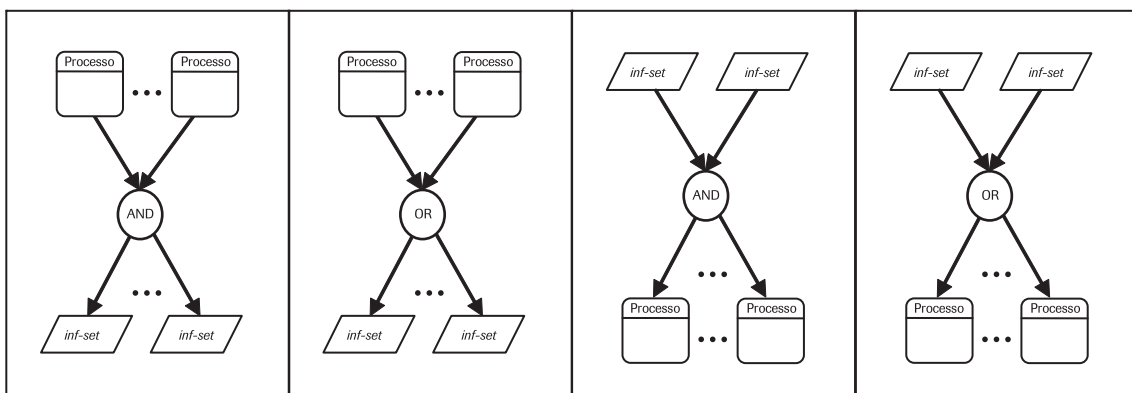
- $C_{IP} \subseteq C$  tal que  $c \in C_{IP}$  se, e somente se, existe um caminho  $p = \langle n_1, n_2, n_3 \rangle$ , tal que  $n_1 \in I, n_2 \in C, n_3 \in P$ .
- $C_{PI} \subseteq C$  tal que  $c \in C_{PI}$  se, e somente se, existe um caminho  $p = \langle n_1, n_2, n_3 \rangle$ , tal que  $n_1 \in P, n_2 \in C, n_3 \in I$ .

A definição 3 possibilita especificar requisitos adicionais que um Modelo de Processos de Negócios do EKD deveria satisfazer.

**Definição 4.** Um Modelo de Processo de Negócio do EKD satisfaz os seguintes requisitos:

- O conjunto I, P e C são conjuntos disjuntos, isto é,  $I \cap P = \emptyset, I \cap C = \emptyset, P \cap C = \emptyset$ .
- Para cada  $i \in I$ :  $|\bullet i| \leq 1$  e  $|i \bullet| \leq 1$ .
- Existe ao menos um *inf-set*  $i \in I$ , tal que  $|\bullet i| = 0$  (*inf-set* inicial).
- Existe ao menos um *inf-set*  $i \in I$ , tal que  $|i \bullet| = 0$  (*inf-set* final).
- Para cada  $p \in P$ :  $|\bullet p| = 1$  e  $|p \bullet| = 1$
- Para cada  $c \in C$ :  $|\bullet c| \geq 1$  e  $|c \bullet| \geq 1$ .
- O gráfico induzido pelo MPN-EKD é fracamente conexo, isto é, se para cada dois nodos  $n_1, n_2 \in N$ ,  $(n_1, n_2) \in (A \cup A^{-1})^*$ .
- $C_j$  e  $C_s$  é divisão de C, isto é  $C_j \cap C_s = \emptyset$  e  $C_j \cup C_s = C$ .
- $C_{IP}$  e  $C_{PI}$  é divisão de C, isto é  $C_{IP} \cap C_{PI} = \emptyset$  e  $C_{IP} \cup C_{PI} = C$

O primeiro requisito da definição 4 declara que cada componente tem um identificador único (nome). Os nomes dos conectores são omitidos no Modelo de Processos de Negócio do EKD. Os outros requisitos correspondem a restrições na relação A. *Inf-sets* não podem ter múltiplos arcos de entrada e deve existir ao menos um *inf-set* inicial e um *inf-set* final.



**Figura 1:** Construções que não são permitidas em um MPN-EKD formal.

Cada processo tem ao menos um *inf-set* inicial e um *inf-set* final, um arco de entrada e um arco de saída, para os dois nodos  $n_1$  e  $n_2$  (ignorando a direção dos arcos). Um conector  $c$  é um conector *join* ( $|\bullet c| = 1$  e  $|c \bullet| \geq 2$ ) ou *split* ( $|c \bullet| = 1$  e  $|\bullet c| \geq 2$ ). O último requisito declara que o conector  $c$  é um caminho de um *inf-set* para um processo ou um caminho de um processo para um *inf-set*. O Modelo de Processos de Negócio do EKD é sintaticamente correto se todos os requisitos declarados na definição 4 são satisfeitos.

#### 4. MAPEAMENTO DO MODELO DE PROCESSOS DE NEGÓCIO EM REDES DE PETRI

Nesta seção, será apresentado o procedimento de mapeamento do Modelo de Processos de Negócio em redes de Petri. O procedimento de mapeamento foi desenvolvido baseado em redes de Petri lugar/transição.

As definições (1) e (4) apresentadas apenas relatam a sintaxe de um modelo de Processos de Negócio do EKD e não a semântica.

Os lugares representam *inf-sets* ou são construções necessárias para modelar o comportamento do conector do MPN-EKD. As transições representam processos ou estão representando o comportamento do conector. Cada conector  $c \in C$  corresponde a lugares, transições e/ou arcos.

O conector pode corresponder a um número de arcos da rede de Petri ou uma pequena rede de lugares e transições. O conector OR corresponde ao comportamento de um lugar. O conector AND corresponde ao comportamento de uma transição. Na definição 5 o elemento Lugar de redes de Petri será representado por  $L$  para evitar confusão com o  $P$  de processo de MPN-EKD. A definição 5, apresentada a seguir, demonstra como é o mapeamento dos conectores do MPN-EKD desenvolvido neste trabalho.

No contexto do MPN-EKD os arcos sempre têm peso igual a 1 porque lugares correspondem a condições. Em uma rede de Petri que corresponde a um MPN-EKD correto (*sound*) um lugar nunca conterá múltiplas marcas. A rede é segura. Os estados com múltiplas marcas em um lugar são resultados de erros de projeto. Para capturar esses erros é necessário considerar redes não-seguras.

**Definição 5.** Considere um MPN-EKD =  $(I, P, C, Q, A)$ .  $N(EKD) = (L^{PN}, T^{PN}, F^{PN})$  é uma rede de Petri gerada pelo MPN-EKD:

$$L^{PN} = I \cup \left( \bigcup_{c \in C} L_c^{PN} \right)$$

O conjunto de lugares ( $L^{PN}$ ) é formado pela união de todos

os *inf-sets* com lugares que foram incluídos para representar conectores ( $\bigcup_{c \in C} L_c^{PN}$ ).

$$TPN = P \cup \left( \bigcup_{c \in C} T_c^{PN} \right)$$

O conjunto de transições ( $T^{PN}$ ) é formado pela união de todos os Processos com transições que foram incluídas para representar conectores ( $\bigcup_{c \in C} T_c^{PN}$ ).

$$F^{PN} = (A \cap ((I \times P) \cup (P \times I))) \cup \left( \bigcup_{c \in C} F_c^{PN} \right)$$

O conjunto de arcos da rede ( $F^{PN}$ ) é formado pelos arcos do modelo que vão de  $I$  a  $P$  e de  $P$  a  $I$  e a união dos arcos incluídos para representar conectores ( $\bigcup_{c \in C} F_c^{PN}$ ).

A seguir serão apresentadas as definições de  $L_c^{PN}$ ,  $T_c^{PN}$  e  $F_c^{PN}$  de acordo com as regras de mapeamento relacionadas ao tipo de conectores do MPN-EKD. Em seguida a cada definição são apresentados exemplos que representam as regras utilizadas para mapear os conectores do MPN-EKD em redes de Petri.

##### Regra 1

$$c \in C_{IP} \cap C_J \cap C_{AND}$$

Quando o conector  $c$  pertence a  $C_{IP}$  (caminho de *inf-set* para processo) intersecção de  $C_J$  (join) intersecção de  $C_{AND}$ , as definições de  $L_c^{PN}$ ,  $T_c^{PN}$  e  $F_c^{PN}$  são as seguintes:

- 1.1  $L_c^{PN} = \emptyset$
- 1.2  $T_c^{PN} = \emptyset$
- 1.3  $F_c^{PN} = \{(x, y) | x \in \bullet c \text{ e } y \in c \bullet\}$

A equação 1.1 determina que para representar esse conector não é necessário acrescentar lugares. A equação 1.2 determina que para representar esse conector não é necessário acrescentar transições. A equação 1.3 determina que os arcos que vão do conjunto de entrada do conector ao conjunto de saída do conector.

Nesse sentido, observa-se que o conector *AND-join* corresponde a dois ou mais arcos em redes de Petri se, e somente se, a saída é um processo. Na Figura 2 é apresentado um exemplo de mapeamento do conector  $c \in C_{IP} \cap C_J \cap C_{AND}$ .

##### Regra 2

$$c \in C_{PI} \cap C_J \cap C_{AND}$$

Quando o conector  $c$  pertence a  $C_{PI}$  (caminho de processo para *inf-set*) intersecção de  $C_J$  (join) intersecção de  $C_{AND}$ , as

definições de  $L_c^{PN}$ ,  $T_c^{PN}$  e  $F_c^{PN}$  são as seguintes:

$$2.1 \quad L_c^{PN} = \{l_x^c \mid x \in \bullet c\}$$

$$2.2 \quad T_c^{PN} = \{t^c\}$$

$$2.3 \quad F_c^{PN} = \{(x, l_x^c) \mid x \in \bullet c\} \cup \{(l_x^c, t^c) \mid x \in \bullet c\} \cup \{(t^c, x) \mid x \in c \bullet\}$$

Na equação 2.1 é determinado que para representar esse conector é necessário acrescentar um lugar para cada processo de entrada do conector. Na equação 2.2 é indicado que para representar esse conector é necessário acrescentar uma transição. Na equação 2.3 é determinado que para representar esse conector é necessário acrescentar arcos que ligam as transições aos lugares de entrada do conector, entre os lugares e a transição correspondente

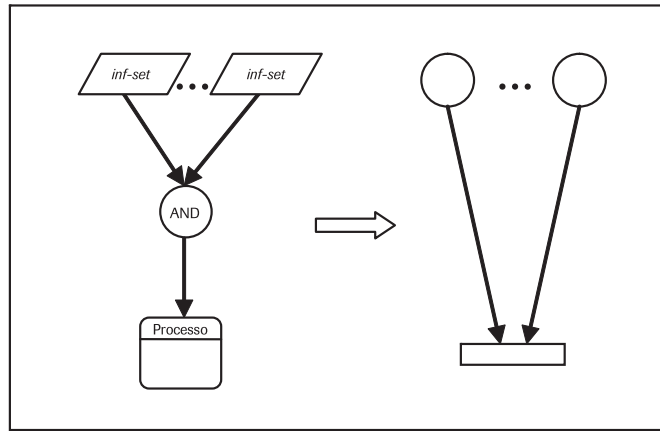
ao conector e entre a transição e o lugar de saída do conector.

Neste caso, o conector AND-join tem o comportamento de uma transição. É acrescentado um lugar para cada processo de entrada do conector. Na Figura 3 é apresentado um exemplo de mapeamento do conector  $c \in C_{PI} \cap C_J \cap C_{AND}$ .

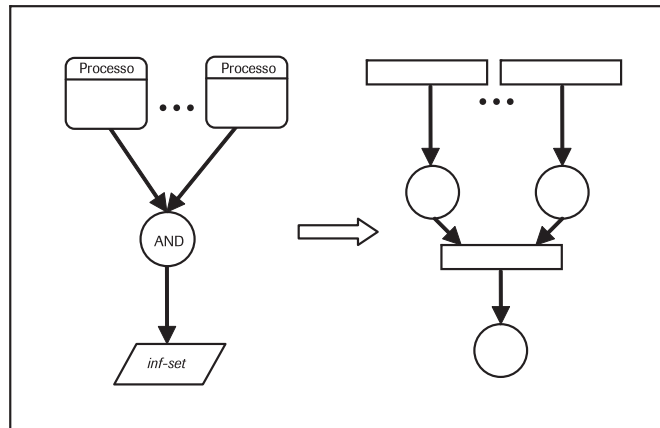
### Regra 3

$$c \in C_{IP} \cap C_J \cap C_{OR}$$

Quando o conector  $c$  pertence a  $C_{IP}$  (caminho de *inf-set* para processo) intersecção de  $C_J$  (join) intersecção de  $C_{OR}$ , as definições de  $L_c^{PN}$ ,  $T_c^{PN}$  e  $F_c^{PN}$  são as seguintes:



**Figura 2: Exemplo de Mapeamento do  $C_{AND}$  entre dois ou mais *inf-sets* para um processo.**



**Figura 3: Exemplo de Mapeamento do  $C_{AND}$  entre dois ou mais processos para um *inf-set*.**



- 3.1  $L_c^{PN} = \{l^c\}$   
 3.2  $T_c^{PN} = \{t_x^c \mid x \in \bullet c\}$   
 3.3  $F_c^{PN} = \{(x, t_x^c) \mid x \in \bullet c\} \cup \{(t_x^c, l^c) \mid x \in \bullet c\} \cup \{(l^c, x) \mid x \in c \bullet\}$

Na equação 3.1 é determinado que para representar este conector é necessário acrescentar um lugar. Na equação 3.2 é estabelecido que para representar este conector é necessário acrescentar uma transição para cada *inf-set* de entrada do conector. Na equação 3.3 é determinado que para representar este conector é necessário acrescentar um conjunto de arcos dos lugares às transições de entrada do conector, entre as transições e o lugar que corresponde ao conector e entre o lugar e a transição de saída do conector.

O conector *OR-join* tem o comportamento de um lugar quando o conector é  $C_{IP}$ . Na Figura 4 é apresentado um exemplo de mapeamento do conector  $c \in C_{IP} \cap C_J \cap C_{OR}$ .

#### Regra 4

$$c \in C_{PI} \cap C_J \cap C_{OR}$$

Quando o conector  $c$  pertence a  $C_{PI}$  (caminho de processo para *inf-set*) intersecção de  $C_J$  (join) intersecção de  $C_{OR}$ , as definições de  $L_c^{PN}$ ,  $T_c^{PN}$  e  $F_c^{PN}$  são as seguintes:

- 4.1  $L_c^{PN} = \emptyset$   
 4.2  $T_c^{PN} = \emptyset$   
 4.3  $F_c^{PN} = \{(x, y) \mid x \in \bullet c \text{ e } y \in c \bullet\}$

Na equação 4.1 é apresentado que não é necessário acrescentar lugares para representar esse conector. Na equação 4.2 é determinado que não é necessário acrescentar transições para representar esse conector. Na equação 4.3 é apresentado

que o conjunto de arcos está entre o conjunto de entrada e o conjunto de saída.

O conector *OR-join* corresponde a dois ou mais arcos em redes de Petri se, e somente se, o conector é  $C_{PI}$ . Na Figura 5 é apresentado um exemplo de mapeamento do conector  $c \in C_{IP} \cap C_J \cap C_{OR}$ .

#### Regra 5

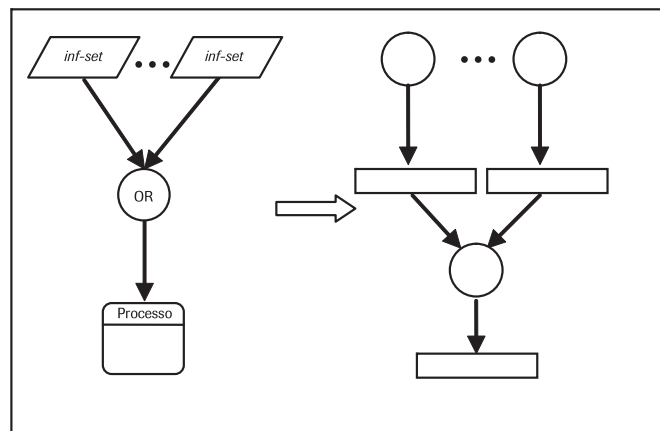
$$c \in C_{IP} \cap C_S \cap C_{AND}$$

Quando o conector  $c$  pertence a  $C_{IP}$  (caminho de *inf-set* para processo) intersecção de  $C_S$  (join) intersecção de  $C_{AND}$ , as definições de  $L_c^{PN}$ ,  $T_c^{PN}$  e  $F_c^{PN}$  são as seguintes:

- 5.1  $L_c^{PN} = \{l_x^c \mid x \in c \bullet\}$   
 5.2  $T_c^{PN} = \{tc\}$   
 5.3  $F_c^{PN} = \{(x, t^c) \mid x \in \bullet c\} \cup \{(t^c, l_x^c) \mid x \in c \bullet\} \cup \{(l_x^c, x) \mid x \in c \bullet\}$

Na equação 5.1 é apresentado que é necessário acrescentar um lugar para cada processo de saída do conector. Na equação 5.2 é determinado que para representar este conector é necessário acrescentar uma transição. Na equação 5.3 é apresentado que o conjunto de arcos necessários para representar este conector deve estar entre o lugar e a transição correspondente ao conector, entre a transição e os lugares de saída do conector e entre os lugares de saída do conector e a transição.

Dessa forma, o conector *AND-split* do tipo  $C_{IP}$  tem o comportamento de uma transição seguida de um número de lugares igual ao número de processos. Na Figura 6 é apresentado um exemplo de mapeamento do conector  $c \in C_{IP} \cap C_S \cap C_{AND}$ .



**Figura 4: Exemplo de Mapeamento  $C_{OR}$  entre dois ou mais *inf-sets* para um processo.**



### Regra 6

$$c \in C_{PI} \cap C_S \cap C_{AND}$$

Quando o conector  $c$  pertence a  $C_{IP}$  (caminho de *inf-set* para processo) intersecção de  $C_S$  (join) intersecção de  $C_{AND}$ , as definições de  $L_c^{PN}$ ,  $T_c^{PN}$  e  $F_c^{PN}$  são as seguintes:

- 6.1  $L_c^{PN} = \emptyset$
- 6.2  $T_c^{PN} = \emptyset$
- 6.3  $F_c^{PN} = \{(x, y) | x \in \bullet c \text{ e } y \in c \bullet\}$

Na equação 6.1 é apresentado que não é necessário acrescentar lugares para representar esse conector. Na equação 6.2 é determinado que não é necessário acrescentar transições para representar esse conector. Na equação 6.3 é apresentado que o conjunto de arcos está entre o conjunto de entrada e o conjunto de saída.

O conector AND-*split* corresponde a um número de arcos em redes de Petri se, e apenas se, a saída são dois ou mais *inf-sets*. Na Figura 7 é apresentado um exemplo de mapeamento do conector  $c \in C_{PI} \cap C_S \cap C_{AND}$ .

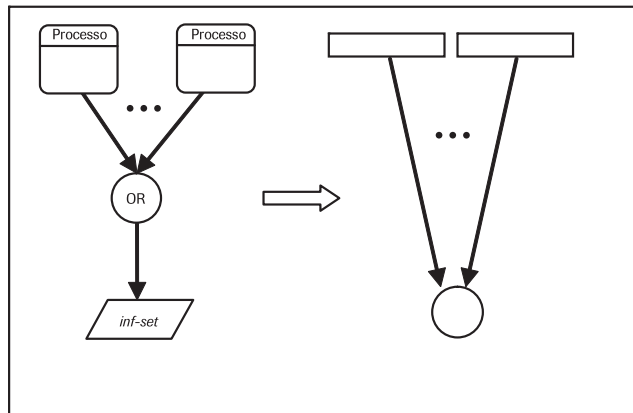
### Regra 7

$$c \in C_{PI} \cap C_S \cap C_{OR}$$

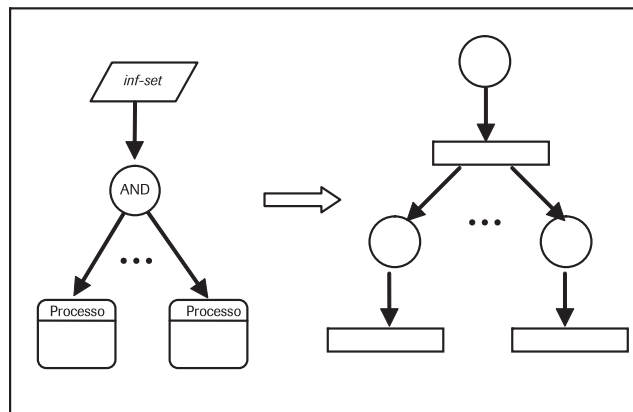
Quando o conector  $c$  pertence a  $C_{IP}$  (caminho de *inf-set* para processo) intersecção de  $C_S$  (join) intersecção de  $C_{OR}$ , as definições de  $L_c^{PN}$ ,  $T_c^{PN}$  e  $F_c^{PN}$  são as seguintes:

- 7.1  $L_c^{PN} = \emptyset$
- 7.2  $T_c^{PN} = \emptyset$
- 7.3  $F_c^{PN} = \{(x, y) | x \in \bullet c \text{ e } y \in c \bullet\}$

Na equação 7.1 é apresentado que não é necessário acrescentar lugares para representar esse conector. Na equação 7.2



**Figura 5: Exemplo de Mapeamento COR entre dois ou mais processos para um *inf-set*.**



**Figura 6: Exemplo de Mapeamento do  $C_{AND}$  um *inf-set* para dois ou mais processos.**

é determinado que não é necessário acrescentar transições para representar esse conector. Na equação 7.3 é apresentado que o conjunto de arcos está entre o conjunto de entrada e o conjunto de saída.

O conector *OR-split* corresponde a um número de arcos em redes de Petri se, e somente se, a saída são dois ou mais processos. Na Figura 8 é apresentado um exemplo de mapeamento do conector  $c \in C_{IP} \cap C_s \cap C_{OR}$ .

#### Regra 8

$$c \in C_{IP} \cap C_s \cap C_{OR}.$$

Quando o conector  $c$  pertence a  $C_{PI}$  (caminho de processo para *inf-set*) intersecção de  $C_s$  (join) intersecção de  $C_{OR}$ , as

definições de  $L_c^{PN}$ ,  $T_c^{PN}$  e  $F_c^{PN}$  são as seguintes:

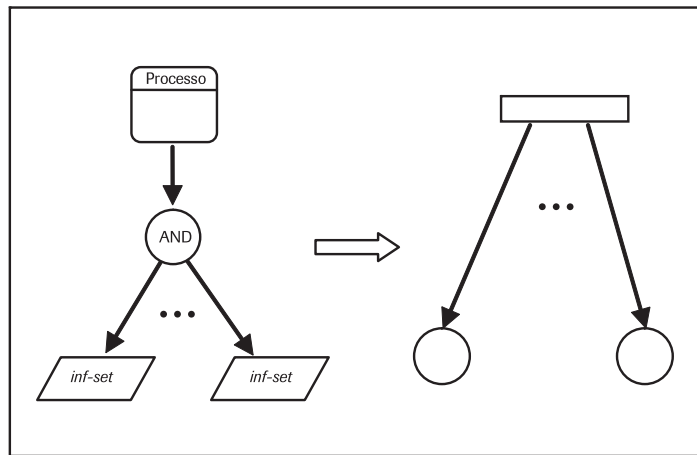
$$8.1 \quad L_c^{PN} = \{l^c\}$$

$$8.2 \quad T_c^{PN} = \{t_x^c \mid x \in c \bullet\}$$

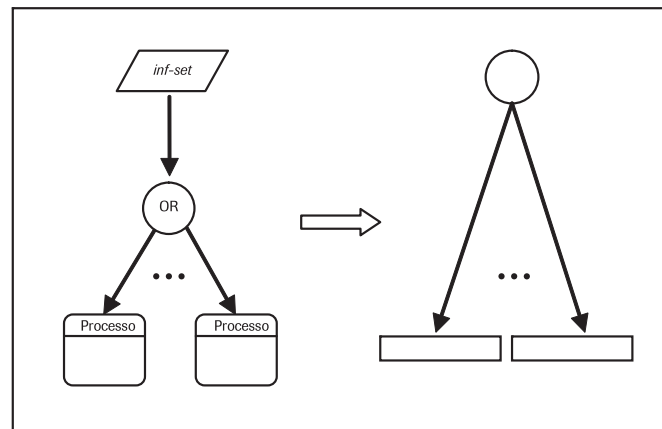
$$8.3 \quad F_c^{PN} = \{(x, l^c) \mid x \in c \bullet\} \cup \{(l^c, t_x^c) \mid x \in c \bullet\} \cup \{(t_x^c, x) \mid x \in c \bullet\}$$

Na equação 8.1 é determinado que para representar este conector é necessário acrescentar um lugar. Na equação 8.2 é apresentado que é necessário acrescentar uma transição para cada *inf-set* de saída do conector. Na equação 8.3 é determinado que o conjunto de arcos deve estar entre a transição inicial e o lugar correspondente ao conector, entre o lugar e as transições de saída do conector e entre as transições de saída do conector e os lugares.

A única forma de existir um lugar com mais de um arco



**Figura 7: Exemplo de Mapeamento do  $C_{AND}$  de um processo para dois ou mais *inf-sets*.**



**Figura 8: Exemplo de Mapeamento do  $C_{OR}$  de um *inf-sets* para dois ou mais processos.**

de saída é o mapeamento do conector *OR-split* de um processo para dois ou mais *inf-sets*. As regras de mapeamento garantem que o número de transições é igual ao número de lugares. A rede é escolha-livre. Na Figura 9 é apresentado um exemplo de mapeamento do conector  $c \in C_{PI} \cap C_s \cap C_{OR}$ .

O MPN-EKD (I,P,C,Q,A) é um Modelo de Processos de Negócio do EKD e  $PN=N(MPN-EKD)$  a rede de Petri gerada pelo MPN-EKD. PN é escolha-livre.

**Definição 6.** Um MPN-EKD é regular se e somente se:

- MPN-EKD tem dois *inf-sets* especiais:  $i_{início}$  e  $i_{final}$ . *inf-set*  $i_{início}$  é um nodo fonte:  $\bullet i_{início} = \emptyset$ . *inf-set*  $i_{final}$  é um nodo final:  $i_{final} \bullet = \emptyset$ .
- Todo nodo  $n \in N$  está no caminho do  $i_{início}$  para o  $i_{final}$ .

A identificação do  $i_{início}$  e do  $i_{final}$  permite uma definição clara do estado inicial e do estado final. Um MPN-EKD com múltiplos *inf-sets* iniciais (por exemplo, *inf-sets* sem arcos iniciais) ou múltiplos *inf-sets* finais (por exemplo, *inf-sets* sem arcos finais) podem facilmente ser estendidos com uma parte de inicialização ou finalização que o primeiro requisito é satisfeito. O segundo requisito precisa que todo *inf-set* ou processo esteja entre  $i_{início}$  e  $i_{final}$ . Se o segundo requisito não é satisfeito, então o MPN-EKD é:

- i. Composto de partes completamente disjuntas.
- ii. Tem partes que nunca são ativadas.
- iii. Partes do MPN-EKD formam armadilhas.

Como o MPN-EKD descreve o ciclo de vida de um caso (por exemplo, uma instância de processo), os dois requisitos são razoáveis. O ciclo de vida deveria ter um evento inicial e um evento final, e todos os passos deveriam estar em um caminho entre esses dois eventos. No presente trabalho será considerado MPN-EKD regulares.

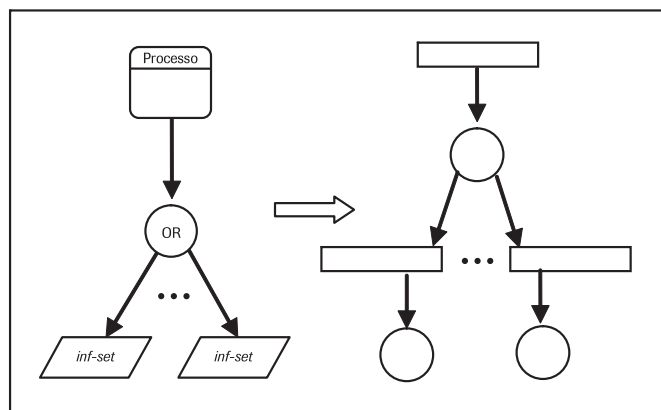
Um MPN-EKD descreve um procedimento com um estado inicial e um estado final. O procedimento deveria ser projetado de tal forma que sempre termine corretamente. Além disso, deveria ser possível executar qualquer processo seguindo a rota apropriada do MPN-EKD.

**Definição 7.** Um MPN-EKD regular é *sound* se e somente se:

1. Para cada marcação M alcançável do estado inicial (por exemplo, o estado, onde o *inf-set*  $i_{início}$  é o único *inf-set* que existe), existe uma sequência de disparos levando da marcação M para a marcação final (por exemplo, onde o *inf-set*  $i_{final}$  é o único *inf-set* que existe).
2. A única marca existente no final do processo está no estado final  $i_{final}$ .
3. Não existir processos *dead*, por exemplo, para cada processo p existe uma sequência disparável, a qual executa p.

O critério de corretude definido em redes de Petri é o requisito mínimo que qualquer MPN-EKD deveria satisfazer. Um MPN-EKD *sound* é livre de potenciais *deadlocks* e *livelocks*. Se assume *fairness*, então os primeiros dois requisitos implica que eventualmente a marcação final será alcançada (nota-se que esse é um resultado da combinação da propriedade de *soundness* e de escolha-livre). A propriedade escolha-livre implica que para cada transição  $t_1$  e  $t_2$ ,  $\bullet t_1 \cap \bullet t_2 \neq \emptyset$  que implica que  $\bullet t = \bullet t_2$ . Mais detalhes dessas propriedades e critérios podem ser consultados em Aalst (1999), Aalst e Hee (2002) e Verbeek et al. (2007).

Os MPN-EKD complexos encontrados na prática não são fáceis de verificar a propriedade *soundness*. Felizmente, técnicas e ferramentas baseadas em redes de Petri podem ser usadas para analisar essa propriedade. A inspeção da árvore



**Figura 9:** Exemplo de Mapeamento do  $C_{OR}$  de um processo para dois ou mais *inf-sets*.

de cobertura da rede de Petri, que corresponde ao MPN-EKD é suficiente para verificar *soundness*. Para MPN-EKD complexos, o gráfico de cobertura pode tornar-se muito grande. Esse fenômeno é conhecido como o “problema de explosão de estado”. Um MPN-EKD com 80 processos pode facilmente ter mais de 200.000 marcações. Embora os computadores atualmente tenham dificuldade para analisar gráficos de cobertura desse tamanho, existem muitas técnicas avançadas que exploram a estrutura de redes de Petri, nesse caso gerada por um MPN-EKD. Essas técnicas permitem

## O formalismo de redes de Petri a torna uma poderosa técnica de modelagem para a representação de processos.

eficientes procedimentos de decisão. Antes de apresentar tal procedimento, é necessário primeiramente listar algumas propriedades presentes em qualquer rede de Petri gerada por um MPN-EKD *sound*.

Considerando que o  $MPN-EKD = (I, P, C, Q, A)$  é *sound* e  $PN = N$  (MPN-EKD) a rede de Petri gerada pelo MPN-EKD. Considere  $PN$  ser  $\overline{PN}$  com uma transição  $t$  adicional conectando  $i_{final}$  para  $i_{início}$  e deixar  $M$  ser a marcação inicial com uma marca em  $i_{início}$ .

- $\overline{PN}$  é fortemente conexa;
- $\overline{PN}$  é passível de cobertura.
- $(\overline{PN}, M)$  é viva.
- $(\overline{PN}, M)$  é limitada.

Uma rede de Petri é fortemente conexa se, e somente se, para cada par de nodos (lugares e transições)  $x$  e  $y$ , existe um caminho de  $x$  até  $y$  (AALST, 1997).

$PN$  é fortemente conexa porque todos os nodos estão no caminho do  $i_{início}$  para o  $i_{final}$  e  $i_{final}$  é conectado em  $i_{início}$  via  $t$  adicional.  $PN$  é WF-net de acordo com Aalst (1997). Portanto, *soundness* coincide com vivacidade e limitação.

$(PN, M)$  é escolha-livre, viva e limitada e, de acordo com Aalst e Hee (2002), implica que  $PN$  é passível de cobertura e  $(PN, M)$  é segura. Construindo os resultados apresentados em Aalst (1997), a propriedade de *soundness* pode ser verificada em tempo polinomial.

Um MPN-EKD corresponde a uma rede WF-net escolha-livre. Uma rede WF é *sound* se e somente se a rede estendida é viva e limitada. Vivacidade e limitação podem ser decididos em tempo polinomial (AALST, 1997). Por essa razão *soundness* pode ser verificado em tempo polinomial.

Nesse sentido, é possível estender ferramentas com pro-

cedimentos eficientes de decisão para verificar a propriedade *soundness* de um MPN-EKD. Para guiar o usuário em procurar e corrigir defeitos em um projeto de um MPN-EKD, é possível também suprir diagnósticos adicionais baseados na estrutura do MPN-EKD/redes de Petri.

## 5. CONSIDERAÇÕES FINAIS

Neste trabalho, foi destacado que o principal problema das abordagens de Modelagem Organizacional, incluindo-se o EKD, é a ausência de técnicas de análise objetivas. As técnicas de análise com rigor matemático não são usuais para profissionais da área de negócio. A revisão da literatura mostrou que as redes de Petri resolvem esse problema, uma vez que elas possuem representação gráfica, são de fácil aprendizado, funcionam como linguagem de comunicação entre especialistas de diversas áreas, permitem a descrição dos aspectos estáticos e dinâmicos do sistema a ser representado, e ainda possuem o formalismo matemático que permite a utilização de importantes métodos de análise.

Os processos de negócio, normalmente, têm uma estrutura simples antes de serem introduzidos nos sistemas de informação avançados, tais como sistema de *workflow*. Essa simplicidade é devida principalmente ao fato de que um documento (papel) pode estar apenas em um lugar em um mesmo momento. O documento atua como um conjunto de marcas que asseguram a execução seqüencial das tarefas. Atualmente, após vários anos de desenvolvimento de sistemas de forma seqüencial, é possível modelar processos de forma completamente diferente, uma vez que as informações e os dados podem ser compartilhados. Várias pessoas podem trabalhar ao mesmo tempo no mesmo caso. Por essa razão nem sempre é possível que as tarefas sejam realizadas seqüencialmente. Graças à utilização de processos de negócio paralelos é possível conseguir enormes reduções no tempo de execução. O ambiente de negócio é propício para executar as tarefas em paralelo de acordo com a necessidade. Contudo o uso de rotas seqüencial, paralela, seletiva e iterativa no mesmo processo torna difícil a avaliação dos processos definidos.

Neste sentido, a literatura mostrou que o Modelo de Processos de Negócio deve ser desenvolvido com muito cuidado, pois, além de os problemas resultantes de erros no projeto serem difíceis de detectar, os custos da correção dos erros são altos. As ambigüidades e conflitos devem ser eliminados dos modelos.

Este trabalho mostrou que ambigüidades e confusões

não podem ser prevenidas em um modelo de Processos de Negócio informal. Para resolver este problema, foi desenvolvido o Modelo de Processos de Negócio com uma semântica formal. Para desenvolver essa semântica foi criado um conjunto de conectores para o Modelo de Processos de Negócio do EKD. O conjunto de conectores é representado por  $C$  e é composto por  $C_{AND}$ ,  $C_{OR}$ ,  $C_J$ ,  $C_S$ ,  $C_{IP}$  e  $C_{PI}$ . Os conectores  $C_{OR}$  e  $C_{AND}$  são importantes para identificar escolha (exclusiva) e paralelismo para que os casos de paralelismo e escolha não sejam modelados exatamente da mesma forma, evitando ambigüidades e dificuldades de compreensão. Os conectores  $C_J$  e  $C_S$  definem conectores do tipo *join* e *split*. Os conectores  $C_{IP}$  e  $C_{PI}$  demonstram que um conector  $c$  é um caminho de um *inf-set* para um processo ou um caminho de um processo para um *inf-set*.

Foram incluídos os estados inicial e final para possibilitar que a formalização fosse efetivamente realizada. Esses estados não são especificados no Modelo de Processos de Negócio do EKD original.

Foi desenvolvido um procedimento de mapeamento formal do Modelo de Processos de Negócio em redes de Petri. O procedimento de mapeamento foi desenvolvido baseado em redes de Petri lugar/transição. Por meio de um modelo de processos de negócio mapeado em redes de Petri de acordo

com este procedimento, foi possível verificar alguns requisitos que garantem se o processo foi modelado corretamente e outros requisitos que permitem uma análise do processo.

Nesse sentido, como trabalho futuro, a partir do procedimento de mapeamento formal do Modelo de Processos de Negócio em redes de Petri será desenvolvida a primeira versão do Método de Avaliação do Modelo de Processos de Negócio do EKD.

A grande conveniência, como afirmado anteriormente, no uso de redes de Petri na modelagem de processos de negócios é a possibilidade de um rastreamento minucioso e não-ambíguo de cada etapa da operação.

Além disso, este trabalho mostra que as redes de Petri possibilitam uma representação matemática formal e disponibilizam mecanismos de análise que tornam possíveis a verificação da correção do modelo e a checagem de suas propriedades.

O fato de algumas construções não serem permitidas pode ser considerado uma desvantagem da formalização do MPN-EKD. Porém, durante o processo de modelagem essas construções devem ser analisadas cuidadosamente, sendo importante o discernimento da equipe ou pessoa que está modelando para que o modelo seja desenvolvido de acordo com as definições criadas neste trabalho.

**Artigo recebido em 15/08/2007**

**Aprovado para publicação em 26/03/2008**

## REFERÊNCIAS

- AALST, W. M. P. V. D. Formalization and verification of event-driven process chains. *Information and Software Technology*, London, v. 41, n. 10, p. 639-650, July, 1999.
- \_\_\_\_\_. Verification of workflow nets. In: AZEMA, P.; BALBO, G. (Ed.). *Application and theory of petri nets*. Berlin: Springer-Verlag. (Lecture Notes in Computer Science). 1997.
- \_\_\_\_\_. The Application of petri nets to workflow management. *The Journal of Circuits, Systems and Computers*, Singapore, v. 8, n. 1, p. 21-68, 1998.
- \_\_\_\_\_. Workflow Verification: Finding control-flow errors petri-net-based techniques. In: AALST, V.D.W.; DESEL, J.; OBERWEIS, A. *Business process management: model, techniques, and empirical studies*. Berlin: Springer-Verlag. p. 161-183. (Lecture Notes in Computer Science, 1806), 2000.
- AALST, W. M. P. V. D.; HOFSTEDE, A.H.M.T. Verification of workflow task structures a petri-net-based approach. *Information Systems*, Oxford, v. 25, n. 1, p. 43-69, 2000.
- AALST, W. V. D.; HEE, V. K. *Workflow management: models, methods and systems*. Cambridge: MIT Press, 2002.
- ADAM, N. R.; ATLURI, V.; HUANG, W. K. Modeling and analysis of workflows using petri nets. *Journal of Intelligent Information Systems*, Norwell, v. 10, n. 2, p. 1-29. Special Issue on Workflow and Process Management, 1998.
- BADOUEL, E.; OLIVER, J. *Reconfigurable nets, a class of high level petri nets supporting dynamic changes within workflow systems*. Publication Interne IRISA PI 1163, 1998.
- BERIO, G.; PETIT, M. Enterprise modelling and UML: (sometimes) a conflict without a case. In: ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING: research and applications, 10., *Proceedings*. Madeira Island, 2003.
- BUBENKO JR., J. A.; STIRNA, J.; BRASH, D. *EKD user guide, Dpt of computer and systems sciences*. Stockholm: Royal Institute of Technology. 1998.
- CHEN-BURGER, Y. Knowledge sharing and inconsistency checking on multiple enterprise models. In: IJCAR INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, Sienna. *Automated reasoning*:

*Proceedings.* Berlin: Springer. (Lecture Notes in Computer Science, 2083), 2001.

CHEN-BURGER, Y.; STADER, J. Formal support for adaptive workflow systems in a distributed environment. In: FISCHER, I.; LAYNA, M. (Ed.). *Workflow handbook 2003*. Lighthouse Point, FL, USA: Future Strategies. (Workflow Management Coalition), 2003.

CHEN-BURGER, Y.; TATE, A.; ROBERTSON, D. Enterprise modelling: a declarative approach for FBPM. In: EUROPEAN CONFERENCE OF ARTIFICIAL INTELLIGENCE, KNOWLEDGE MANAGEMENT AND ORGANISATIONAL MEMORIES WORKSHOP, *Proceedings.* Lyon. 2002.

CHYZASTOWSKI-WACHTEL, P. et al. A top-down petri net-based approach for dynamic workflow modeling. In: INTERNATIONAL CONFERENCE BPM. Eindhoven. *Proceedings.* Berlin: Springer. p. 336-353. (Lecture Notes in Computer Science, 2678), 2003.

CINDIO, F. et al. *CHAOS: a knowledge-based system for conversing within offices, office knowledge representation, management and utilization.* Lamersdorf: W.Elsevier Science; B.V.North-Holland, p. 257-275. 1998.

DALLAVALLE, S. I. CAZARINI, E. W. Modelagem Organizacional Desenvolvimento do Conhecimento Organizacional, Facilitador de Desenvolvimento de Sistemas de Informação. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 11. Salvador, Anais. Salvador: [s.n.]. CD-ROM. 2001

DERHNERT, J. Four steps towards sound business process models. In: EHRIG., H. et al. (Ed.). *Petri net technology for communication-based systems- advances in petri nets.* Berlin : Springer. p. 66-82. (Lecture Notes in Computer Science, 2472). 2003.

DELLAROCAS, C.; KLEINS, M. A knowledge-based approach for designing robust business processes. In: VAN DER AALST, W.; DESEL, J.; OBERWEIS, A. (Ed.). *Business process management: models, techniques, and empirical studies.* Berlin: Springer. p. 50-65. (Lecture Notes in Computer Sciences, 1806). 2000.

DESEL, J.; ERWIN, T. Modeling simulation and analysis of business process. In: AALST,V.D.W.; DESEL, J.; OBERWEIS, A. *Business process management: models, techniques, and em-*

*pirical studies.* Berlin: Springer. p. 129-141. (Lecture Notes in Computer Sciences, 1806). 2000.

DONGEN, V. B. F et al. Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants. *Computers in Industry.* v. 58 , n. 6, p. 578-601, 2007.

ELLIS, C. A.; NUTT, G. J. Workflow: the process spectrum. In: NSF WORKSHOP ON WORKFLOW AND PROCESS AUTOMATION IN INFORMATION SYSTEM: state-of-the-art and future directions, 1996, Athens. *Proceedings.* Arlington: NSP. p. 140-145. 1996.

ESHUIS, R.; DEHNERT, J. Reactive petri nets for workflow modeling. In: INTERNATIONAL CONFERENCE ON THE APPLICATIONS AND THEORY OF PETRI NETS, 24. *Proceedings.* Berlin: Springer. p. 296-315. (Lecture Notes in Computer Science, 2679). 2003

FERSCHA, A. Qualitative and quantitative analysis of business workflows using generalized stochastic petri nets. In: WORKFLOW MANAGEMENT - challenges, paradigms and products, 1994, Linz. *Proceedings.* Oldenbourg: Verlag. p. 222-234. 1994.

GRIGOROVA, K. Process modelling using petri nets. In: INTERNATIONAL CONFERENCE ON COMPUTER SYSTEMS AND TECHNOLOGIES: e-learning, 4., 2003, Rouse. p.95-100. Disponível em: <<http://doi.acm.org/10.1145/973620.973636>>. Acesso em: 14 June. 2004.

GUAN, F. et al. Grid-Flow: a Grid-enabled scientific workflow system with a Petri-net-based interface. *Concurrency and Computation: Practice and Experience*, v. 18, n. 10, p. 1115-1140. 2006.

HEE, K. M.; REIJERS, H. A. Using formal analysis techniques in business process redesign. In: VAN DER AALST, W.; DESEL, J.; OBERWEIS, A. (Ed.). *Business process management: models, techniques, and empirical studies.* Berlin: Springer. p. 142-160. (Lecture Notes in Computer Sciences, 1806). 2000.

HEE, K. M. V.; REIJERS, H. M. Using formal analysis techniques in business process redesign. In: AALST,V.D.W.; DESEL, J.; OBERWEIS, A. *Business process management: models, techniques, and empirical studies.* Berlin: Springer. p. 142-160. (Lecture Notes in Computer Sciences, 1806). 2000.

HOLT, A. P. Management-oriented models of business process. In: AALST, V. D. W.; DESEL, J.; OBERWEIS, A. *Business process management: models, techniques, and empirical studies.* Berlin: Springer. p. 99-109. (Lecture Notes in Computer Sciences, 1806). 2000.

JANSSENS, G. K.; VERELST, J.; WEYN, B. Techniques for modeling workflows their support of reuse. In: AALST, V. D. W.; DESEL, J.; OBERWEIS, A. *Business process management: models, techniques, and empirical studies.* Berlin: Springer. p. 1-15. (Lecture Notes in Computer Sciences, 1806). 2000.

JONKERS, H. et al. Towards a language for coherent enterprise architecture descriptions. In: *IEEE International Enterprise Distributed Object Computing Conference*, 7., 2003, Brisbane. *Proceedings.* Los Alamitos: IEEE Computer Society. 2003a

\_\_\_\_\_. Concepts for modelling enterprise architectures. In: LANDELUK ARCHITECTUT CONGRESS, 2003. Disponível em: <[http://www.serc.nl/lac/2003/ontvangen\\_papers.htm](http://www.serc.nl/lac/2003/ontvangen_papers.htm)>. Acesso em: 16 Apr. 2003

JUNGINGER, S. et al. Building complex workflow applications: how to overcome the limitations for the waterfall model. In: FISCHER, L. (Ed.). *Workflow Management Coalition: the workflow handbook 2001.* Disponível em: <[http://www.boc-eu.com/english/papers/Complex\\_Workflow\\_Appl.pdf](http://www.boc-eu.com/english/papers/Complex_Workflow_Appl.pdf)>. Acesso em: 16 Apr. 2001.

KIRITSIS, D.; NEUENDORF, K.; XIROUCHAKIS, P. Petri net techniques for process planning and cost estimation. *Advances in Engineering Software*, Oxford, v. 30, p. 375-387, June, 1999.

KNOLMAYER, G. et al. Modeling processes and workflows by business rules. In: VAN DER AALST, W.; DESEL, J.; OBERWEIS, A. (Ed.). *Business process management: models, techniques, and empirical studies.* Berlin: Springer. p.16-29. (Lecture Notes in Computer Sciences, 1806). 2000.

KOUBARAKIS, M.; PLEXOUSAKIS, D. A formal framework for business process modelling and design. *Information Systems*, Oxford, v. 27, n. 5, p. 299-319, 2002

KRUCHTEN, P. *The rational unified process.* 2. ed. Harlow: Addison-Wesley. 2000.



LENZ, K.; MEVIUS, M.; OBERWEIS, A. Process-Oriented Business Performance Management with Petri Nets. In: IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05), 2005, p. 89-92, 2005.

LI, J. et al. MAS: a declarative formalism for hierarchical representation of procedural knowledge. *IEEE Transaction on Knowledge and Data Engineering*, New York, v. 6, n. 4, p. 639-643, 1994.

McDERMID, D. C. Integrated business process management: using state-based business rules to communicate between disparate stakeholders. In: INTERNATIONAL CONFERENCE BPM 2003, 2003, Eindhoven. *Proceedings*. Berlin: Springer. p. 58-71. (Lecture Notes in Computer Science, 2678). 2003.

MERZ, M. et al. Workflow modeling and execution with coloured petri net in COSM. In: WORKSHOP ON APPLICATIONS OF PETRI NET TO PROTOCOLS WITHIN THE INTERNATIONAL CONFERENCE ON APPLICATION AND THEORY OF PETRI NET, 16., 1995, Turin. *Proceedings*. Berlin: Springer. p. 1-12, 1995.

MEVIUS, M. OBERWEIS, A. A Petri-Net Based Approach to Performance Management of Collaborative Business Processes. In: 16TH INTERNATIONAL WORKSHOP ON DATABASE AND EXPERT SYSTEMS APPLICATIONS (DEXA'05). University of Karlsruhe, University of Karlsruhe. 987-991, 2005.

MOLD, D.; VALK, R. Object oriented petri net in business process modeling. In: AALST, V. D. W.; DESEL, J.; OBERWEIS, A. *Business process management: models, techniques, and empirical studies*. Berlin: Springer. p. 254-273. (Lecture Notes in Computer Sciences, 1806), 2000.

NEIGER, D.; CHURILOV, L. Structuring business objectives: a business process modeling perspective. In: INTERNATIONAL CONFERENCE BPM 2003, 2003, Eindhoven. *Proceedings*. Berlin: Springer. p. 72-87. (Lecture Notes in Computer Science, 2678), 2003.

NURCAN, A.; ROLLAND, C. A multi-method for defining the organizational change. *Journal of Information and Software Technology*, London, v. 45, n. 2, p. 61-82, 2003.

NURCAN, S. Analysis and design of co-operative work process a framework. *Information and Software Technology*. London, v. 40, n. 3, p. 143-156, 1998.

NURCAN, S.; BARRIOS, J. Enterprise knowledge and information system modelling in na evolving enviroment. In INTERNATIONAL WORKSHOP ON ENGINEERING METHODS TO SUPPRTED INFORMATION SYSTEMS EVOLUTION IN CONJUNCTION WITH, *Proceedings*. Geneva, 2003.

NUTT, G. J. The Evolution towards. *Flexible workflow systems. Distributed Systems Engineering*, v. 3, n. 4, p. 276-294, 1996.

OBERWEIS, A. An Integrated approach for the specification of processes and related complex structured objects in business applications. *Decision Support Systems*, Amsterdam, v. 17, n. 1, p. 31-53, 1996.

OBERWEIS, A. et al.. INCOME/WF-A petri-net based approach to workflow Management. In: KRALLMANN, H. (Ed.). *Wirtschaftsinformatik'97*. Berlin : Springer-Verlag. p. 557-580, 1997.

OU-YANG, C.; LIN Y. D. BPMN-based business process model feasibility analysis: a petri net approach. *International Journal of Production Research*. v. 45, n. 12, 2007.

PÁDUA, S.I.D. *Investigação do processo de desenvolvimento de software a partir da modelagem organizacional, enfatizando regras do negócio*. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 145 p., 2001.

PÁDUA, S. I. D. *Método de Avaliação do modelo de processos de negócios*. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 252 p., 2004.

PÁDUA, S. I. D.; CAZARINI, E. W.; INAMASU, R. Y. Modelagem Organizacional: captura dos requisitos organizacionais no desenvolvimento de sistemas de informação. *Revista Gestão e Produção*. v. 11, n. 2, p. 1-20. 2004.

PÁDUA, S. I. D.; SILVA, A. R. Y.; INAMASU, R. Y.; PORTO, A. J. V. Aplicações e Potencial das Redes de Petri na Engenharia de Produção. In: Simpósio de Engenharia de Produção, 10, Bauru, UNESP, 2003. *Anais*. Available from WWW: URL: <http://www.bauru.unesp.br/acontece/simpep.html> [30 Nov. 2003].

PÁDUA, S. I. D.; SILVA, A. R. Y.; INAMASU, R. Y.; PORTO, A. J. V. O potencial das redes de Petri em modelagem e análise de processos de negócios. *Revista Gestão e Produção*. v.11, n. 1, p. 1-11 2004.

PÁDUA, S. I. D.; BISPO, C. A. F. Sistema de Gerenciamento de Workflow: um overview e um estudo de caso. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 13. Ouro Preto, *Anais*. Ouro Preto: [s.n.]. CD-ROM, 2003.

PÁDUA, S. I. D.; SILVA, A. R. Y.; INAMASU, R. Y.; PORTO, A. J. V. Redes de petri aplicadas aos sistemas de gerenciamento de Workflow (CD-ROM). In: Encontro Nacional de Engenharia de Produção, 12. Curitiba, *Anais*. Curitiba: [s.n.]. CD-ROM, 2002.

PERSSON, A. The utility of participative enterprise modelling in IS development - challenges and research issues. In: INTERNATIONAL WORKSHOP ON THE REQUIREMENTS ENGINEERING PROCESS, 2., 2000, Greenwich. *Proceedings*. Berlin: Springer. p. 978-982, 2000.

RINDERLE, S.; REICHERT, M.; DADAM, P. Evaluation of correctness criteria for dynamic workflow changes. In: INTERNATIONAL CONFERENCE BPM 2003, 2003, Eindhoven. *Proceedings*. Berlin: Springer. p. 41-57. (Lecture Notes in Computer Science, 2678), 2003.

ROLLAND, C.; PRAKASH, N.; KAABI, R. S. Variability in Business Process Families. *Information Resources Management Association (IRMA)*, 2007.

ROLLAND, C.; SOUVEYET, C.; AYED, M. B. Guiding Lyee User Requirements Capture. *Knowledge Based System Journal (KBSJ)*, Intention and Software Process, v. 16, n. 7, p. 351-359, 2003.

ROLLAND, C.; NURCAN, S.; GROSZ, G. A Decision making pattern for guiding the enterprise knowledge development process. *Journal of Information and Software Technology*, v. 42, n. 5, p. 313-331, 2000.

SALIMIFARD, S.; WRIGHT M. Petri net based modelling of workflow systems: an overview. *European Journal of Operational Research*, Amsterdam, v. 134, n. 3, p. 664-676, 2001.



SANTANDER, V. F. A. *Integrando modelagem organizacional com modelagem funcional.* Tese (Doutorado) - Universidade Federal de Pernambuco, Recife, 210 p., 2002.

SCHÖMIG, A. K.; RAU, H. A Petri net approach for the performance analysis of business processes. Würzburg: University of Würzburg. Report n. 116 In: *Seminar at IBFI*, Schloss Dagstuhl, 1995.

TAYLOR, R.; TOFTS, C. *Modelling, myth vs reality, map vs territory.* Technical Reports HPL-2003-246 20031211 External. Disponível em: <<http://www.hpl.hp.com/techreports/2003/HPL-2003-246.html>>. Acesso em: 16 Apr, 2004.

VERBEEK H. M. W.; AALST, W. M. P.; HOFSTEDE A. H. M. Verifying Workflows with Cancellation

Regions and OR-joins: An Approach Based on Relaxed Soundness and Invariants. *The Computer Journal Advance*. Access published online on January 26, 2007.

VERBEEK, H. M. W.; BASTEN, T.; AALST, W. M. P. *Diagnosing workflow using woflan.* Eindhoven: Eindhoven University of Technology. BETA Working Paper Series, WP 48, 2002.

VOORHOEVE, M.. Compositional modeling and verification of workflow process. In: AALST, V. D. W.; DESEL, J.; OBERWEIS, A. *Business process management: models, techniques, and empirical studies.* Berlin: Springer. p.184-200. (Lecture Notes in Computer Sciences, 1806), 2000.

WIKARSKI, D. *An Introduction to modular process net.* Berkeley: International Computer Science Institute. Technical Report TR-96-019. 1996.

WORKFLOW Management coliation: reference model. Document Number WFMC-TC00-1003 Document status - Issue, 1996

WORKFLOW Management coliation: terminology and glossary. Document Number WFMC-TC-1011 Document status - Issue, 1996.

ZHANG, L. SHUZHEN, Y Research on Workflow Patterns based on Petri nets. In: IEEE Conference on Cybernetics & Intelligent Systems (CIS) Robotics, Automation and Mechatronics (RAM), 2006, Bangkok, p. 1-6, 2006.

ZISMAN, M. D. *Representation, specification and automation of office procedures.* Thesis (PhD.) - University of Pennsylvania, Wharton School of Business. 1977.

## SOBRE OS AUTORES

### Silvia Inês Dallavalle de Pádua

Departamento de Administração  
FEARP-USP

End.: Av. dos Bandeirantes, 3900 - FEA-RP - Bloco A - Campus da USP - Monte Alegre - 14040-900 - Ribeirão Preto - SP - Brasil

Tel.: (16) 3602-3903

E-mail: [dallavalle.silvia@gmail.com](mailto:dallavalle.silvia@gmail.com)

### Ricardo Yassushi Inamasu

Centro Nacional de Pesquisa e Desenvolvimento de Instrumentação Agropecuária - EMBRAPA

End.: Rua XV de Novembro, 1452 - São Carlos - SP - CEP 13560-970

Tel.: (16) 3374-2477

E-mail: [ricardo@cnpdia.embrapa.br](mailto:ricardo@cnpdia.embrapa.br)