

Método de avaliação do modelo de processos de negócio do *Enterprise Knowledge Development*

Sílvia Inês Dallavalle de Pádua

Ricardo Yassushi Inamasu



Resumo

Apresenta o método de avaliação do modelo de processos de negócio do *Enterprise Knowledge Development* (MPN-EKD) e um exemplo de aplicação. Para criar o método de avaliação, foi desenvolvida a formalização do modelo de processos de negócio do EKD (MPN-EKD) e o mapeamento desse modelo em redes de Petri. Por meio desse método, é possível verificar se o modelo tem erros de construção e travamentos. O EKD é uma metodologia que fornece uma forma sistemática e controlada de analisar, entender, desenvolver e documentar uma organização. A metodologia EKD não possui uma sintaxe e uma semântica bem definidas, dificultando análises mais complexas dos modelos e a verificação da consistência do modelo. Essa dificuldade motivou um estudo baseado em redes de Petri. O formalismo de redes de Petri a torna uma importante técnica de modelagem para a representação de processos. Além disso, redes de Petri permitem rastrear cada etapa da operação sem ambigüidade e possuem métodos eficientes de análise que garantem que o modelo está livre de erros.

Palavras-chave: Redes de Petri. Modelagem organizacional. Modelo de processos de negócio. Workflow. EKD. Avaliação do modelo de processos de negócio.

1 Introdução

Os processos de negócio são blocos fundamentais da construção de uma organização de sucesso. A tecnologia da informação, quando 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. As necessidades do negócio devem ser atendidas pela tecnologia da informação buscando os objetivos de negócio como concorrência, competitividade e estratégias. Sistemas que não satisfazem as necessidades da organização podem impedir o desenvolvimento do negócio.

De acordo com Jacobson et al. (1999) e Kruchten (2000), as técnicas aplicadas ao desenvolvimento de sistemas não ajudam a buscar soluções alternativas aos problemas da organização, não adicionam valor ao negócio e, na maioria das vezes, os processos manuais são automatizados sem nenhuma modificação. Isso ocorre porque não foram considerados aspectos mais amplos como os objetivos da organização, regras do negócio, restrições, aspectos não funcionais relacionados à qualidade, confiabilidade e usabilidade.

A modelagem organizacional, nesse contexto, facilita a compreensão do ambiente empresarial e é reconhecida como uma atividade valiosa para o desenvolvimento de

sistemas de informação de acordo com Nurcan e Barrios (2003) e Persson (2000). 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 *Enterprise Knowledge Development* (EKD) 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; NURCAN, 1998). 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. Essa metodologia é explicada em detalhes em Pádua et al. (2004a), Dallavalle e Cazarini (2001) e Pádua (2001).

O principal problema das abordagens de Modelagem Organizacional, incluindo-se o EKD, é a ausência de técnicas de análise 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, pode-se mencionar: Dongen et al. (2007), Lenz et al. (2005), Mevius e Oberweis (2005), Pádua et al. (2004b), Koubarakis e Plexousakis (2002), Junginger et al. (2001), Jonkers et al. (2003), Dehnert (2003), Pádua (2004), Pádua et al. (2003), Pádua et al. (2002) e Aalst (1999).

De acordo com Pádua (2004), 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. Os principais conceitos de redes de Petri são discutidos em Pádua et al. (2002).

Muitos trabalhos têm valorizado a estrutura formal das redes de Petri para representação de processos de negócios, entre eles podem ser mencionados 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), Pádua et al. (2004b), Pádua et al. (2002) e Aalst (1999).

Assim, este trabalho apresenta o método de avaliação do modelo de processos de negócios do EKD. Para que o método fosse criado, foi necessário desenvolver a formalização do modelo de processos de negócio do EKD e o mapeamento do modelo de processos de negócio em redes de Petri.

Foi aplicado o método de avaliação no modelo processo de planejamento estratégico de recursos humanos desenvolvido no projeto ESPRIT ELEKTRA (*Electrical Enterprise Knowledge for Transforming Applications*) (BUBENKO et al., 1998). O modelo mapeado em redes de Petri foi simulado na ferramenta *Petri Net Tools* desenvolvida e implementada no Laboratório de Simulação e Controle de Sistemas Discretos da USP de São Carlos (SOARES, 2001). O editor possuía seis módulos. Quatro módulos estavam em funcionamento: rede de Petri L/T (lugar-transição), *Mark Flow Graph* (MFG), *Sequential Flow Chart* (SFC) e PN Estocásticas. O editor permite os seguintes tipos de análises: árvore de alcançabilidade; matriz de incidência; limitação; vivacidade; verificação do estado final, transições e lugares invariantes.

Nos trabalhos de Aalst e Hee (2002), Verbeek et al. (2002), Salimifard e Wright (2001), Aalst (1999), Aalst e Hofstede (2000), Voorhoeve (2000), Mold e Valk (2000) os processos de negócio são modelados diretamente em redes de Petri. Neste trabalho, a construção do modelo de processos de negócio seguiu o método de modelagem organizacional EKD e não diretamente em redes de Petri.

O procedimento de mapeamento do modelo de processos de negócio em redes de Petri foi desenvolvido baseado em redes de Petri lugar/transição. O poder de análise das propriedades do modelo seria reduzido caso o mapeamento fosse baseado em redes estendidas e de alto nível. Métodos para avaliar uma rede de Petri colorida são computacionalmente caros sendo viáveis apenas para modelos mais simples (JENSEN, 1997).

O trabalho está estruturado em nove seções, incluindo a presente introdução. Na seção dois, apresentam-se conceitos importantes relacionados a redes de Petri no contexto de modelo de processos de negócios. Na seção três, faz-se uma apresentação da formalização do modelo de processos de negócio. As seções quatro e cinco apresentam respectivamente mapeamento do modelo de processos de negócio em redes de Petri e o método de avaliação do modelo de processos de negócios do EKD. A seção seis mostra a aplicação do método. As considerações finais são apresentadas na seção sete.

2 Redes de Petri e Modelo de Processos de Negócio

De acordo com Aalst e Hofstede (2000), alguns erros são facilmente identificados nos modelos de redes de Petri como travamento (*deadlock*), quando não é possível executar nenhuma tarefa; *livelock*, quando um caso fica em *loop* infinito sendo possível executar tarefas, mas nenhum progresso é possível e tarefas mortas (*deadtask*), quando uma tarefa nunca pode ser executada em nenhuma situação.

Alguns trabalhos investigam o uso de subclasses de redes de Petri para aumentar o poder de decisão sem reduzir o poder de modelagem das redes de Petri. Nestas subclasses, são feitas algumas restrições estruturais às redes de Petri. A subclasse de redes de Petri denominada escolha-livre possibilita a modelagem de conflito do paralelismo e a sincronização. Quando um lugar é entrada de diversas transições, este lugar é a única entrada destas transições. Desta forma, todas as transições estarão habilitadas ou nenhuma estará, possibilitando a escolha do evento livremente. Formalmente a definição de escolha-livre é: Seja uma rede de Petri = (P, T, I, O, K). I é o conjunto das entradas às transições e O é o conjunto das saídas das transições. K é a capacidade dos lugares. Esta rede é classificada como uma rede de escolha-livre se, e somente se, $I(t_j) = \{p_i\}$ ou $O(p_i) = \{t_j\}$, $\forall t_j \in T$ e $p_i \in I(t_j)$.

O principal problema das abordagens de Modelagem Organizacional, incluindo-se o EKD, é a ausência de técnicas de análise objetivas. Nesse caso, as redes de Petri têm um excelente potencial para resolver 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 métodos de análise. As diversas aplicações das redes de Petri na Engenharia são apresentadas em Pádua et al. (2003).

Desde que Zisman (1977) usou as redes de Petri para modelar *workflow* pela primeira vez, muitos autores publicaram trabalhos que procuravam, também, integrar os dois assuntos. Entre eles, podem ser mencionados: Chrzastowski-Wachtel et al. (2003), Rinderle et al. (2003), Dehnert (2003), Grigorova (2003) e Verbeck et al. (2002). Aalst e Hee (2002) e Pádua et al. (2004) explicam que existem diversas razões para se usar redes de Petri para modelagem de processos de negócio: semântica formal, natureza gráfica, expressividade, propriedades, análise e por não depender de fornecedor.

O critério de verificação de corretude definido para *workflow-nets* é chamado *Soundness*. *Sound* é sinônimo de correto de acordo com Aalst e Hee (2002). Aalst (1997) desenvolveu uma técnica que verifica se o procedimento satisfaz os seguintes requisitos (de *soundness*): não deve existir tarefa que não contribui para o processamento dos casos; para qualquer caso, o procedimento terminará eventualmente e no momento em que o procedimento termina para casos específicos, todas as referências a esse caso devem ser removidas.

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, baseada 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.

Visando à 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_J , 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_J 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, sendo executadas 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; e
- *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 MPN-EKD, foi necessário criar esses estados para que a

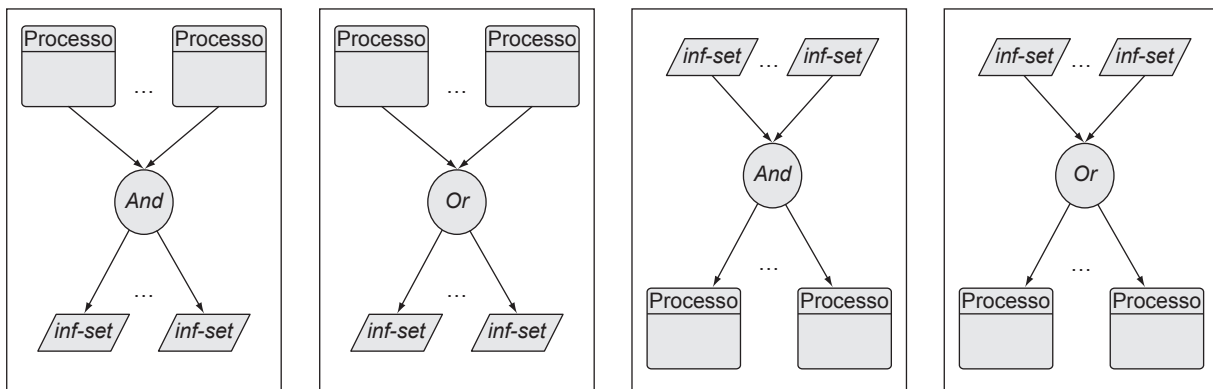


Figura 1. Construções que não são permitidas em um MPN-EKD formal.

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; e
- $A \subseteq (I \times P) \cup (P \times I) \cup (I \times C) \cup (C \times I) \cup (P \times C) \cup (C \times P)$ 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_1 para um nodo n_k é uma seqüência $\langle n_1, n_2, \dots, n_k \rangle$, 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} . O conjunto C_j e C_s é usado 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 $n_1 \in I$, $n_2 \in C$, $n_3 \in P$; e
- $C_{PI} \subseteq C$ 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 MPN-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, $I \cap P = \emptyset$, $I \cap C = \emptyset$, e $P \cap C = \emptyset$;
- 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 grafo 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 $C_j \cup C_s = C$; e
- 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 MPN-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. 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* ($|\bullet c| = 1$ e $|c \bullet| \geq 2$). 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 MPN-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 compor-

tamento 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 (Equação 1).

$$L^{PN} = I \cup \left(\bigcup_{c \in C} L_c^{PN} \right) \quad (1)$$

O conjunto de lugares (L^{PN}) é formado pela união de todos os *inf-sets* com lugares que foram incluídos para representar conectores ($\bigcup_{c \in C} L_c^{PN}$) (Equação 2).

$$T^{PN} = P \cup \left(\bigcup_{c \in C} T_c^{PN} \right) \quad (2)$$

O conjunto de transições (T^{PN}) é formado pela união de todos os processos com transições que foram incluídas para representar conectores ($\bigcup_{c \in C} T_c^{PN}$) (Equação 3).

$$F^{PN} = (A \cap ((I \times P) \cup (P \times I))) \cup \left(\bigcup_{c \in C} F_c^{PN} \right) \quad (3)$$

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} estão nas Equações 4, 5 e 6, respectivamente.

$$L_c^{PN} = \emptyset \quad (4)$$

$$T_c^{PN} = \emptyset \quad (5)$$

$$F_c^{PN} = \{(x, y) \mid x \in \bullet c \text{ e } y \in c \bullet\} \quad (6)$$

A Equação 4 determina que para representar esse conector não é necessário acrescentar lugares. A Equação 5 determina que para representar esse conector não é necessário acrescentar transições. A Equação 6 determina 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} estão nas Equações 7, 8 e 9, respectivamente.

$$L_c^{PN} = \{l_x^c \mid x \in \bullet c\} \quad (7)$$

$$T_c^{PN} = \{t^c\} \quad (8)$$

$$F_c^{PN} = \{(x, l_x^c) \mid x \in \bullet c\} \cup \{(l_x^c, t^c) \mid x \in \bullet c\} \cup \{(t^c, x) \mid x \in c \bullet\} \quad (9)$$

Na Equação 7, é determinado que, para representar esse conector, é necessário acrescentar um lugar para cada processo de entrada do conector. Na Equação 8, é indicado que, para representar esse conector, é necessário acrescentar uma transição. Na Equação 9, 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

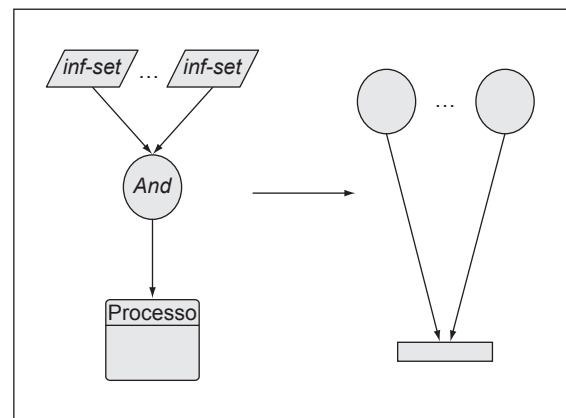


Figura 2. Exemplo de mapeamento do C_{AND} entre dois ou mais *inf-sets* para um processo.

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} estão nas Equações 10, 11 e 12, respectivamente.

$$L_c^{PN} = \{l^c\} \tag{10}$$

$$T_c^{PN} = \{t_x^c \mid x \in \bullet c\} \tag{11}$$

$$F_c^{PN} = \{(x, t_x^c) \mid x \in \bullet c\} \cup \{(t_x^c, l^c) \mid x \in \bullet c\} \cup \{(l^c, x) \mid x \in c \bullet\} \tag{12}$$

Na Equação 10, é determinado que, para representar este conector, é necessário acrescentar um lugar. Na Equação 11, é estabelecido que, para representar este conector, é necessário acrescentar uma transição para cada *inf-set* de entrada do conector. Na Equação 12, 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 o 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} estão nas Equações 13, 14 e 15, respectivamente.

$$L_c^{PN} = \emptyset \tag{13}$$

$$T_c^{PN} = \emptyset \tag{14}$$

$$F_c^{PN} = \{(x, y) \mid x \in \bullet c \text{ e } y \in c \bullet\} \tag{15}$$

Na Equação 13, é apresentado que não é necessário acrescentar lugares para representar esse conector. Na Equação 14, é determinado que não é necessário acrescentar transições para representar esse conector. Na Equação 15, 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 $c \in C_{PI} \cap C_J \cap C_{OR}$.

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

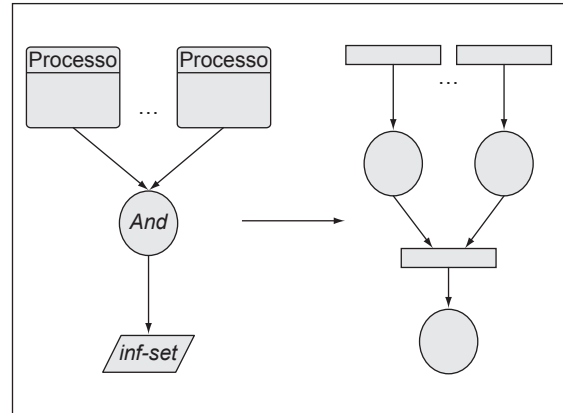


Figura 3. Exemplo de mapeamento do C_{AND} entre dois ou mais processos para um *inf-set*.

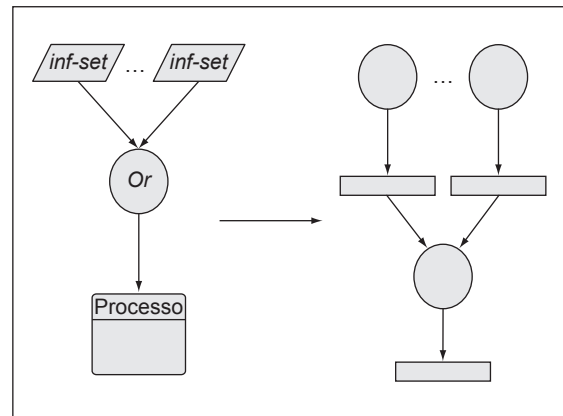


Figura 4. Exemplo de mapeamento C_{OR} entre dois ou mais *inf-sets* para um processo.

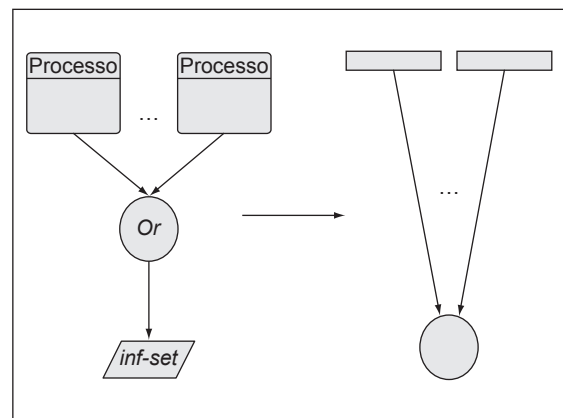


Figura 5. Exemplo de mapeamento C_{OR} entre dois ou mais processos para um *inf-set*.

C_{AND} , as definições de L_c^{PN} , T_c^{PN} e F_c^{PN} estão nas Equações 16, 17 e 18, respectivamente.

$$L_c^{PN} = \{l_x^c \mid x \in c\} \quad (16)$$

$$T_c^{PN} = \{t^c\} \quad (17)$$

$$F_c^{PN} = \{(x, t^c) \mid x \in c\} \cup \{(t^c, l_x^c) \mid x \in c\} \cup \{(l_x^c, x) \mid x \in c\} \quad (18)$$

Na Equação 16, é apresentado que é necessário acrescentar um lugar para cada processo de saída do conector. Na Equação 17, é determinado que para representar este conector é necessário acrescentar uma transição. Na Equação 18, 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 $c \in C_{IP} \cap C_s \cap C_{AND}$.

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} estão nas Equações 19, 20 e 21, respectivamente.

$$L_c^{PN} = \emptyset \quad (19)$$

$$T_c^{PN} = \emptyset \quad (20)$$

$$F_c^{PN} = \{(x, y) \mid x \in c \text{ e } y \in c\} \quad (21)$$

Na Equação 19, é apresentado que não é necessário acrescentar lugares para representar esse conector. Na Equação 20, é determinado que não é necessário acrescentar transições para representar esse conector. Na Equação 21, 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 é 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

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

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} estão nas Equações 22, 23 e 24, respectivamente.

$$L_c^{PN} = \emptyset \quad (22)$$

$$T_c^{PN} = \emptyset \quad (23)$$

$$F_c^{PN} = \{(x, y) \mid x \in c \text{ e } y \in c\} \quad (24)$$

Na Equação 22, é apresentado que não é necessário acrescentar lugares para representar esse conector. Na Equação 23, é determinado que não é necessário acrescentar transições para representar esse conector. Na Equação 24, 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 é 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

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

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} estão nas Equações 25, 26 e 27, respectivamente.

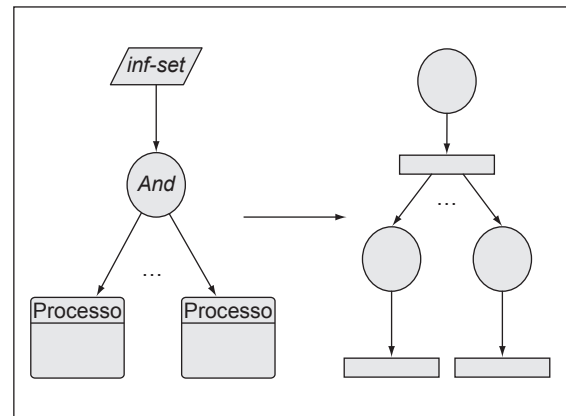


Figura 6. Exemplo de mapeamento do C_{AND} um *inf-set* para dois ou mais processos.

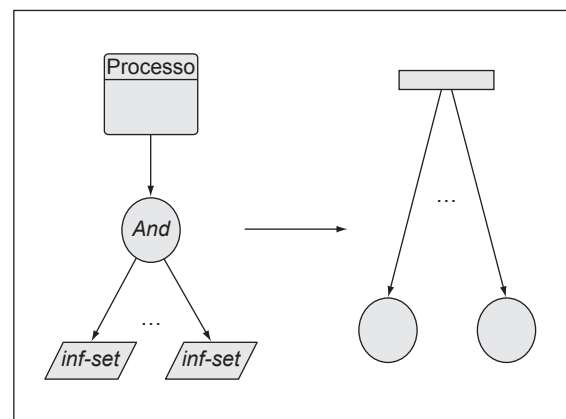


Figura 7. Exemplo de mapeamento do C_{AND} de um processo para dois ou mais *inf-sets*.

$$L_c^{PN} = \{l^c\} \tag{25}$$

$$T_c^{PN} = \{t_x^c \mid x \in c \bullet\} \tag{26}$$

$$F_c^{PN} = \{(x, l^c) \mid x \in c \bullet\} \cup \{(l^c, t_x^c) \mid x \in c \bullet\} \cup \{(t_x^c, x) \mid x \in c \bullet\} \tag{27}$$

Na Equação 25, é determinado que, para representar este conector, é necessário acrescentar um lugar. Na Equação 26, é apresentado que é necessário acrescentar uma transição para cada *inf-set* de saída do conector. Na Equação 27, 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 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

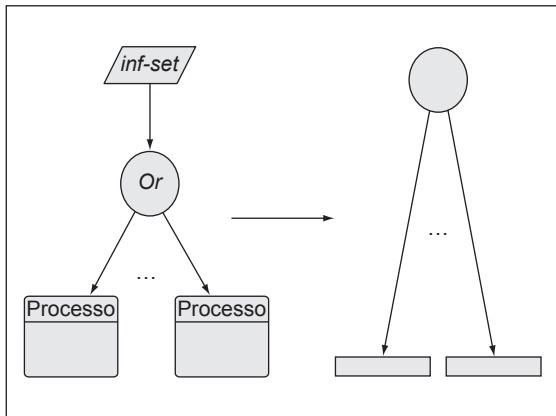


Figura 8. Exemplo de mapeamento do C_{OR} de um *inf-set* para dois ou mais processos.

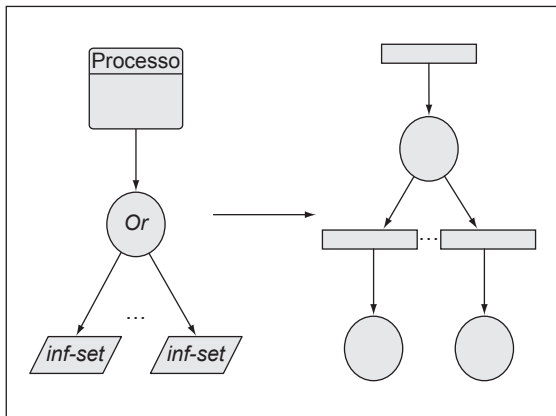


Figura 9. Exemplo de mapeamento do C_{OR} de um processo para dois ou mais *inf-sets*.

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) pode facilmente ser estendido 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:

- a) composto de partes completamente disjuntas;
- b) tem partes que nunca são ativadas; e
- c) 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 regular.

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:

- para cada marcação M alcançável do estado inicial (por exemplo, o estado, no qual o *inf-set* $i_{início}$ é o único *inf-set* que existe), existe uma seqüência de disparos levando da marcação M para a marcação final (por exemplo, em que o *inf-set* i_{final} é o único *inf-set* que existe);
- a única marca existente no final do processo está no estado final i_{final} ; e
- não existirem processos *dead*, por exemplo, para cada processo p existe uma seqüência disparável, a qual executa p .

A conformidade, de acordo com as das redes de Petri, é o requisito mínimo que qualquer MPN-EKD deveria satisfazer. Um MPN-EKD *sound* é livre de potenciais travamentos e *livelocks*. Se assumir *fairness*, então os primeiros dois requisitos implicam 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 $\bullet t = \bullet t_2$.

Para os MPN-EKD complexos encontrados na prática, a verificação da propriedade *soundness* não é simples. Felizmente, técnicas e ferramentas baseadas em redes de Petri podem ser usadas para analisar essa propriedade. A inspeção da árvore de cobertura da rede de Petri, que corresponde ao MPN-EKD é suficiente para verificar *soundness*. Para MPN-EKD complexos, a árvore 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 árvores 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 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*.

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 PN com uma transição t adicional conectando i_{final} para $i_{início}$ e deixar M ser a marcação inicial com uma marca em $i_{início}$ (AALST, 1999).

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

(\overline{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 verificadas em tempo polinomial. Por essa razão, *soundness* pode ser verificado em tempo polinomial.

Nesse sentido, é possível estender ferramentas com procedimentos 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 Método de avaliação do Modelo de Processos de Negócio do Enterprise Knowledge Development

O método de avaliação do modelo de processos de negócio consiste em:

- 1) Desenvolver o modelo organizacional EKD utilizando as diretrizes apresentadas em Bubenko et al. (1998).
- 2) Desenvolver o Modelo de Processos de Negócio de acordo com a formalização do MPN-EKD apresentada neste trabalho. Conferir se o modelo atende aos seguintes requisitos:
 - 2.1) Todos os processos devem ter condição de entrada e saída. Quando um processo não tem nenhuma condição de entrada, não fica claro quando poderá ser realizado. Quando um processo não tem condições de saída, não contribui para o sucesso do caso e pode ser omitido;
 - 2.2) Deve haver ao menos um *inf-set* final e um *inf-set* inicial;
 - 2.3) A entrada do processo deve ser igual a 1;
 - 2.4) A saída do processo deve ser igual a 1;
 - 2.5) A saída do *inf-set* deve ser igual ou menor que 1. Menor que 1 caso seja um *inf-set* final;
 - 2.6) A entrada do *inf-set* deve ser igual ou menor que 1. Menor que 1 caso seja um *inf-set* inicial;
 - 2.7) A entrada do conector deve ser maior ou igual a 1;
 - 2.8) A saída do conector deve ser maior ou igual a 1;
 - 2.9) Todo conector deve ser do tipo OR ou AND;
 - 2.10) Todo conector deve ser do tipo *Split* ou *Join*;
 - 2.11) Todo conector deve ser do tipo PI ou IP;
 - 2.12) Um conector do tipo *split* deve ter a entrada igual a 1;
 - 2.13) Um conector do tipo *split* deve ter a saída igual ou maior que 2;
 - 2.14) Um conector do tipo *join* deve ter a entrada maior ou igual a 2;
 - 2.15) Um conector do tipo *join* deve ter a saída igual a 1;
 - 2.16) Não é permitido conectar processo a processo;
 - 2.17) Não é permitido conectar *inf-set* a *inf-set*;
 - 2.18) Não é permitido utilizar o conector ligando processo(s) a processo(s) e *inf-set*(s) a *inf-set*(s);

- 2.19) Não é permitido a conexão de conector(es) com conector(es); e
- 2.20) Todos os *inf-sets* que não foram gerados pelo processo devem ser habilitados.
- 3) Mapear o modelo em redes de Petri. Os *inf-sets* são representados por lugares e os processos são representados por transições. Para mapear os conectores, é necessário seguir as regras apresentadas a seguir:
- 3.1) O conjunto de lugares é formado pela união de todos os *inf-sets* com lugares que foram incluídos para representar os conectores;
- 3.2) O conjunto de transições é formado pela união de todos os Processos com transições que foram incluídas para representar conectores;
- 3.3) O conjunto de arcos da rede é formado pelos arcos de modelo que vão de I (*inf-set*) a P (processo) e de P a I e a união dos arcos incluídos para representar conectores;
- 3.4) A Regra 1 estabelece que, quando o conector c pertence a C_{IP} (caminho de *inf-set* para processo) intersecção de C_J (*join*) intersecção de C_{AND} , o conector ($c \in C_{IP} \cap C_J \cap C_{AND}$) corresponde a dois ou mais arcos em redes de Petri;
- 3.5) A Regra 2 estabelece que quando o conector c pertence a C_{PI} (caminho de processo para *inf-set*) intersecção de C_J (*join*) intersecção de C_{AND} , o conector ($c \in C_{PI} \cap C_J \cap C_{AND}$) tem o comportamento de uma transição. É acrescentado um lugar para cada processo de entrada do conector;
- 3.6) A Regra 3 estabelece que, quando o conector c pertence a C_{IP} (caminho de *inf-set* para processo) intersecção de C_J (*join*) intersecção de C_{OR} , o conector ($c \in C_{IP} \cap C_J \cap C_{OR}$) tem o comportamento de um lugar;
- 3.7) A Regra 4 estabelece que, quando o conector c pertence a C_{PI} (caminho de processo para *inf-set*) intersecção de C_J (*join*) intersecção de C_{OR} , o conector ($c \in C_{PI} \cap C_J \cap C_{OR}$) corresponde a dois ou mais arcos em redes de Petri;
- 3.8) A Regra 5 estabelece que, quando o conector c pertence a C_{IP} (caminho de *inf-set* para processo) intersecção de C_S (*split*) intersecção de C_{AND} , o conector ($c \in C_{IP} \cap C_S \cap C_{AND}$) tem o comportamento de uma transição seguida de um número de lugares igual ao número de processos;
- 3.9) A Regra 6 estabelece que, quando o conector c pertence a C_{IP} (caminho de *inf-set* para processo) intersecção de C_S (*split*) intersecção de C_{AND} , o conector ($c \in C_{PI} \cap C_S \cap C_{AND}$) corresponde a um número de arcos em redes de Petri;
- 3.10) A Regra 7 estabelece que, quando o conector c pertence a C_{IP} (caminho de *inf-set* para processo) intersecção de C_S (*split*) intersecção de C_{OR} , o conector ($c \in C_{IP} \cap C_S \cap C_{OR}$) corresponde a um número de arcos em redes de Petri; e
- 3.11) A Regra 8 estabelece que quando o conector c pertence a C_{PI} (caminho de processo para *inf-set*) intersecção de C_S (*split*) intersecção de C_{OR} , o conector ($c \in C_{PI} \cap C_S \cap C_{OR}$) corresponde a um lugar seguido de um número de transições igual ao número de processos de saída do conector.
- 4) Construir a árvore de alcançabilidade. Por meio da árvore de alcançabilidade é possível verificar vários erros que podem ocorrer na definição de processo, mesmo sem conhecimento específico do processo de negócio. Na ausência de ferramenta de edição de redes de Petri, é possível verificar a propriedade *soundness* por meio da inspeção da árvore de alcançabilidade que corresponde ao MPN-EKD.
- 5) Depois de fazer o mapeamento em redes de Petri, avaliar o modelo utilizando uma ferramenta de edição de redes de Petri. Neste trabalho, está sendo utilizado *Petri Net Tools*. Os seguintes itens devem ser considerados:
- 5.1) Verificação de possível travamento (*deadlock*), ou seja, quando não é possível executar nenhuma tarefa;
- 5.2) Eliminação de casos que ficam em *loop* infinito (*livelock*);
- 5.3) Verificação de possíveis tarefas que não podem ser executadas (*deadtask*);
- 5.4) Eliminação de conflitos;
- 5.5) Verificação de possíveis caminhos;
- 5.6) Verificar a existência de marcas nos outros lugares depois que a condição fim foi completada. Uma vez que a marca aparece no lugar fim, todas as outras marcas devem ter desaparecido; e
- 6) Apresentar relatório de problemas encontrados.

6 Aplicação do método

O modelo que será apresentado foi desenvolvido no projeto ESPRIT ELEKTRA (*Electrical Enterprise Knowledge for Transforming Applications*) (ELEKTRA, 2000). O projeto ELEKTRA concentra-se principalmente na aplicação do método EKD para problemas de gerenciamento de mudanças dentro de organizações da Grécia

e Suécia, gerando um conjunto de práticas genéricas de forma a aplicá-las em outras companhias.

O caso Vattenfall foi escolhido no projeto ELEKTRA. O projeto foi baseado em uma cuidadosa análise não apenas das práticas correntes e processo, mas também de problemas, das necessidades, das oportunidades e objetivos futuros percebidos.

O modelo é do processo de planejamento estratégico de recursos humanos. O processo de planejamento é realizado em um nível estratégico. Envolve a formulação de políticas e objetivos para o planejamento de recursos humanos para Vattenfall, tratando as métricas sobre a realização dos objetivos formulados. O modelo descreve como o planejamento de recursos humanos deveria ser integrado com o planejamento estratégico de negócio. Inicialmente, partindo dos objetivos do planejamento do negócio, é formulado o grupo de objetivos e indicadores dentro do domínio de competência, em seguida, são comunicados os objetivos e indicadores para área de negócio nos documentos: pré-condições para planejamento de negócio/diretrizes de orçamento. Paralelamente ou não, são formuladas as políticas, as diretrizes e as instruções a partir do termo dos objetivos particulares e políticos. Em seguida, são feitas comunicações sobre essas políticas. Os processos seguintes são: executar atividades dentro do domínio do suplente na competência; retornar trimestralmente os objetivos e indicadores com a ajuda do Sistema de Revisão do Grupo; apresentar um sumário dos objetivos alcançados e indicadores; comparar objetivos realizados e tendências com objetivos propostos de acordo com as políticas particulares e planejamento de negócio; e revisar objetivos. Na Figura 10, é apresentado o modelo descrito alterado para ser mapeado em redes de Petri. As entradas do processo 1 foram modificadas por não atender ao requisito do método que afirma que o processo só poderá ter uma entrada. Embora o método afirme que deve ser colocado um *inf-set* inicial e final, não foi possível colocar *inf-set* final por não estar claro o fim do procedimento. Foram adicionados os conectores.

Na Figura 11, é apresentado o mesmo modelo mapeado em redes de Petri de acordo com o método apresentado neste trabalho. Os conectores do modelo de processo de planejamento estratégico de recursos humanos e suas respectivas regras de mapeamento são apresentados no Quadro 1. Os elementos de redes de Petri correspondentes aos elementos do MPN-EKD são apresentados no Quadro 2.

6.1 Resultado do teste

Na Figura 12, é apresentado o modelo simulado na ferramenta Petri Net Tools. A árvore de alcançabilidade apresentada na Figura 13 demonstra que a transição Pr6 será disparada apenas se houver uma marca no lugar IS8. Como não existe essa marca antes do disparo da transição

Pr7 haverá um travamento (*deadlock*). Além disso, não existe uma condição fim clara. O modelo não é *sound*.

7 Considerações finais

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. Constatou-se 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, possuem uma estrutura simples antes de serem introduzidos nos sistemas de informação avançados, tais como o 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 pesquisa 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.

Foi possível constatar 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_I , 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

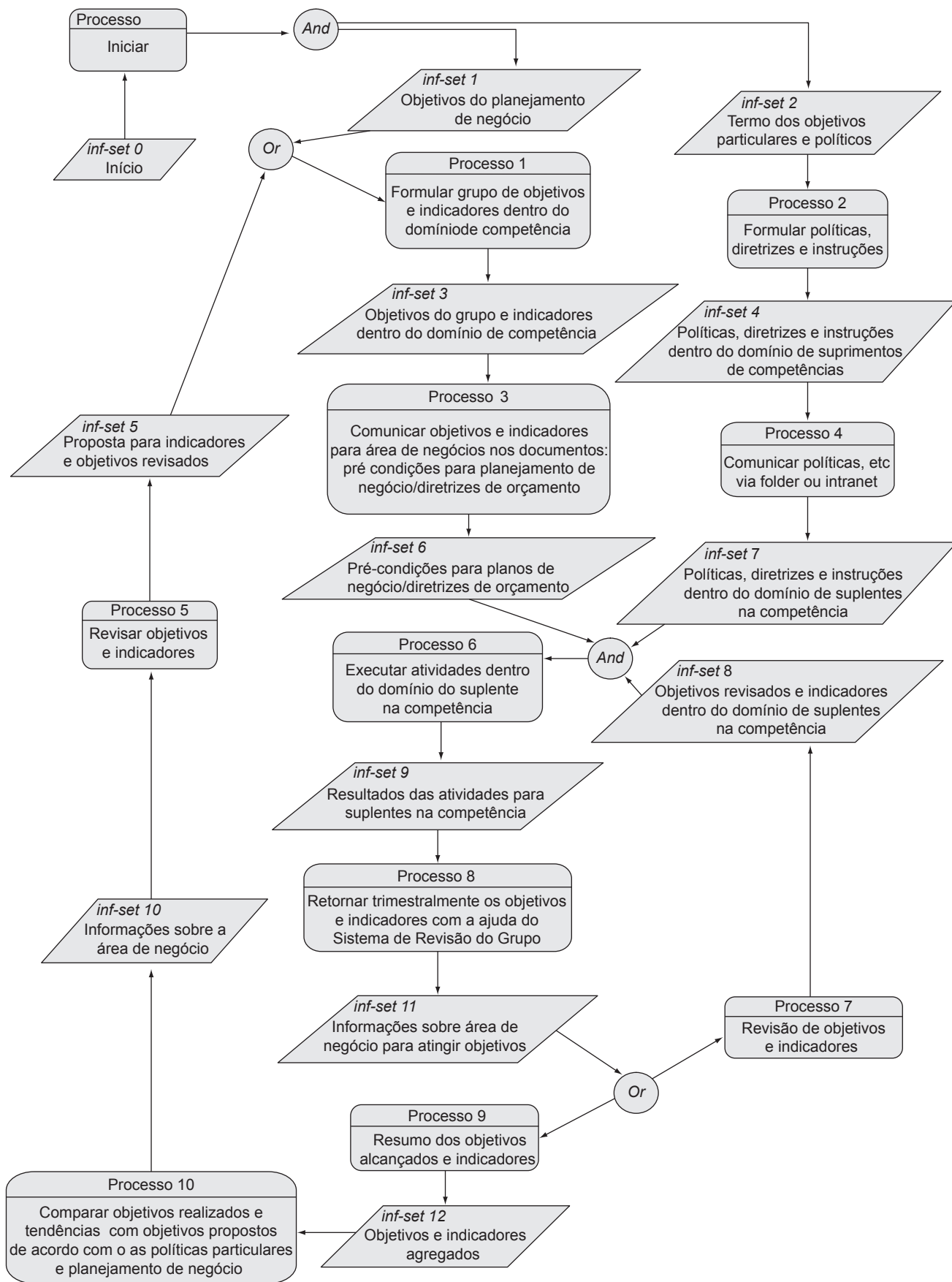


Figura 10. Modelo de Processo de planejamento estratégico de recursos humanos modificado.

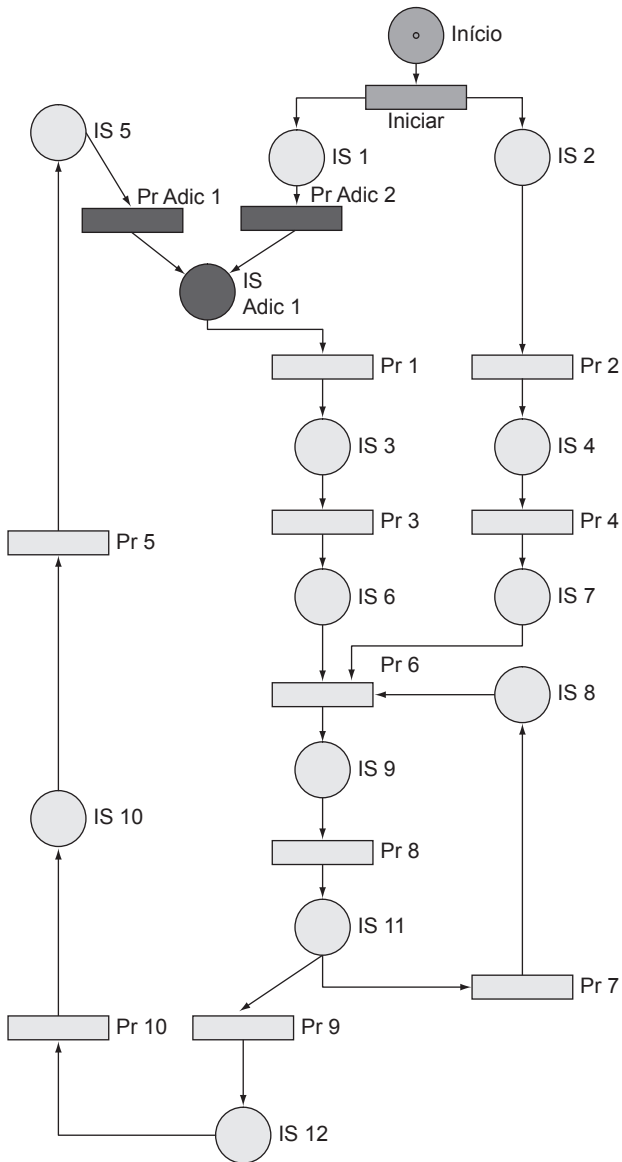


Figura 11. Modelo de Planejamento estratégico mapeado em redes de Petri.

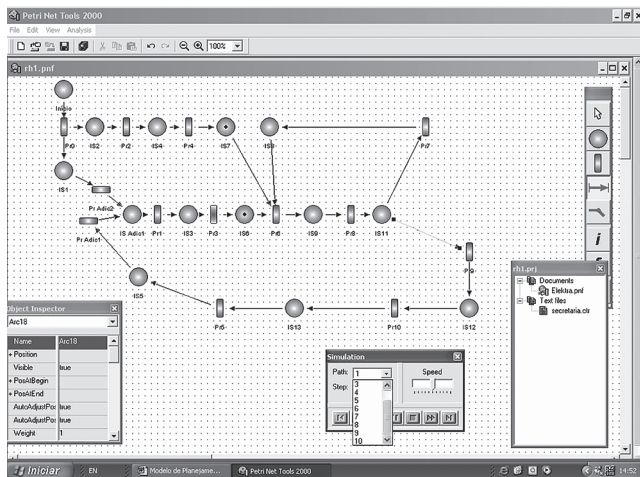


Figura 12. Simulação do Modelo de Planejamento estratégico na ferramenta Petri Net Tools.

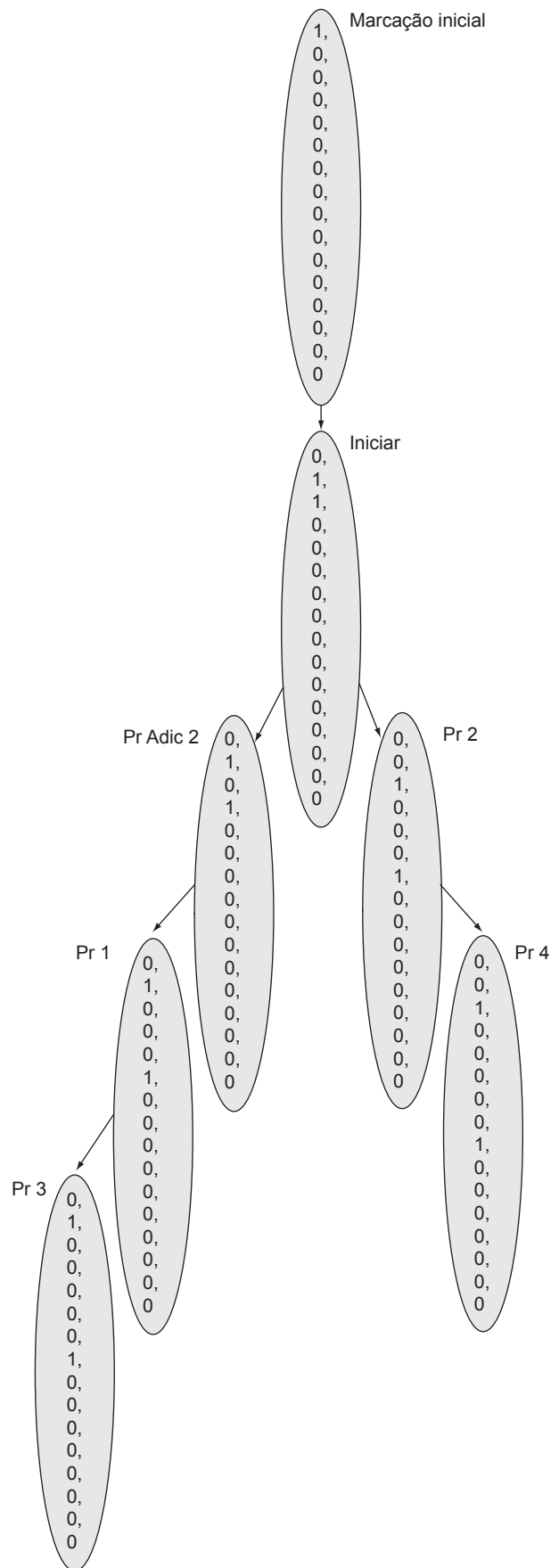


Figura 13. Árvore de alcançabilidade do Modelo de Planejamento estratégico mapeado em redes de Petri.

Quadro 1. Conectores utilizados no Modelo de Planejamento estratégico de recursos humanos e a regra de mapeamento correspondente.

Conector	Regra correspondente
$C_{PI} C_S C_{AND}$	Regra 6
$C_{IP} C_J C_{OR}$	Regra 3
$C_{IP} C_J C_{AND}$	Regra 1
$C_{IP} C_S C_{OR}$	Regra 7

Quadro 2. Elementos de redes de Petri correspondentes aos elementos do MPN-EKD de planejamento estratégico.

PN	MPN-EKD	
Lugar	<i>inf-set</i>	Descrição
IS1	1	Objetivos do planejamento de negócio
IS2	2	Termo dos objetivos particulares e políticos
IS3	3	Objetivos do grupo e indicadores dentro do domínio de competência
IS4	4	Políticas, diretrizes e instruções dentro do domínio de suprimentos de competências
IS5	5	Proposta para indicadores e objetivos revisados
IS6	6	Pré-condições para planos de negócio/diretrizes de orçamento
IS7	7	Políticas, diretrizes e instruções dentro do domínio de suplentes na competência
IS8	8	Objetivos revisados e indicadores dentro do domínio de suplentes na competência
IS9	9	Resultados das atividades para suplentes na competência
IS10	10	Informações sobre a área de negócio
IS11	11	Informações sobre a área do negócio para atingir objetivos
IS12	12	Objetivos e indicadores agregados
Tr	Pr	Descrição
Pr1	1	Formular grupo de objetivos e indicadores dentro do domínio de competência
Pr2	2	Formular políticas
Pr3	3	Comunicar objetivos e indicadores para área de negócio nos documentos: pré-condições para planejamento de negócio/diretrizes para orçamento
Pr4	4	Comunicar políticas via folder ou intranet
Pr5	5	Revisar objetivos e indicadores
Pr6	6	Executar atividades dentro do domínio do suplente na competência
Pr7	7	Revisão dos objetivos e indicadores
Pr8	8	Retornar trimestralmente os objetivos e indicadores com a ajuda do sistema de revisão do grupo
Pr9	9	Resumo dos objetivos alcançados e indicadores
Pr10	10	Comparar objetivos realizados e tendências com objetivos propostos de acordo com as políticas particulares e planejamento de negócio

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 MPN-EKD original.

Foi desenvolvido neste trabalho um procedimento de mapeamento formal do modelo de processos de negócio em redes de Petri. O procedimento de mapeamento foi desenvolvido com base 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.

Dessa forma, a partir do procedimento de mapeamento formal do modelo de processos de negócio em redes de Petri, foi criado o Método de Avaliação do MPN-EKD. O método consiste em uma seqüência de passos que vai desde o desenvolvimento do modelo organizacional até a construção da árvore de alcançabilidade e simulação do modelo em ferramenta.

A aplicação do método permitiu verificar a presença de *deadlock* (travamento), no qual o processo nunca poderá ser realizado. Além disso, não existe uma condição fim clara e dessa forma, o modelo não é *sound*.

Com base nesses problemas, pode-se afirmar que o cuidado no processo de modelagem é fundamental para que o modelo represente fielmente como o processo é realizado e que esses problemas são minimizados quando o modelo é desenvolvido de acordo com o método desenvolvido neste trabalho.

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 disponibiliza 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 apresentadas neste trabalho.

É importante ressaltar que a aplicação do método de avaliação em muitos Modelos de Processos de Negócio poderá ser inviável sem uma ferramenta computacional que apóie todos os passos do método.

Assessment Method of the Business Process Model of EKD

Abstract

The EKD – Enterprise Knowledge Development - is a methodology that offers a systematic and guided process to analyze, understand, develop, and document an Enterprise. Unfortunately, its grammatical structure and context formation are not clear nor easy to understand and interpret making the analyses of more complex models difficult.

As a result, the Enterprise Process model of EKD can be ambiguous and hard to analyze, especially regarding more complex systems, and so it can also be difficult to verify the consistency and entireness the model. In this work, these problems will be studied based on the Petri Nets approach. Due to the Petri Nets formalism, this is an important modeling technique to process representation. Furthermore, Petri Nets allows the tracking of every step of the operation with no ambiguity, and it offers efficient methodology for the analyses, which can guarantee the accuracy of the model. Therefore, the objective of this study is to develop an evaluation methodology of the business process model of EKD (MPN-EKD). Such methodology will allow the verification of possible building and locking model errors. This methodology can be applied to information systems or workflow. It can also be used to study strategies of work and workflow simulations.

Keywords: Petri nets. Enterprise modeling. Business rules. Software development. Business process model. Workflow. EKD. Assessment of business process model.

Referências bibliográficas

- AALST, W. M. P. V. D. Formalization and verification of event-driven process chains. **Information and Software Technology**, v. 41, n. 10, p. 639-650, July, 1999.
- _____. Verification of workflow nets. In: AZEMA, P.; BALBO, G. (Eds.). **Application and theory of petri nets**. Berlin: Springer-Verlag, 1997. (Lectures Notes in Computer Science).
- AALST, W. M. P. V. D.; HOFSTEDDE, A. H. M. T. Verification of workflow task structures a petri-net-based approach. **Information Systems**, 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.
- BUBENKO JR., J. A.; STIRNA, J.; BRASH, D. **EKD user guide, Dpt of computer and systems sciences**. Stockholm: Royal Institute of Technology, 1998.
- CHRZASTOWSKI-WACHTEL, P. et al. A top-down petri net-based approach for dynamic workflow modeling. In: INTERNATIONAL CONFERENCE BPM, 2003, Eindhoven. **Proceeding...**Berlin: Springer, 2003, p. 336-353.
- 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, 2001, Salvador. **Anais...** Salvador, 2001. CD-ROM.
- DEHNERT, J. Four steps towards sound business process models. In: EHRIG., H. et al. (Eds.). **Petri net technology for communication-based systems- advances in petri nets**. Berlin: Springer, 2003. p. 66-82. (Lecture Notes in Computer Science, 2472).
- 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.
- ELEKTRA - Electrical Enterprise Knowledge for Transforming Applications. **The ELEKTRA project programme**. Disponível em: <www.singular.gr/elektra.ekd.htm>. Acesso em: 27 Nov. 2000.
- GRIGOROVA, K. Process modelling using petri nets. In: INTERNATIONAL CONFERENCE ON COMPUTER SYSTEMS AND TECHNOLOGIES: E-Learning, 4, 2003, Rouse. 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.
- INAMASU, R. Y. **Modelo de FMS: uma plataforma para simulação e planejamento**. São Carlos, 1995. 134p. Tese de Doutorado - Escola de Engenharia de São Carlos, Universidade de São Paulo.
- JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **The unified software development process**. Reading: Addison-Wesley, 1999.
- JENSEN, K. A Brief Introduction to Coloured Petri Nets. In: Brinksma, E. Lecture Notes in Computer Science, Vol. 1217: Tools and Algorithms for the Construction and Analysis of Systems. TACAS'97 Workshop, Enschede, The Netherlands, 1997, p. 201-208. **Proceedings...** The Netherlands: Springer-Verlag, 1997.
- 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, 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>. Acesso em: 16 apr. 2001.
- KOUBARAKIS, M.; PLEXOUSAKIS, D. A formal framework for business process modelling and design. **Information Systems**, v. 27, n. 5, p. 299-319, jul. 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: CHEUNG, W.; HSU, J. (Eds.). IEEE conference on e-technology, e-commerce and e-service, 2, Hong-Kong, 2005, p. 89-92. **Proceedings...** Hong – Kong, 2005.
- MEVIUS, M.; OBERWEIS, A. A Petri-Net based approach to performance management of collaborative business processes. In: International Workshop on Database and Expert Systems

- Applications (DEXA'05), 16, 2005. **Proceeding ...** Karlsruhe: University of Karlsruhe, 2005. 987-991.
- 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, 2000. p. 254-273. (Lectures Notes in Computer Sciences, 1806).
- MURATA, T. 1989. Petri net: properties, analysis and applications. **Proceedings of the IEEE**, v. 77, n. 4, p. 541-579.
- NURCAN, A.; ROLLAND, C. A multi-method for defining the organizational change. **Journal of Information and Software Technology**, v. 45, n. 2, p. 61-82, feb. 2003.
- NURCAN, S. Analysis and design of co-operative work process a framework. **Information and Software Technology**, v. 40, n. 3, p. 143-156, jun. 1998.
- NURCAN, S.; BARRIOS, J. Enterprise knowledge and information system modelling in a evolving environment. In: INTERNATIONAL WORKSHOP ON ENGINEERING METHODS TO SUPPORT INFORMATION SYSTEMS EVOLUTION IN CONJUNCTION WITH, 2003, Geneva. **Proceedings...** Geneva: Switzerland, 2003. Disponível em <<http://cui.unige.ch/db-research/EMSISE03/Rp07.pdf>>. Acesso em: 11 apr. 2008.
- 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**. São Carlos, 2001. 145 p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.
- _____. **Método de avaliação do modelo de processos de negócios**. São Carlos, 2004. 252 p. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.
- 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, maio-ago. 2004a.
- 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, 2003. Disponível em: <http://www.bauru.unesp.br/acontece/simpep.html>. Acesso em: 30 nov. 2003.
- _____. 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, abr. 2004b.
- _____. Redes de petri aplicadas aos sistemas de gerenciamento de Workflow. In: Encontro Nacional de Engenharia de Produção, 12, 2002, Curitiba. **Anais...** Curitiba, 2002. CD-ROM.
- PERSOON, 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, 2000. p. 978-982.
- PETERSON, J. L. **Petri nets: theory and modelling of systems**. Englewood Cliffs: Prentice-Hall, 1981.
- RINDERLE, S.; REICHERT, M.; DADAM, P. Evaluation of correctness criteria for dynamic workflow changes. In: INTERNATIONAL CONFERENCE BPM 2003, Eindhoven. **Proceedings...** Berlin: Springer, 2003, p.41-57. (Lecture Notes in Computer Science, 2678).
- 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, apr. 2000.
- SALIMIFARD, S.; WRIGHT M. Petri net based modelling of workflow systems: an overview. **European Journal of Operational Research**, v.134, n. 3, p. 664-676, nov. 2001.
- SOARES, J. B. **Editor de modelos de sistemas de eventos discretos, baseado em redes de Petri interpretadas**. São Carlos, 2001. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.
- 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**, v. 50, n. 3, p. 294-314, 2007.
- VERBEEK, H. M. W.; BASTEN, T.; AALST, W. M. P. **Diagnosing workflow using woflan**. Eindhoven: Eindhoven University of Technology, 2002. (BETA Working Paper Series, WP 48).
- 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, 2000, p. 184-200. (Lectures Notes in Computer Sciences, 1806).
- WORKFLOW management coliation: reference model. Hampshire, 1996. (Document Number WFMC-TC00-1003).
- 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. **Proceeding...** Bangkok: IEEE Computer Society, 2006. p. 1-6.
- ZISMAN, M. D. **Representation, specification and automation of office procedures**. Pennsylvania, 1977. Thesis (PhD) - University of Pennsylvania, Wharton School of Business.

Sobre os autores

Sílvia Inês Dallavalle de Pádua

Faculdade de Economia, Administração e Contabilidade de Ribeirão Preto, Universidade de São Paulo – FEA-RP/USP, Av. Bandeirantes, 3900, CEP 14040-900, Monte Alegre, Ribeirão Preto, SP, e-mail: dallavalle@fearp.usp.br

Ricardo Yassushi Inamasu

EMBRAPA – Centro Nacional de Pesquisa e Desenvolvimento de Instrumentação Agropecuária, Rua XV de Novembro, 1452, CEP 13560-970, São Carlos, SP, e-mail: ricardo@cnpdia.embrapa.br

Agradecimentos: Os autores deste trabalho agradecem especialmente ao Prof. Dr. Adenildo da Silva Simão, à CAPES, EMBRAPA e FIPAII.

Recebido em 08/11/2007

Aceito em 27/10/2008