

Adaptatividade em Process Mining

S. G. Dada e J. José Neto

Resumo— O mapeamento de processos de negócio é fundamental para a eficiência de um serviço prestado. No mapeamento dos processos é possível descobrir quais tarefas são dependentes umas das outras e quais são gargalo da operação. Atualmente, o projeto do processo não é o que ocorre na realidade. Assim, criou-se um conceito chamado de Process Mining. Com o Process Mining, a partir do processo já funcionando, é mapeado o fluxo que realmente está acontecendo no serviço. Essa técnica utiliza algoritmos que transformam logs em redes de Petri. Um dos algoritmos mais utilizados é o algoritmo Alpha. A proposta deste estudo é sanar, com a implementação da adaptatividade, duas desvantagens apresentada pelo algoritmo alpha: (1) tratamento de ruídos, (2) criação da rede de Petri em tempo real.

Keywords— process mining, adaptatividade, redes de petri, algoritmo alpha, log de eventos.

INTRODUÇÃO

Todos querem produzir mais em menos tempo. Uma maneira de realizar isso é através da boa definição de um processo de negócio que explicita as tarefas dependentes entre si e as tarefas que poderiam ser executadas em paralelo. Em uma situação ideal, o fluxo de tarefas é desenhado antes de se iniciar as atividades e é seguido. Na prática, gasta-se muito tempo desenhando o processo e muitas vezes ele não é seguido [1].

Uma maneira de simplificar o mapeamento de processos é utilizando uma técnica chamada Process Mining [1]. Process Mining são técnicas que permitem extrair informações através dos logs de eventos. Com essa metodologia, é possível checar, reparar, construir e fazer previsões de processos [2].

Um dos algoritmos básicos utilizados para transformar logs em processo é o algoritmo alpha. A saída deste algoritmo é uma rede de Petri. Uma das desvantagens do algoritmo alpha é que ele não trata ruído [3]. Outra desvantagem é que o algoritmo alpha deve ler toda a base de logs para depois gerar a rede de Petri, isso faz perder o dinamismo da descoberta de algumas tarefas que são gargalos do processo.

Com o objetivo de sanar essas duas desvantagens do algoritmo alpha, este artigo propõe a implementação de adaptatividade no algoritmo.

A área de estudo de adaptatividade permite que a rede de Petri se modifiquem dinamicamente, sem influências externas [4] [5]. Com essa característica, a adaptatividade irá permitir que o algoritmo alpha seja construído dinamicamente, conforme logs vão sendo criados, e irá contar a frequência que cada fluxo ocorre para retirar ruídos.

PROCESS MINING

Process Mining cria fluxos, redes de Petri, através da leitura de log de eventos. A figura 1 exemplifica essa área de estudo:

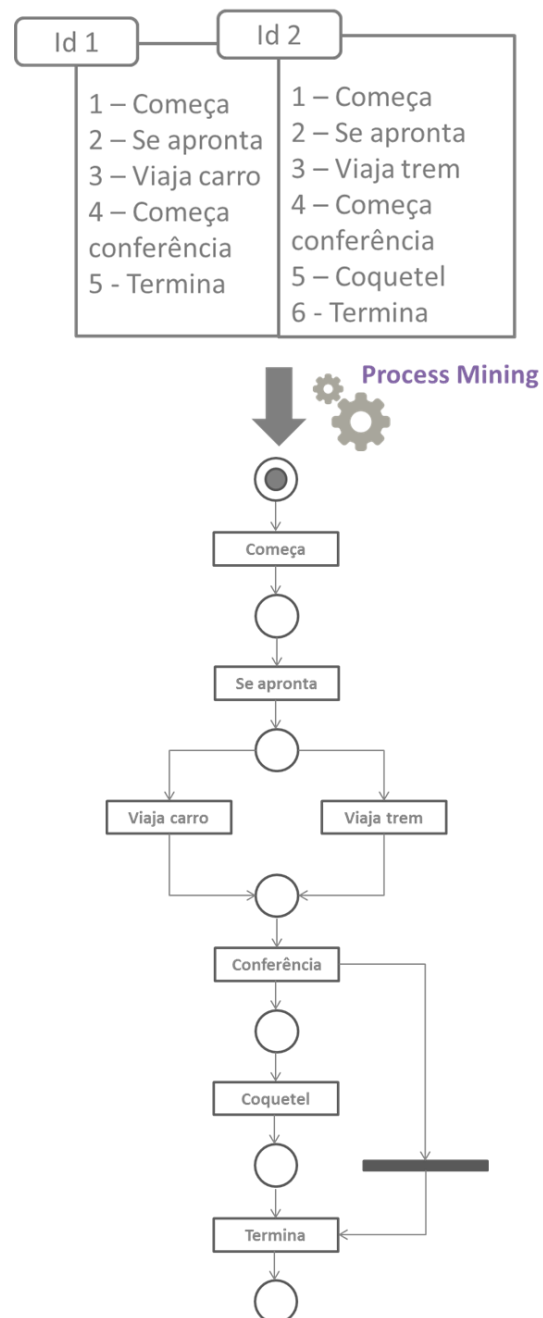


Figura 1: Ilustração de Process Mining.

ALGORITMO ALPHA

O algoritmo alpha é um dos algoritmos mais usados no Process Mining para criar a rede de Petri [3].

O algoritmo alpha é descrito a seguir. Passo 1: todo o log do processo é lido. Passo 2: todas as possíveis seqüências de eventos são mapeadas. Passo 3: analisa-se a relação entre os eventos. São elas: (a) sucessão direta $x > y$ – se após o acontecimento do evento x , acontece o y ; (b) casualidade $x \rightarrow y$ – se sempre o evento x acontece antes de y e $x > y$; (c) paralelo $x \parallel y$ – se o evento x acontece antes do y e o y acontece antes do x ($x > y$ e $y > x$); (d) escolha $x \# y$ – se o após o evento x não ocorre o y e após o y não ocorre o x (não $x > y$ e não $y > x$). Passo 4: cria-se a rede de Petri através de essas relações, para cada relação há uma configuração diferente da rede de Petri. A figura 2 [6] apresenta qual a configuração da rede de Petri a partir da relação existente entre os eventos.

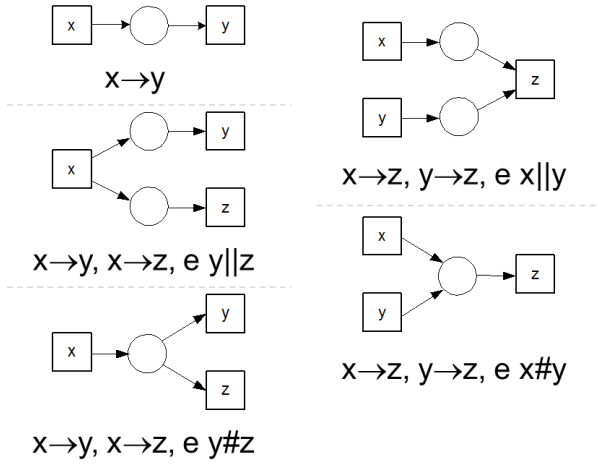


Figura 2: relações e suas respectivas redes de Petri. Apresentada por AALST e HEE [6].

A figura 3 apresenta um exemplo do algoritmo Alpha. A figura 4 mostra a codificação do algoritmo. Elas foram baseadas na explicação de AALST e HEE [6].

1 - Seqüência de logs

Caso 1: Tarefa A
Caso 2: Tarefa A
Caso 3: Tarefa A
Caso 3: Tarefa B
Caso 1: Tarefa B
Caso 1: Tarefa C
Caso 2: Tarefa C
...

2 - Todas as seqüências possíveis são mapeadas

ABCD
ACBD
EF

3 - Analisa-se a relação dos eventos

$A > B$ $A > C$ $B > C$
 $B > D$ $C > B$ $C > D$
 $E > F$ $B \parallel C$ $C \parallel B$
 $A \rightarrow B$ $A \rightarrow C$ $B \rightarrow D$
 $C \rightarrow D$ $E \rightarrow F$

4 - Rede de Petri a partir da relação de cada evento

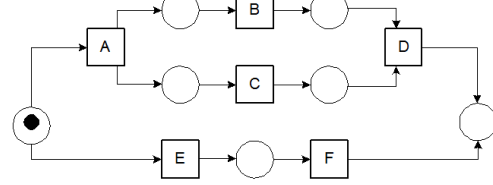


Figura 3: Exemplo do algoritmo alpha apresentado por AALST e HEE [6].

Let W be a workflow log over T . $\alpha(W)$ is defined as follows.

- $T_W = \{ t \in T \mid \exists \sigma \in W, t \in \sigma \}$,
- $T_1 = \{ t \in T \mid \exists \sigma \in W, t = \text{first}(\sigma) \}$,
- $T_0 = \{ t \in T \mid \exists \sigma \in W, t = \text{last}(\sigma) \}$,
- $X_W = \{ (A, B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall a \in A, \forall b \in B, a \rightarrow_W b \wedge \forall a_1, a_2 \in A, a_1 \#_W a_2 \wedge \forall b_1, b_2 \in B, b_1 \#_W b_2 \}$,
- $Y_W = \{ (A, B) \in X \mid \forall (A', B') \in X, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B') \}$,
- $P_W = \{ p_{(A, B)} \mid (A, B) \in Y_W \} \cup \{ i_W, o_W \}$,
- $F_W = \{ (a, p_{(A, B)}) \mid (A, B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A, B)}, b) \mid (A, B) \in Y_W \wedge b \in B \} \cup \{ (i_W, t) \mid t \in T_1 \} \cup \{ (t, o_W) \mid t \in T_0 \}$, and
- $\alpha(W) = (P_W, T_W, F_W)$.

Figura 4: codificação do algoritmo Alpha apresentado por AALST e HEE [6].

ADAPTATIVIDADE

Um dispositivo adaptativo se modifica dinamicamente, sem influências externas. Ele é influenciado somente através das entradas do sistema. Ele é formado por uma camada não adaptativa e uma adaptativa, composta de conjuntos de ações de Consulta, Adição e Eliminação [4].

Existe um modelo de Rede de Petri Adaptativa desenvolvido por GOMES e CAMOLESI [5]. Este modelo visa

à consulta das regras de entrada em uma transição e a adição ou eliminação de transições e posições da Rede de Petri. Quais regras serão definidas é um dos pontos a ser desenvolvido.

PROPOSTA

Este estudo tem o objetivo de (1) retirar os ruídos da análise de logs e (2) criar a rede de Petri através dos logs dinamicamente, sem necessitar ler toda a base existente antes de iniciar o algoritmo. A adoção de adaptatividade para cumprir o objetivo possibilita um maior dinamismo para o serviço.

Para se retirar o ruído utiliza-se um autômato adaptativo. O autômato adaptativo atua como um coletor de palavras na rede de Petri e faz a contagem de quantas vezes aquele caminho do fluxo é percorrido. Para se identificar um caminho fora do padrão utiliza-se o conceito apresentado por HAWKINS [7]. Neste conceito, caminhos que aparecem na frequência do interquartil $\times 1,5$ estão dentro da normalidade.

Para se fazer as podas da rede de Petri, após analisar que aquele caminho apresenta uma frequência baixa, então possivelmente é um fora do padrão, utiliza-se do conceito de tabela de decisão adaptativa. Se o caminho possui uma frequência alta, nenhum estado é alterado, se o caminho possui uma frequência baixa (operação de consulta), os estados envolvidos são retirados (operação de subtração) e nenhum estado é adicionado (operação de soma).

Para se fazer a rede de Petri dinamicamente, conforme logs vão sendo criados a rede de Petri vai se modificando, utiliza-se o conceito apresentado por PISTORI [8]. PISTORI [8] apresentou uma técnica de árvore de decisão adaptativa. Com a adaptatividade, foi possível elaborar um algoritmo incremental para gerar a árvore de decisão. Assim, durante o processo de classificação, foi permitida a incorporação de novos exemplos. Além disso, foi possível reordenar a árvore periodicamente.

RESULTADOS

O estudo proposto será válido se, após a aplicação de adaptatividade, o algoritmo alpha deixar de considerar ruídos em seus dados e se ele construir a rede de Petri conforme logs vão sendo criados, em tempo real. O algoritmo adaptativo não deve alterar significativamente o tempo de processamento do algoritmo alpha original. Essa análise é uma proposta para um próximo estudo.

REFERÊNCIAS

- [1] MEDEIROS, A.K.A.; SCHOENSLEBEN, Paul. **Workflow Mining: Current Status and Future Directions**. CoopIS/DOA/ODBASE 2003, LNCS 2888, pp. 389–406, 2003.
- [2] AALST, Wil van Der. **Process Mining: Discovery, Conformance and Enhancement of Business Processes**. Berlin: Springer-verlag Berlin Heidelberg, 2011.
- [3] BANERJEE, Anoopam; GRUPTA, Preeti. **Extension to Alpha Algorithm for Process Mining**. International Journal of Engineering & Computer Science, 2015.
- [4] NETO, João José. **Adaptive Rule-Driven Devices - General Formulation and Case Study**. Proc. 2001 Lecture Notes in Computer Science. Watson, B.W. and Wood, D. (Eds.): Implementation and Application of Automata 6th International Conf., Springer-Verlag, Vol.2494, pp. 234-250.
- [5] GOMES, Wilson C. M.; CAMOLESI, Almir R.. **Visualizador gráfico para Redes de Petri Adaptativa**. WTA - Workshop de Tecnologia Adaptativa. São Paulo, out. 2008.
- [6] AALST, Wil van Der.; HEE, Van K.M.. **Workflow Management: Models, Methods, and Systems**. MIT press, Cambridge, MA, 2002.
- [7] HAWKINS, D. **Identification of Outliers**. Chapman and Hall, London, 1980.
- [8] PISTORI, Hemerson. **Tecnologia Adaptativa em Engenharia de Computação: Estado da Arte e Aplicações**. Tese de Doutorado, EPUSP, São Paulo, 2003.

Sheila Genesine Dada, graduada em engenharia elétrica com ênfase em computação pela Escola Politécnica da Universidade de São Paulo (2014). Atualmente é mestranda do Programa de Pós-Graduação em engenharia de computação da Escola Politécnica da Universidade de São Paulo, atuando como aluna pesquisadora no LTA – Laboratório de Linguagens e Técnicas Adaptativas – do PCS. Curriculum Lattes: <http://lattes.cnpq.br/1162729917105344>

João José Neto é graduado em Engenharia de Eletricidade (1971), mestre em Engenharia Elétrica (1975), doutor em Engenharia Elétrica (1980) e livre-docente (1993) pela Escola Politécnica da Universidade de São Paulo. Atualmente, é professor associado da Escola Politécnica da Universidade de São Paulo e coordena o LTA – Laboratório de Linguagens e Técnicas Adaptativas do PCS – Departamento de Engenharia de Computação e Sistemas Digitais da EPUSP. Tem experiência na área de Ciência da Computação, com ênfase nos Fundamentos da Engenharia da Computação, atuando principalmente nos seguintes temas: dispositivos adaptativos, tecnologia adaptativa, autômatos adaptativos, e em suas aplicações à Engenharia de Computação, particularmente em sistemas de tomada de decisão adaptativa, análise e processamento de linguagens naturais, construção de compiladores, robótica, ensino assistido por computador, modelagem de sistemas inteligentes, processos de aprendizagem automática e inferências baseadas em tecnologia adaptativa. Curriculum Lattes: <http://lattes.cnpq.br/4091709928353457>.