Análise qualitativa e quantitativa de WorkFlow nets utilizando Lógica Linear

Lígia Maria Soares Passos[1], **Stéphane Julia**[1]

[1]Faculdade de Computação – Universidade Federal de Uberlândia (UFU)
38400-902 – Uberlândia – MG – Brazil

ligia@pos.facom.ufu.br, stephane@facom.ufu.br

Abstract. *This article presents a method for qualitative and quantitative analysis of WorkFlow nets based on the proof trees of linear logic. The qualitative analysis is concerned with the proof of Soundness correctness criterion for WorkFlow nets. The quantitative analysis is concerned with the resource planning for each task of the workflow process and is based on the computation of symbolic date intervals for task execution.*

Resumo. *Este artigo apresenta um método para a análise qualitativa e quantitativa de WorkFlow nets baseada na construção de árvores de prova canônica da lógica linear. A análise qualitativa proposta neste trabalho diz respeito à prova do critério de corretude para WorkFlow nets denominado Soundness. A análise quantitativa basea-se no cálculo de intervalos de datas simbólicas para a execução de cada tarefa do processo de workflow modelado, possibilitando, assim, o planejamento dos recursos a serem utilizados em todas as tarefas do processo.*

1. Introdução

O principal objetivo dos Sistemas de Gerenciamento de Workflow é executar processos de workflow. Os processos de workflow representam as sequências de atividades que devem ser executadas em uma organização para tratar casos específicos e alcançar uma meta bem definida [van der Aalst and van Hee 2004].

De acordo com [Murata 1989], as redes de Petri são apropriadas para modelar Sistemas de Tempo Real, uma vez que elas permitem uma boa representação de situações de conflito, compartilhamento de recursos, comunicação síncrona e assíncrona, restrições de precedência e restrições de tempo explícito no caso das redes de Petri temporizadas.

Muitos artigos já consideraram a teoria das redes de Petri como uma ferramenta eficiente na modelagem e análise de Sistemas de Gerenciamento de Workflow. Em [van der Aalst 1998] e em [van der Aalst and van Hee 2004], por exemplo, são definidas as WorkFlow nets, que são redes de Petri que modelam processos de workflow. Em [Kotb and Badreddin 2005], é definido um modelo estendido de redes de Petri para a modelagem de workflow. Tal modelo permite o tratamento de recursos críticos que devem ser utilizados em atividades específicas em tempo real. Em [Ling and Schmidt 2000], uma extensão das WorkFlow nets é apresentada. Este modelo é chamado de Time WorkFlow net e associa intervalos de tempo às transições do modelo de rede de Petri correspondente. Em particular, um exemplo de sistema de saúde é usado para ilustrar uma abordagem onde uma enfermeira que cuida de dois pacientes é representada por uma ficha simples em um

lugar compartilhado. Em [Vilallonga et al. 2003], especificações de "clock" são combinadas com a teoria das redes de Petri para a especificação de requisitos temporais em Processos de Negócios. Esta nova formalização permite algumas verificações temporais que dependem da construção de um grafo das marcações acessíveis. Em [Lin and Qu 2004], a análise quantitativa proposta é baseada em reduções de padrões básicos de workflow. Reduções sucessivas levam à perda de informação em relação ao tempo de execução de cada tarefa, individualmente. Assim, é possível responder quando o processo global finalizará. No entanto, não é possível responder questões sobre quando cada tarefa será executada e, consequentemente, planejar a utilização de recursos envolvidos nestas.

Há várias técnicas para analisar processos de workflow. [van der Aalst and van Hee 2004] definem duas metodologias de análise distintas: *análise qualitativa* e *análise quantitativa*. A análise qualitativa preocupa-se com a corretude de uma WorkFlow net, enquanto a análise quantitativa preocupa-se com a performance e dimensionamento de recursos. Para a análise qualitativa três métodos são apresentados: o primeiro método é baseado na construção de grafos da alcançabilidade, o segundo é baseado na análise das propriedades *liveness* e *boundedness* para uma rede "short-circuited" e o terceiro é baseado na substituição de blocos bem formados. Para a análise quantitativa, três métodos são citados: análises Markovianas, teoria das filas e simulação (para maiores detalhes, ver [Marsan et al. 1995]).

Segundo [van der Aalst and van Hee 2004], o problema de planejamento de recursos tem como meta mostrar quais recursos e de que tipo são necessários considerando um dado período de tempo. Em particular, é importante encontrar um equilíbrio entre os recursos requisitados e os disponíveis.

Neste artigo, uma abordagem baseada na lógica linear é proposta para analisar qualitativamente e quantitativamente uma WorkFlow net. A análise qualitativa refere-se à prova do critério de corretude *Soundness* definido para WorkFlow nets [van der Aalst 1998]. Já a análise quantitativa preocupa-se com o planejamento de recursos para cada atividade de um processo mapeado em uma t-Time WorkFlow net.

Na sessão 2 são apresentadas as WorkFlow nets, o critério de corretude *Soundness* e as t-Time WorkFlow nets. Na sessão 3 é mostrada uma visão geral sobre a lógica linear. As duas análises, qualitativa e quantitativa, propostas neste artigo são apresentadas nas sessões 4 e 5, respectivamente. Finalmente, a última sessão conclui este trabalho com um pequeno resumo, uma avaliação sobre a abordagem proposta e uma perspectiva sobre os trabalhos futuros.

2. WorkFlow nets

Uma rede de Petri que modela um processo de workflow é uma WorkFlow net [van der Aalst and van Hee 2004],[van der Aalst 1998]. Uma WorkFlow net satisfaz as seguintes propriedades [van der Aalst 1998]:

- Tem apenas um lugar de início (denominado *Start*) e apenas um lugar de término (denominado *End*), sendo estes dois lugares tratados como lugares especiais; o lugar *Start* tem apenas arcos de saída e o lugar *End* apenas arcos de entrada.
- Uma ficha em *Start* representa um caso que precisa ser tratado e uma ficha em *End* representa um caso que foi tratado.

- Toda tarefa t (transição) e condição p (lugar) deve estar em um caminho que se encontra entre o lugar *Start* e o lugar *End*.

Soundness é o critério de corretude definido para WorkFlow nets. Uma WorkFlow net é *Sound* se, e somente se, os três requisitos abaixo são satisfeitos:

- Para cada ficha colocada no lugar de início, uma (e apenas uma) ficha eventualmente aparecerá no lugar de término.
- Quando uma ficha aparece no lugar de término, todos os outros lugares estão vazios, considerando o caso em questão.
- Considerando uma tarefa associada à uma transição, é possível evoluir da marcação inicial até uma marcação que sensibiliza tal transição, ou seja, não deve haver nenhuma transição morta na WorkFlow net.

2.1. Processos

Um processo define quais tarefas precisam ser executadas e em qual ordem a execução deve ocorrer. Modelar um processo de workflow em termos de uma WorkFlow net é bem direto: transições são componentes ativos e modelam as tarefas, lugares são componentes passivos e modelam as condições (pré e pós) e as fichas modelam os casos [van der Aalst 1998]. Para ilustrar o mapeamento de processos em WorkFlow nets, considera-se o processo de tratamento de reclamações mostrado em [van der Aalst and van Hee 2004]: uma reclamação é inicialmente gravada. Então, o cliente que efetuou a reclamação e o departamento responsável pela reclamação são contactados. O cliente é questionado para maiores informações. O departamento é informado sobre a reclamação. Estas duas tarefas podem ser executadas em paralelo, isto é, simultaneamente ou em qualquer ordem. Depois disso, os dados são recolhidos e uma decisão é tomada. Dependendo da decisão, ou um pagamento de compensação é efetuado, ou uma carta é enviada. Finalmente, a reclamação é armazenada. A Figura 1 mostra a WorkFlow net que representa este processo.

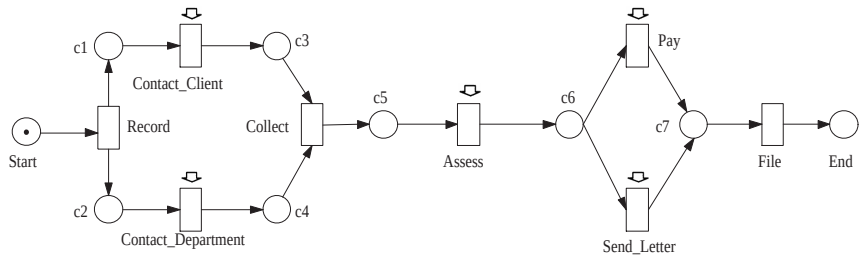

Figura 1. O processo de tratamento de reclamações e os seus acionamentos

2.1.1. Roteamentos

De acordo com [van der Aalst and van Hee 2004], pela rota de um caso, ao longo de uma série de tarefas, pode-se determinar quais tarefas precisam ser executadas (e em que ordem). Quatro construções básicas para o roteamento de tarefas são consideradas: sequencial, paralela, condicional (rota seletiva) e iterativa.

Na abordagem proposta, uma rota iterativa será substituída por uma tarefa global. Na verdade, na prática, se um processo de workflow deve respeitar um limite de tempo, ele não poderá repetir indefinidamente uma mesma atividade. A estrutura hierárquica das redes de Petri, baseadas no conceito de "Well-formed block" [Valette 1979], pode ser utilizada para representar uma rota iterativa através de uma única tarefa. Assim, uma duração máxima será associada a esta tarefa, a fim de especificar, de modo implícito, o número de vezes que a tarefa poderá ser executada dentro do bloco antes de ser detectado um problema na execução desta atividade.

2.1.2. Acionamentos

Um acionamento é uma condição externa que guia a execução de uma tarefa sensibilizada [van der Aalst 1998]. Há quatro tipos distintos de tarefas: *Usuário*: uma tarefa é acionada por um recurso humano e este acionamento é mostrado em uma WorkFlow net através do símbolo ⇩ nas transições; *Mensagem*: um evento externo aciona uma tarefa sensibilizada, o que é mostrado em uma WorkFlow net através do símbolo ✉ nas transições; *Tempo*: uma tarefa sensibilizada é acionada por um relógio, isto é, a tarefa é executada em um tempo pré-definido. Isto é mostrado em uma WorkFlow net através do símbolo ⏲ nas transições; *Automática*: uma tarefa é acionada no momento em que é sensibilizada e não requer interação humana. Para este tipo de tarefa não há nenhuma representação na WorkFlow net.

Nota-se que quando uma tarefa do tipo *Usuário* é considerada, uma tarefa é acionada por um recurso humano, isto é, há uma alocação de recurso associada a esta tarefa. Nos demais tipos de tarefa não há alocação de recursos.

2.2. t-Time WorkFlow nets

De acordo com [van der Aalst and van Hee 2004], dado um processo modelado por uma rede de Petri, realizar afirmações sobre sua performance é sempre desejado. Para estar apto a responder questões como o tempo esperado de conclusão de um processo e a capacidade de recursos requerida é necessário incluir informações pertinentes sobre o tempo neste modelo de processo. As redes de Petri clássicas não permitem a modelagem do tempo, mas estas podem ser estendidas em relação ao tempo.

As redes de Petri temporais e temporizadas podem associar tempo aos lugares (redes de Petri p-temporais e p-temporizadas) ou às transições (redes de Petri t-temporais e t-temporizadas). Nas redes de Petri p-temporizadas [Sifakis 1977], o tempo é representado por durações (números racionais positivos ou nulo) associadas aos lugares. Nas redes de Petri t-temporizadas [Ramchandani 1974], o tempo é representado por durações (números reais positivos ou nulo) associadas às transições. Nas redes de Petri t-temporais [Merlin 1974], o tempo é representado por um intervalo $[\theta_{min}, \theta_{max}]$ associado a cada transição, de forma que este intervalo de tempo corresponde a uma duração de sensibilização. Assim, o intervalo $[8, 12]$ indica que a transição irá disparar pelo menos oito unidades de tempo após a transição ter sido sensibilizada e no máximo doze unidades de tempo após a sensibilização desta transição. Nas redes de Petri p-temporais [Khansa 1997], é associado um intervalo de tempo $[\theta_{min}, \theta_{max}]$ aos lugares.

No contexto das redes de Petri t-temporais, uma t-Time WorkFlow net será então

uma WorkFlow net estendida com intervalos de tempo associados às transições, já que a execução das tarefas ficará associada às transições do modelo.

Definição 1. *Uma t-Time WorkFlow net N é uma quádrupla (P, T, F, I) tal que:*

- (P, T, F) *é uma WorkFlow net, onde:*
  - P *é um conjunto finito de lugares,*
  - T *é um conjunto finito de transições e*
  - $F \subseteq (P \times T) \cup (T \times P)$ *é um conjunto de arcos (relação de fluxo).*
- I *é uma aplicação que associa a cada transição* $t \in T$ *um intervalo de sensibilização* $I(t) = \left[\theta_{mim(t)}, \theta_{max(t)}\right]$*, onde* $\theta_{mim(t)}$ *representa o tempo mínimo de disparo da transição e* $\theta_{max(t)}$ *representa o tempo máximo de disparo da transição* t*.*

3. Lógica Linear

A lógica linear foi proposta em 1987 por Girard em [Girard 1987]. Na lógica linear, as proposições são consideradas como recursos que são consumidos ou produzidos a cada mudança de estado [Pradin-Chezalviel et al. 1999]. A lógica linear introduz novos conectivos, como os conectivos *"par"*($\parr$), *"times"*($\otimes$), *"with"*($\&$), *"plus"*($\oplus$) e *implicação linear*($\multimap$). Para maiores detalhes, pode-se consultar [Girard 1987] e [Pradin-Chezalviel et al. 1999]. Neste artigo apenas dois conectivos da lógica linear serão explorados:

- O conectivo *times*, denotado pelo símbolo $\otimes$, que representa a disponibilidade simultânea de recursos. Por exemplo, $A \otimes B$ representa a disponibilidade simultânea dos recursos A e B.
- O conectivo *implicação linear*, denotado pelo símbolo $\multimap$, representa uma mudança de estado. Por exemplo, $A \multimap B$ denota que consumindo A, B é produzido. Pode-se notar que após a produção de B, A não estará mais disponível.

De acordo com [Pradin-Chezalviel et al. 1999], algumas dificuldades são conhecidas no contexto do uso das lógicas modal e temporal quando durações explícitas são associadas ao disparo das transições, como a ocorrência da explosão do espaço de estados e o fato das durações serem delimitadas por valores, ou seja, não oferece computação simbólica. Por estas razões, a abordagem apresentada neste artigo utiliza a lógica linear.

A tradução de uma rede de Petri em fórmulas da lógica linear é apresentada em [Pradin-Chezalviel et al. 1999]. Uma marcação M é um monômio em $\otimes$, ou seja, uma marcação é representada por $M = A_1 \otimes A_2 \otimes \ldots \otimes A_k$ onde A_i são nomes de lugares. Uma transição é uma expressão da forma $M_1 \multimap M_2$ onde M_1 e M_2 são marcações. Um sequente $M, t_i \vdash M'$ representa um cenário onde M e M' são, respectivamente, a marcação inicial e final e t_i é uma lista de transições não ordenadas [Julia and Soares 2003]. Um sequente pode ser provado aplicando regras do cálculo de sequentes. De acordo com [Girault 1997], há equivalência entre a prova de sequentes da lógica linear e a alcançabilidade de uma rede de Petri. O sistema de dedução linear é similar ao sistema de dedução de Gentzen, que foi proposto em 1934 [Gochet and Gribomont 1990]. Uma árvore de prova da lógica linear é lida de baixo para cima (bottom-up) e termina quando todas as folhas forem sequentes identidade (sequentes do tipo $A \vdash A$).

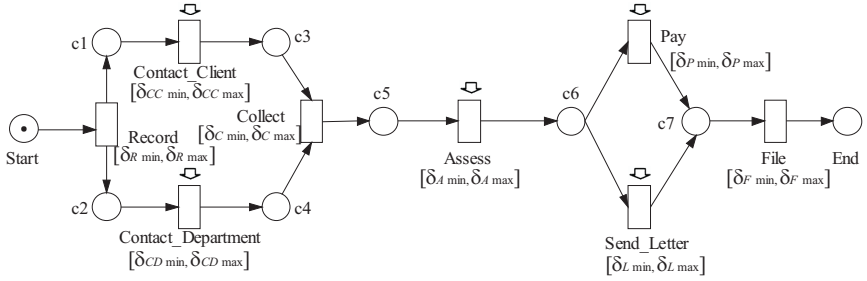

Figura 2. O processo de tratamento de reclamações, seus acionamentos e intervalos de tempo simbólico associados às tarefas

Neste artigo, serão consideradas apenas algumas regras da lógica linear. Estas regras serão utilizadas na construção das árvores de prova canônica. Para tanto, considere que F, G e H são fórmulas e que Γ e Δ são blocos de fórmulas da lógica linear. As regras são apresentadas abaixo [Riviere et al. 2001]:

- A regra $\multimap_L$, dada por $\dfrac{\Gamma \vdash F \quad \Delta, G \vdash H}{\Gamma, \Delta, F \multimap G \vdash H} \multimap_L$, expressa o disparo de uma transição e gera dois sequentes, tal que o sequente à direita representa o subsequente restante a ser provado e o sequente à esquerda representa as fichas consumidas pelo disparo da transição.
- A regra $\otimes_L$, dada por $\dfrac{\Gamma, F, G \vdash H}{\Gamma, F \otimes G \vdash H} \otimes_L$, transforma uma marcação em uma lista de átomos.
- A regra $\otimes_R$, dada por $\dfrac{\Gamma \vdash F \quad \Delta \vdash G}{\Delta, \Gamma \vdash F \otimes G} \otimes_R$, transforma um sequente do tipo $A, B \vdash A \otimes B$ em dois sequentes identidade $A \vdash A$ e $B \vdash B$.

Em uma árvore de prova, cada disparo de transição pode gerar uma data simbólica associada a cada átomo (token) como mostrado em [Riviere et al. 2001]. Neste artigo, D_i denota uma data e d_i uma duração associada a um disparo de uma transição (t_i). Um par (D_p, D_c) será associado a cada átomo da árvore de prova, tal que D_p e D_c representam, respectivamente, as datas de produção e consumo de um átomo. O cálculo de datas em árvores de prova canônica é dado pelos seguintes passos:

- Determine uma data de produção D_i para todas as fichas da marcação inicial;
- para cada instância da regra $\multimap_L$, calcule a data de disparo desta transição: isto é igual ao maior valor de data de produção dos átomos consumidos por esta transição, acrescido pela duração de sensibilização d_i associada à transição considerada;
- atualizar as datas de todos os átomos que foram consumidos e produzidos.

Em um modelo de rede de Petri t-temporal, toda duração de sensibilização d_i de uma transição t_i tem um valor que pertence a um intervalo de tempo $\Delta_i = \left[\delta_{imin}, \delta_{imax}\right]$. Logo, uma vez que as datas simbólicas computadas dependem de d_i, seus domínios também serão em função de intervalos de tempo.

4. Análise Qualitativa: verificação da propriedade *Soundness*

No contexto das WorkFlow nets, o objetivo principal da análise qualitativa é provar o critério de corretude para WorkFlow nets denominado *Soundness*. Para isso, é necessário

provar sequentes lineares que representem a WorkFlow net e analisar as árvores de prova construídas para provar estes sequentes. Inicialmente, se a WorkFlow net tiver mais de uma rota (ou seja, uma ou mais rotas condicionais), é necessário construir e provar um sequente linear para cada cenário da WorkFlow net. Por exemplo, na Figura 1, há dois cenários diferentes: o primeiro cenário, Sc_1, onde a tarefa *Pay* será executada, isto é, a transição *Pay* será disparada, e o segundo cenário, Sc_2, onde a tarefa *Send_Letter* será executada. Assim, para esta WorkFlow net, dois sequentes lineares deverão ser provados. Cada sequente linear considera apenas uma rota da WorkFlow net analisada. Por exemplo, para o cenário Sc_1, é necessário provar o sequente:

$$Start,Record,Contact_Client,Contact_Department,Collect,Assess,Pay,File \vdash End$$

e para o cenário Sc_2 é necessário provar o sequente:

$$Start,Record,Contact_Client,Contact_Department,Collect,Assess,Send_Letter,File \vdash End$$

onde: $Record = t_1 = Start \multimap c1 \otimes c2$, $Contact_Client = t_2 = c1 \multimap c3$, $Contact_Department = t_3 = c2 \multimap c4$, $Collect = t_4 = c3 \otimes c4 \multimap c5$, $Assess = t_5 = c5 \multimap c6$, $Pay = t_6 = c6 \multimap c7$, $Send_Letter = t_7 = c6 \multimap c7$ e $File = t_8 = c7 \multimap End$.

Para provar estes sequentes, árvores de prova da lógica linear, sem cálculo de datas serão utilizadas. Nota-se que cada sequente linear tem apenas um átomo $Start$ que representa a marcação inicial da WorkFlow net. Para analisar o critério de corretude *Soundness*, apenas uma ficha no lugar *Start* é necessária e suficiente. Assim, o método de prova proposto neste artigo sempre considera apenas uma ficha no lugar *Start*. Na sequência são mostradas as árvores de prova, sem cálculo de datas, para os cenários Sc_1 e Sc_2. Árvore de prova para o cenário Sc_1:

$$\dfrac{\dfrac{Start \vdash Start \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{c1 \vdash c1 \quad \dfrac{c2 \vdash c2 \quad \dfrac{\dfrac{c3 \vdash c3 \quad c4 \vdash c4}{c3,c4 \vdash c3 \otimes c4}\otimes_R \quad \dfrac{c5 \vdash c5 \quad \dfrac{c6 \vdash c6 \quad \dfrac{c7 \vdash c7 \quad End \vdash End}{c7, c7 \multimap End \vdash End}\multimap_L}{c6, c6 \multimap c7, t_8 \vdash End}\multimap_L}{c5, c5 \multimap c6, t_6, t_8 \vdash End}\multimap_L}{c3, c4, c3 \otimes c4 \multimap c5, t_5, t_6, t_8 \vdash End}\multimap_L}{c2, c3, c2 \multimap c4, t_4, t_5, t_6, t_8 \vdash End}\multimap_L}{c1, c2, c1 \multimap c3, t_3, t_4, t_5, t_6, t_8 \vdash End}\otimes_L}{c1 \otimes c2, c1 \multimap c3, t_3, t_4, t_5, t_6, t_8 \vdash End}\multimap_L}{Start, Start \multimap c1 \otimes c2, t_2, t_3, t_4, t_5, t_6, t_8 \vdash End}$$

Árvore de prova para o cenário Sc_2:

$$\dfrac{\dfrac{Start \vdash Start \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{c1 \vdash c1 \quad \dfrac{c2 \vdash c2 \quad \dfrac{\dfrac{c3 \vdash c3 \quad c4 \vdash c4}{c3,c4 \vdash c3 \otimes c4}\otimes_R \quad \dfrac{c5 \vdash c5 \quad \dfrac{c6 \vdash c6 \quad \dfrac{c7 \vdash c7 \quad End \vdash End}{c7, c7 \multimap End \vdash End}\multimap_L}{c6, c6 \multimap c7, t_8 \vdash End}\multimap_L}{c5, c5 \multimap c6, t_7, t_8 \vdash End}\multimap_L}{c3, c4, c3 \otimes c4 \multimap c5, t_5, t_7, t_8 \vdash End}\multimap_L}{c2, c3, c2 \multimap c4, t_4, t_5, t_7, t_8 \vdash End}\multimap_L}{c1, c2, c1 \multimap c3, t_3, t_4, t_5, t_7, t_8 \vdash End}\otimes_L}{c1 \otimes c2, c1 \multimap c3, t_3, t_4, t_5, t_7, t_8 \vdash End}\multimap_L}{Start, Start \multimap c1 \otimes c2, t_2, t_3, t_4, t_5, t_7, t_8 \vdash End}$$

Quando todas as árvores de prova para a WorkFlow net analisada estão construídas, elas deverão ser analisadas seguindo os seguintes passos:

1. Para cada árvore de prova construída verificar se:
   (a) Apenas um átomo End foi produzido (isto é representado, na árvore de prova, pelo sequente identidade $End \vdash End$). Este fato prova o primeiro requisito para a verificação da propriedade *Soundness*, isto é, que apenas uma ficha aparece no lugar *End* para o caso tratado.
   (b) Quando a prova está finalizada, isto é, quando há apenas sequentes identidade na árvore de prova, não há nenhum átomo disponível para consumo na árvore de prova. Este fato prova que todos os lugares da WorkFlow net estão vazios, ou seja, o segundo requisito para afirmar que uma WorkFlow net é *Soundness*.
2. Considerando todos os cenários $Sc_1, Sc_2, ..., Sc_n$ da WorkFlow net analisada, cada transição $t \in T$ precisa aparecer em um cenário, pelo menos. Isto prova que todas as transições são disparadas, ou seja, nenhuma transição é morta.

Se as condições 1 e 2 acima são satisfeitas, a WorkFlow net analisada é *Sound*.

Considerando as árvores de prova construídas para os cenários Sc_1 e Sc_2, a WorkFlow net mostrada na Figura 1 é *Sound*, uma vez que as três regras para *Soundness* são satisfeitas:

- Apenas um átomo End foi produzido em cada cenário.
- Considerando as árvores de prova finalizadas, não há nenhum átomo disponível para consumo nestas árvores de prova.
- Considerando os cenários Sc_1 e Sc_2 da WorkFlow net analisada, cada transição desta WorkFlow net aparece em pelo menos um cenário e cada uma destas transições foi disparada. Assim, é possível afirmar que não há nenhuma transição morta nesta WorkFlow net.

5. Análise Quantitativa: planejamento de recursos

O principal objetivo do planejamento de recursos, apresentado neste artigo, é calcular janelas de datas onde poderá ocorrer a utilização de um recurso (pessoa, equipe, empresa terceirizada etc.) para o tratamento de atividades. Assim, será possível prever a disponibilidade dos recursos envolvidos na execução das tarefas correspondentes. No caso das WorkFlow nets analisadas através de árvores de prova canônica da lógica linear, as datas de execução das tarefas poderão ser dadas através de datas simbólicas, ao invés de datas numéricas. A maior vantagem da utilização de datas simbólicas é que quando estas já estiverem calculadas poderão ser utilizadas diretamente para qualquer caso que será tratado pelo processo de workflow analisado – mesmo considerando casos cujo instante de iniciação e/ou duração global são diferentes. Através das árvores de prova, datas simbólicas para a execução de tarefas podem ser derivadas. A análise quantitativa também considera todos os possíveis cenários de uma t-Time WorkFlow net. Isto é necessário uma vez que o planejamento de recursos precisa calcular a data de utilização dos recursos associados a cada tarefa presente no processo de workflow analisado. Como exemplo, considera-se a t-Time WorkFlow net mostrada na Figura 2. Para realizar o planejamento de recursos para esta WorkFlow net é necessário construir árvores de prova canônica com cálculo de datas para os cenários Sc_1 e Sc_2, os mesmos cenários da análise qualitativa.

Para tanto, considera-se que: $Seq = D_S + d_R + max(d_{CC}, d_{CD}) + d_C$. As árvores de prova canônica, com cálculo de datas, que correspondem aos cenários Sc_1 e Sc_2 são dadas por:

Árvore de prova, com cálculo de datas, para o cenário Sc_1:

$$
\begin{array}{rl}
\dfrac{c7(Seq+d_A+d_P,Seq+d_A+d_P+d_F)\vdash c7 \qquad End(Seq+d_A+d_P+d_F,.)\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c6(Seq+d_A,Seq+d_A+d_P)\vdash c6 \qquad c7(Seq+d_A+d_P,.),c7\multimap End\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c5(Seq,Seq+d_A)\vdash c5 \qquad c6(Seq+d_A,.),c6\multimap c7,t_8\vdash End}{} & \multimap_L \\[2ex]
\dfrac{\dfrac{c3(D_S+d_R+d_{CC},Seq)\vdash c3 \qquad c4(D_S+d_R+d_{CD},Seq)\vdash c4}{c3(D_S+d_R+d_{CC},Seq),c4(D_S+d_R+d_{CD},Seq)\vdash c3\otimes c4}\otimes_R \qquad c5(Seq,.),c5\multimap c6,t_6,t_8\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c2(D_S+d_R,D_S+d_R+d_{CD})\vdash c2 \qquad c3(D_S+d_R+d_{CC},.),c4(D_S+d_R+d_{CD},.),c3\otimes c4\multimap c5,t_5,t_6,t_8\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c1(D_S+d_R,D_S+d_R+d_{CC})\vdash c1 \qquad c2(D_S+d_R,.),c3(D_S+d_R+d_{CC},.),c2\multimap c4,t_4,t_5,t_6,t_8\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c1(D_S+d_R,.),c2(D_S+d_R,.),c1\multimap c3,t_3,t_4,t_5,t_6,t_8\vdash End}{} & \otimes_L \\[2ex]
\dfrac{Start(D_S,D_S+d_R)\vdash Start \qquad c1(D_S+d_R,.)\otimes c2(D_S+d_R,.),t_2,t_3,t_4,t_5,t_6,t_8\vdash End}{} & \multimap_L \\[2ex]
Start(D_S,.),Start\multimap c1\otimes c2,t_2,t_3,t_4,t_5,t_6,t_8\vdash End &
\end{array}
$$

Árvore de prova, com cálculo de datas, para o cenário Sc_2:

$$
\begin{array}{rl}
\dfrac{c7(Seq+d_A+d_L,Seq+d_A+d_L+d_F)\vdash c7 \qquad End(Seq+d_A+d_L+d_F,.)\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c6(Seq+d_A,Seq+d_A+d_L)\vdash c6 \qquad c7(Seq+d_A+d_L,.),c7\multimap End\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c5(Seq,Seq+d_A)\vdash c5 \qquad c6(Seq+d_A,.),c6\multimap c7,t_8\vdash End}{} & \multimap_L \\[2ex]
\dfrac{\dfrac{c3(D_S+d_R+d_{CC},Seq)\vdash c3 \qquad c4(D_S+d_R+d_{CD},Seq)\vdash c4}{c3(D_S+d_R+d_{CC},Seq),c4(D_S+d_R+d_{CD},Seq)\vdash c3\otimes c4}\otimes_R \qquad c5(Seq,.),c5\multimap c7,t_7,t_8\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c2(D_S+d_R,D_S+d_R+d_{CD})\vdash c2 \qquad c3(D_S+d_R+d_{CC},.),c4(D_S+d_R+d_{CD},.),c3\otimes c4\multimap c5,t_5,t_7,t_8\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c1(D_S+d_R,D_S+d_R+d_{CC})\vdash c1 \qquad c2(D_S+d_R,.),c3(D_S+d_R+d_{CC},.),c2\multimap c4,t_4,t_5,t_7,t_8\vdash End}{} & \multimap_L \\[2ex]
\dfrac{c1(D_S+d_R,.),c2(D_S+d_R,.),c1\multimap c3,t_3,t_4,t_5,t_7,t_8\vdash End}{} & \otimes_L \\[2ex]
\dfrac{Start(D_S,D_S+d_R)\vdash Start \qquad c1(D_S+d_R,.)\otimes c2(D_S+d_R,.),t_2,t_3,t_4,t_5,t_7,t_8\vdash End}{} & \multimap_L \\[2ex]
Start(D_S,.),Start\multimap c1\otimes c2,t_2,t_3,t_4,t_5,t_7,t_8\vdash End &
\end{array}
$$

Para cada tarefa do tipo usuário modelada na WorkFlow net analisada, deve-se extrair as datas de produção, D_P, e consumo, D_C, do átomo que representa a pré-condição da transição correspondente a esta tarefa. Quando há mais de uma pré-condição associada à transição, considera-se a data máxima das produções dos átomos correspondentes a estas pré-condições. A data de produção deste átomo, D_P, corresponde ao início da execução da tarefa associada à transição e a data de consumo, D_C, corresponde ao término da execução da mesma. Assim, é gerado um intervalo $[D_P, D_C]$ de datas onde o recurso que executará a referida tarefa poderá ser requisitado para realizá-la.

Uma vez que as datas de produção e consumo são dependentes de durações de sensibilização d_i, cujo valor pertence a um intervalo de tempo $\Delta_i = \left[\delta_{imin}, \delta_{imax}\right]$, pode-se considerar vários intervalos possíveis de execução das tarefas, de acordo com um planejamento estratégico. Por exemplo, o intervalo de execução $I_{Exec} = [D_{Pmin}, D_{Cmax}]$ considera que a alocação do recurso para a execução da tarefa poderá ocorrer entre o início ao mais cedo e o término ao mais tarde da atividade considerada. Este intervalo de datas é o mais flexível, no sentido de que ele considera a janela de tempo de utilização do recurso a mais extensa possível. De acordo com a disponibilidade do recurso envolvido, poderão ser consideradas na estratégia de escalonamento de atividades do processo de workflow

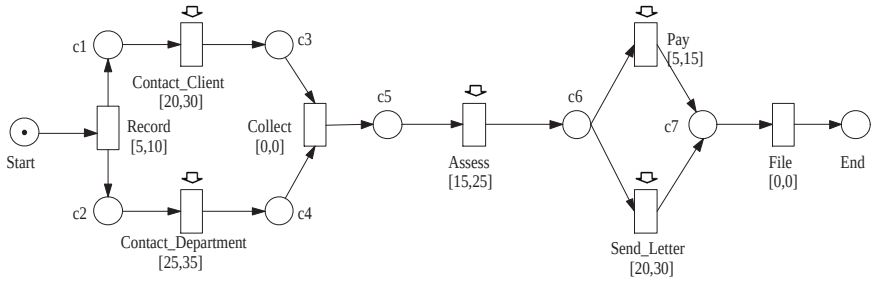

Figura 3. O processo de tratamento de reclamações, seus acionamentos e intervalos de tempo numérico associados às tarefas

outras datas fornecidas pela árvore de prova, como D_{Pmax} (data de início ao mais tarde da tarefa) ou D_{Cmin} (data de término ao mais cedo da tarefa).

A informação sobre quando o processo global finalizará, para um dado caso, é dado pela data de produção do átomo *End*. Assim, pode-se calcular a data ao mais cedo de finalização do processo, dada pela data mínima D_{Pmin} de produção deste átomo e a data ao mais tarde de finalização do processo, dada pela data máxima D_{Pmax} de produção do átomo *End*.

A Tabela 1 mostra os intervalos de datas simbólicas de execução para as tarefas que são do tipo usuário, dos cenários Sc_1 e Sc_2, considerando o intervalo de execução $I_{Exec} = [D_{Pmin}, D_{Cmax}]$. Os intervalos de datas calculados poderão ser utilizados por qualquer caso tratado pelo processo de workflow modelado pela t-Time WorkFlow net da Figura 2. A título de exemplificação, considera-se o processo de tratamento de reclamações mapeado na t-Time WorkFlow net mostrada na Figura 3. A única diferença entre as Figuras 2 e 3 é que a primeira associa intervalos de tempo simbólico às transições enquanto que a segunda associa às transições intervalos de tempo numérico. Assim, considerando os intervalos de tempo da t-Time WorkFlow net mostrada na Figura 3 e considerando que o processo a ser tratado inicia-se na data 0, ou seja, $D_S = 0$, os intervalos de datas para a execução das tarefas podem ser calculados apenas substituindo as datas simbólicas presentes na Tabela 1 pelas datas numéricas associadas às transições da t-Time WorkFlow net da Figura 3. A Tabela 2 mostra o resultado deste processo. Por exemplo, o recurso utilizado para executar a tarefa *Send_Letter* do cenário Sc_2 deverá ser alocado ao mais cedo na data 40 e deverá ser liberado ao mais tarde na data 100. Assim, o recurso que executará esta tarefa poderá ser alocado neste intervalo de datas.

6. Conclusão

Este artigo apresenta um método de análise qualitativa e quantitativa de WorkFlow nets baseado nas árvores de prova canônica da lógica linear. A análise qualitativa tem como objetivo provar o critério de corretude para WorkFlow nets denominado *Soundness* e a análise quantitativa basea-se no cálculo de datas simbólicas para o planejamento de recursos utilizados na realização de cada tarefa do processo de workflow.

As vantagens desta abordagem são diversas. O fato de trabalhar com lógica linear permite provar o critério de corretude *Soundness* sem que seja necessária a construção de um grafo das marcações acessíveis, considerando diretamente a própria estrutura da WorkFlow net. O cálculo de datas simbólicas correspondentes à execução de cada tarefa

Tabela 1. Intervalos de datas para execução de tarefas do tipo usuário dos cenários Sc_1 e Sc_2

Transição	Intervalos de data Sc1 e Sc2
$t_2{=}c1{\multimap}c3$	$[D_S{+}\delta_{Rmin}, D_S{+}\delta_{Rmax}{+}\delta_{CCmax}]$
$t_3{=}c2{\multimap}c4$	$[D_S{+}\delta_{Rmin}, D_S{+}\delta_{Rmax}{+}\delta_{CDmax}]$
$t_5{=}c5{\multimap}c6$	$[D_S{+}\delta_{Rmin}{+}max(\delta_{CCmin}, \delta_{CDmin}){+}\delta_{Cmin},$
	$D_S{+}\delta_{Rmax}{+}max(\delta_{CCmax}, \delta_{CDmax}){+}\delta_{Cmax}{+}\delta_{Amax}]$
Transição	**Intervalos de data Sc1**
$t_6{=}c6{\multimap}c7$	$[D_S{+}\delta_{Rmin}{+}max(\delta_{CCmin}, \delta_{CDmin}){+}\delta_{Cmin}{+}\delta_{Amin},$
	$D_S{+}\delta_{Rmax}{+}max(\delta_{CCmax}, \delta_{CDmax}){+}\delta_{Cmax}{+}\delta_{Amax}{+}\delta_{Pmax}]$
Transição	**Intervalos de data Sc2**
$t_7{=}c6{\multimap}c7$	$[D_S{+}\delta_{Rmin}{+}max(\delta_{CCmin}, \delta_{CDmin}){+}\delta_{Cmin}{+}\delta_{Amin},$
	$D_S{+}\delta_{Rmax}{+}max(\delta_{CCmax}, \delta_{CDmax}){+}\delta_{Cmax}{+}\delta_{Amax}{+}\delta_{Lmax}]$

Tabela 2. Intervalos de tempo numéricos para a execução de tarefas do tipo usuário dos cenários Sc_1 e Sc_2

Tarefa	Intervalos de data Sc1	Intervalos de data Sc2
$Contact_Client$	[5,40]	[5,40]
$Contact_Department$	[5,45]	[5,45]
$Assess$	[25,70]	[25,70]
Pay	[40,85]	—
$Send_Letter$	—	[40,100]

mapeada em uma t-Time WorkFlow net permite planejar a utilização dos recursos envolvidos nas atividades do processo de workflow, através de fórmulas que podem ser utilizadas por qualquer caso tratado pelo processo de workflow correspondente, sem que seja necessário percorrer novamente o processo de workflow inteiro para recalcular, para cada novo caso, datas de início e término das atividades envolvidas no processo.

Além disso, alguns provadores de teoremas da lógica linear automáticos, como LinTAP [Mantel and Otten 1999], Linprove [Tammet and Tammet 1994] e llprover [Tamura 1997], podem ser adaptados e utilizados para automatizar a prova dos sequentes lineares propostos neste artigo.

Como trabalhos futuros, seria interessante aplicar esta abordagem a um estudo de caso no contexto de planejamento de recursos em gestão de projetos de software e adaptar ou desenvolver um provador de teoremas automático para a prova dos sequentes lineares apresentados neste artigo.

Referências

Girard, J.-Y. (1987). Linear logic. *Theor. Comput. Sci.*, 50(1):1–102.

Girault, F. (1997). *A logic for Petri nets*, volume 31. Eddition Hermes.

Gochet, P. and Gribomont, P. (1990). *Logique: méthodes pour l'informatique fondamentale*, volume Vol 1. Hermès.

Julia, S. and Soares, M. S. (2003). Verification of real time uml specifications through a specialized inference mechanism based on a token player algorithm and the sequent calculus of linear logic. In *Proceedings of the 15th European Simulation Symposium and Exhibition*, pages 65–70, Delft, The Netherlands.

Khansa, W. (1997). *Réseaux de Petri p-temporels contribution à l'étude des systèmes à évenements discrets*. PhD thesis, Université de Savoie, France.

Kotb, Y. T. and Badreddin, E. (2005). Synchronization among activities in a workflow using extended workflow petri nets. In *CEC '05: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology*, pages 548–551, Washington, DC, USA. IEEE Computer Society.

Lin, C. and Qu, Y. (2004). Temporal inference of workflow systems based on time petri nets: Quantitative and qualitative analysis. *Int. J. Intell. Syst.*, 19(5):417–442.

Ling, S. and Schmidt, H. (2000). Time petri nets for workflow modelling and analysis. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, pages 3039–3044 vol.4.

Mantel, H. and Otten, J. (1999). lintap: A tableau prover for linear logic. In *TABLEAUX '99: Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 217–231, London, UK. Springer-Verlag.

Marsan, M. A., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G., Computing, P., Wiley, J., Almeida, V., Almeida, J., and Analysis, C. M. P. (1995). Modelling with generalized stochastic petri nets. In *Series in Parallel Computing*. Wiley.

Merlin, P. (1974). *A study of recoverability of computer systems*. PhD thesis, University of California, Irvine.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

Pradin-Chezalviel, B., Valette, R., and Kunzle, L. A. (1999). Scenario durations characterization of t-timed petri nets using linear logic. In *PNPM '99: Proceedings of the The 8th International Workshop on Petri Nets and Performance Models*, page 208, Washington, DC, USA. IEEE Computer Society.

Ramchandani, C. (1974). Analysis of asynchronous concurrent systems by timed petri nets. Technical report, Cambridge, MA, USA.

Riviere, N., Valette, R., Pradin-Chezalviel, B., and Ups, I. A. . (2001). Reachability and temporal conflicts in t-time petri nets. In *PNPM '01: Proceedings of the 9th international Workshop on Petri Nets and Performance Models (PNPM'01)*, page 229, Washington, DC, USA. IEEE Computer Society.

Sifakis, J. (1977). Use of petri nets for performance evaluation. In *Proceedings of the Third International Symposium on Measuring, Modelling and Evaluating Computer Systems*, pages 75–93, Amsterdam, The Netherlands, The Netherlands. North-Holland Publishing Co.

Tammet, T. and Tammet, T. (1994). Proof strategies in linear logic. *Journal of Automated Reasoning*, 12:273–304.

Tamura, N. (1997). A linear logic prover (llprover). Disponível em: http://bach.istc.kobe-u.ac.jp/llprover. Acesso em 26 Fev. 2009.

Valette, R. (1979). Analysis of Petri Nets by Stepwise Refinements. *Journal of Computer and System Sciences*, 18:35–46.

van der Aalst, W. M. P. (1998). The aplication of petri nets to workflow management. In *The Journal of Circuits,Systems and Computers*, pages 21–66.

van der Aalst, W. M. P. and van Hee, K. (2004). *Workflow Management: Models, Methods, and Systems*. The MIT Press.

Vilallonga, G., Riesco, D., Montejano, G., and Uzal, R. (2003). Petri nets with clocks for the analytical validation of business process. pages 11–25.