



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
ESCOLA DE INFORMÁTICA APLICADA

APLICAÇÃO DA VERIFICAÇÃO DE CONFORMIDADE DE PROCESSOS NA  
AUDITORIA DE PROCESSOS

Thaís Mester Barbosa

**Orientadora**  
Flávia Santoro

RIO DE JANEIRO, RJ – BRASIL  
JULHO DE 2018

Catálogo informatizada pelo(a) autor(a)

B238 Barbosa, Thaís Mester  
APLICAÇÃO DA VERIFICAÇÃO DE CONFORMIDADE DE  
PROCESSOS NA AUDITORIA DE PROCESSOS / Thaís Mester  
Barbosa. -- Rio de Janeiro, 2018.  
67 f

Orientadora: Flávia Santoro.  
Trabalho de Conclusão de Curso (Graduação) -  
Universidade Federal do Estado do Rio de Janeiro,  
Graduação em Sistemas de Informação, 2018.

1. Mineração de processos. 2. Auditoria de  
processos. 3. Verificação de conformidade. 4. ProM .  
5. Disco. I. Santoro, Flávia, orient. II. Título.



APLICAÇÃO DA VERIFICAÇÃO DE CONFORMIDADE DE PROCESSOS NA  
AUDITORIA DE PROCESSOS

Thaís Mester Barbosa

Projeto de Graduação apresentado à Escola de  
Informática Aplicada da Universidade Federal do  
Estado do Rio de Janeiro (UNIRIO) para obtenção do  
título de Bacharel em Sistemas de Informação.

Aprovado por:

---

Flávia Santoro (UNIRIO)

---

Kate Revoredo (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

JULHO DE 2018

## **Agradecimentos**

Agradeço à minha orientadora Flávia Santoro por acreditar em mim e ter sido sempre muito solícita e paciente em me orientar até mesmo nos finais de semana. Agradeço aos professores que tive o prazer de ser aluna durante meus anos de faculdade na UNIRIO, pois sem a motivação e empenho deles, eu não seria a profissional que sou hoje no mercado da tecnologia da informação.

Em especial, agradeço aos meus amigos do trabalho que estiveram ao meu lado nesta caminhada, incentivando-me nos momentos desanimadores, e fazendo-me crescer como pessoa e como profissional. Agradeço aos meus pais por terem provido os meios para que minha educação fosse a melhor possível.

E por fim, agradeço a Deus, pois sempre esteve ao meu favor dando-me forças para alcançar meus objetivos na vida. Desta vez não foi diferente.

## RESUMO

Com o aumento da preocupação em evitar fraudes e quaisquer atos não conformes com a lei SOX e as políticas internas das companhias, a atuação da auditoria de processos vem crescendo a cada ano. Com isso a carga de trabalho dos auditores e a exigência de análises precisas em um curto espaço de tempo tendem somente a aumentar.

Em contrapartida, o avanço tecnológico das ferramentas utilizadas pelos auditores de processos não acompanhou tal crescimento. O presente trabalho visa mostrar como a mineração de processos pode auxiliar auditores a realizarem análises de conformidade de processos de uma forma simples, precisa e ágil. Com o auxílio das ferramentas de mineração de processo Disco e ProM, será mostrado como manipular, por meio de visões gráficas, os logs de eventos provenientes de ERPs, a fim de descobrir modelos de processos e compará-los com o log da real execução destes processos. Para que assim possa ser gerada uma rede Petri com diversas informações sobre como um processo está sendo executado com relação ao seu modelo, tais como: fitness entre o log e o modelo, quantificação e evidência dos caminhos alternativos ao modelo, pontos de sobrecarga de recursos e análise do desempenho individual das atividades modeladas.

**Palavras-chave:** Mineração de processos, auditoria de processos, verificação de conformidade, ProM e Disco.

## ABSTRACT

Regarding the concernment around avoiding fraud and any acts that are not in conformance with SOX and internal company's policies, the auditing process performance are growing every year. As a result, the workload of the auditors and the requirement for accurate analysis in a short period tend only to increase. On the other Hand, the technological advancement of the tools used by process auditors did not followed such growth. This paper aims to show how process mining can assist auditors to perform conformance checking analysis in a simple, precise, and agile way. With the process mining tools Disco and ProM support, it will be shown how to manipulate, through graphics visions, event logs from ERP's in order to discovery process models and compare them with the log of the actual execution of these processes. In order to be able to generate a Petri net with several information about how a process execution can be related to its model, such as fitness between log and model, quantification, and evidence of alternative paths to the model, resource overload regions and performance individual analysis of the modeled activities.

**Keywords:** Process mining, process auditing, conformance checking, ProM and Disco.

## Índice

1	Introdução	11
1.1	Motivação	13
1.2	Objetivos	15
1.3	Organização do texto	15
2	Auditoria Interna de Processos	17
2.1	Domínios de atuação da auditoria	17
2.2	Controles internos	18
2.3	Técnicas de auditoria interna	18
2.4	Problemas em testes de auditoria	19
2.5	Modelagem e Descoberta de Processos	22
2.6	Mineração de Processos	23
2.7	DISCO	28
2.8	ProM	28
3	A Mineração de Processos na Auditoria de Processos	29
3.1	Mecânica da auditoria antes da mineração de processos	29
3.2	Valor agregado da mineração de processos na auditoria	30
3.3	A verificação de conformidade em mineração de processos	31
3.4	Replay	31
3.5	Trabalhos relacionados	32
3.1	Rede de Petri	34
4	Cenário de Aplicação	36
4.1	Descobrir as atividades do log	37
4.2	Analisar a dinâmica entre atividades	41
4.3	Identificar campos definidores do log	43
4.4	Idealização do log modelo e segregação do log original	44
4.5	Aplicar plug-in de verificação de conformidade	46

4.6	Informações fornecidas pelo plug-in de conformidade	55
4.6.1	Cores das atividades	55
4.6.2	Barra verde	56
4.6.3	Barra lilás	56
4.6.4	Atividades com borda vermelha	56
4.6.5	Círculos brancos e amarelos	57
4.6.6	Tamanho dos círculos	58
4.6.7	Retângulos pretos	58
4.7	Comparação dos resultados	59
	• Comparação com alterações no campo “ <i>Move on Model Cost</i> ”	59
	• Comparação com alterações no campo “ <i>Move on Log Cost</i> ”	61
5	Conclusão	64

## Índice de Tabela

Tabela 1 – Frequência relativa dos sub processos.....	40
---	----

## Índice de Figuras

Figura 1 – Mineração de Processos.....	23
Figura 2 – Exemplo de log de eventos.....	24
Figura 3 – Formato XES (parte 1).....	26
Figura 4 – Formato XES (parte 2) .....	27
Figura 5 – Procedimento de confecção da simulação, (ACCORSI e STOCKER, 2012).....	33
Figura 6 – Exemplo de uma rede Petri.....	35
Figura 7 – Modelo com todos os caminhos e atividades do processo.....	37
Figura 8 – Modelo intermediário com o máximo de atividades e mínimo de caminhos.....	38
Figura 9 – Modelo simplificado com mínimo de atividades e caminhos.....	40
Figura 10 – Visão tabular do log.....	43
Figura 11 – Exclusão da coluna “Variant”.....	44
Figura 12 – Modelo de referência.....	45
Figura 13 – Filtro “Variation” considerando 20% dos casos do log modelo.....	46
Figura 14 – Input e Output do plug-in de verificação de conformidade.....	47
Figura 15 – Input e Output do plug-in gerador da rede de Petri.....	48
Figura 16 – Configurações de produção da rede de Petri.....	48
Figura 17 – Rede de Petri do processo de pagamento de contas.....	49
Figura 18 – Configurações iniciais do plug-in de verificação de conformidade.....	50
Figura 19 – Caixa de diálogo informando atividades sem mapeamento com modelo....	51
Figura 20 – Configurações de finalidade das análises do plug-in.....	52
Figura 21 – Configurações de custo de movimentos.....	52
Figura 22 – Output do plug-in “Replay a Log on Petri Net for Conformance Analysis”.....	54

Figura 23 – Atividade “finish payment”, output do plug-in de replay de conformidade.....	55
Figura 24 – Atividade “begin editing”, output do plug-in de replay de conformidade.....	55
Figura 25– Ilustração do possível trajeto após atividade “decide”.....	56
Figura 26 – Caminhos executados pelo log teste logo após execução da atividade “decide”.....	57
Figura 27 – Caminhos executados pelo log teste logo após execução da atividade “begin editing”.....	57
Figura 28 – Caminhos executados pelo log teste após execução da atividade “begin editing”.....	58
Figura 29 – Cada decisão de trajeto é representada por um retângulo preto.....	58
Figura 30 – Parametrização do campo “Move on Model Cost”.....	60
Figura 31 – Comparativo das estatísticas da execução de referência, ao lado esquerdo, com a execução com custo em “Move on Model Cost”, ao lado direito.....	60
Figura 32 – Parametrização do campo “Move on Log Cost”.....	62
Figura 33 – Comparativo das estatísticas da execução de referência, ao lado esquerdo, com a execução com custo em “Move on Log Cost”, ao lado direito.....	62

# 1 Introdução

## 1.1 Motivação

Nas últimas décadas, os escândalos de fraudes empresariais concentraram a atenção das mídias. Principalmente após o emblemático caso da grande empresa estadunidense Enron, a qual após escândalos relacionados à área contábil veio a fechar suas portas. O ocorrido teve como efeito a criação da lei *Sarbanes-Oxley*, abreviada como lei SOX<sup>1</sup>. Esta lei foi elaborada nos Estados Unidos afim de garantir a conformidade dentro das empresas e para evitar a fuga dos investidores causada pela insegurança e perda de confiança em relação às escrituras contábeis e aos princípios de governança da empresa. Para isso, ela obriga às empresas a reestruturarem seus processos para aumentar os controles, a segurança e a transparência na condução dos negócios, na administração financeira, nas escriturações contábeis e na gestão e divulgação das informações. Para garantir estes pontos, a Lei SOX exige que as empresas adotem mecanismos de auditoria que possam auxiliar à alta administração a avaliar e acompanhar a adequação, eficácia e eficiência dos processos dentro de uma empresa. Com esta exigência, o profissional da área de auditoria de processos veio ganhando destaque entre as grandes empresas devido à sua experiência em atuar neste suporte ao corpo executivo.

O processo, mencionado anteriormente, pode ser entendido como: uma sequência de atividades ordenadas em um modo lógico que possui como objetivo produzir um serviço ou produto que tenha valor para um determinado grupo de clientes<sup>2</sup>. Dentre vários processos envolvidos na produção de produtos ou serviços, existem os processos de negócio. Segundo GONÇALVES (2000, p.12), processos de negócio são aqueles que caracterizam a atuação da empresa e que são suportados por outros processos internos, resultando no produto ou serviço que é recebido por um cliente externo.

Para GONÇALVES (2000),

[...] a tecnologia tem um papel fundamental no gerenciamento dos processos empresariais. Ela influencia tanto a forma de realizar o trabalho como a maneira de gerenciá-lo. Muitas vezes, o processo

---

<sup>1</sup> <https://portaldeauditoria.com.br/introducao-lei-sarbanes-oxley-sox/> último acesso 25/04/2018

<sup>2</sup> [https://www.infoescola.com/administracao/\\_processo-de-negocio/](https://www.infoescola.com/administracao/_processo-de-negocio/), último acesso: 21/04/2018

obedece a uma sequência estrita de atividades, ditada pela sua tecnologia característica ou pela própria lógica do trabalho. (p. 17)

Com isso, é possível concluir que os avanços tecnológicos detêm forte influência sobre processos operacionais. Tais avanços, segundo SCHERR (2006, p. 22), contemplam todas as atividades que estão ligadas diretamente com o desenvolvimento da atividade fim da empresa: gestão do relacionamento com o cliente (Customer Relationship Management – CRM), logística, desenvolvimento de produto, planejamento e controle da produção (PCP), gestão de material, entre outros.

Uma questão relevante na gestão de processos operacionais é a garantia de que os processos estejam alinhados aos objetivos estratégicos da organização e que sigam padrões e políticas previamente definidos. Em paralelo, um dos principais objetivos da auditoria de processos no ambiente corporativo é exatamente avaliar o nível desta conformidade de seus processos operacionais e seus respectivos controles. Esta atividade tem como finalidade corrigir falhas e/ou erros na cadeia de processos trazendo qualidade para tais processos, e conseqüentemente valor para a empresa que está sendo auditada (DIAS, 2015).

Com base em CUNHA e HEINS (2005), para atingir este objetivo um auditor interno necessita analisar os processos operacionais atualmente ativos em sua completude. Sendo assim, dependendo do porte da empresa, é preciso coletar um volume grande de informações sobre cada processo operacional em exercício. De acordo com CUNHA e HEIN (2005) este tipo de análise é exercida com base em uma técnica chamada teste substantivo. Esta técnica consiste em analisar, atualmente de forma manual, cem por cento dos logs de transações dos processos operacionais afim de comprovar a conformidade destes com relação às normas internas e de mercado. Por este motivo, o custo de tempo deste tipo de análise torna-se muito dispendioso. Além desta técnica, a auditoria costuma efetua a maior parte de suas análises por meio de técnicas de amostragem. Contudo, por estas técnicas não levarem em consideração a população inteira de dados do processo em análise, podem ocorrer casos em que atividades não conformes passem despercebidas e não sejam tratadas.

Por outro lado, técnicas mais recentes, como as de mineração de processos oferecem soluções automatizadas para a análise de conformidade de processos. Segundo AALST e VERBEEK (2006) a mineração de processos de negócios é formada por um conjunto de técnicas que visam analisar logs de eventos de sistemas de informações afim

de descobrir estruturas organizacionais, sociais, estruturas de processos, de controles e de dados de uma empresa. Tendo isto em vista, o presente trabalho visa apresentar e comparar técnicas de mineração de processos para melhorar o tempo de conclusão das análises dos processos e controles em questão. Para que assim, um auditor não necessite realizar excessivas horas de trabalho para atender aos curtos prazos impostos pelo cliente. Além de trazer à alta administração maior valor e confiabilidade no serviço prestado pela auditoria.

## **1.2 Objetivos**

O presente projeto tem como objetivo principal mostrar como técnicas de mineração de processos podem otimizar e trazer agilidade para a fase de descoberta de fluxos processuais e para a fase de verificação de conformidade do modelo de processo de uma empresa com o processo efetivamente realizado por seus funcionários. Com esta verificação, será possível detectar algum erro e/ou falha em alguma atividade ou controle do processo em execução.

Para tal, os seguintes passos foram realizados:

- Estudo dos conceitos importantes para o desenvolvimento do trabalho;
- Levantamento de logs da área de pagamentos de uma organização;
- Mapeamento de todas as atividades deste processo com a ferramenta DISCO;
- Análise da dinâmica de interação destas atividades;
- Descoberta através da mineração de processos de um modelo de referência para a comparação dos logs;
- Aplicação de algoritmo de checagem de conformidade com a ferramenta PROM entre o log do processo padrão e o log de teste;
- Análise das conclusões obtidas com o resultado da aplicação do algoritmo.

## **1.3 Organização do texto**

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo II: Auditoria Interna Focada em Processos.
- Capítulo III: Mineração de Processos na Auditoria Interna de Processos.
- Capítulo IV: Aplicação da Teoria.
- Capítulo V: Conclusões – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.

## 2 Auditoria Interna de Processos

Auditoria interna é uma atividade de avaliação independente e de assessoramento da administração, voltada para o exame e avaliação da adequação, eficiência e eficácia dos sistemas de controle, bem como da qualidade do desempenho das áreas, em relação às atribuições e aos planos, às metas, aos objetivos e às políticas definidos para as mesmas (FRANCO e MARRA, 2000, p.217). Na auditoria interna de processos o princípio é o mesmo, porém com foco voltado aos processos da empresa. A seguir serão apresentados os conceitos que serão abordados nesse trabalho a fim de fornecer uma contextualização para mostrar o trabalho que foi desenvolvido em torno do tema em questão.

### 2.1 Domínios de atuação da auditoria

Segundo NETO e SOLONCA (2007) a auditoria interna é dividida em seis domínios dentro de uma empresa. Sendo eles: auditoria de programas de governo, administrativa, contábil, financeira, operacional e a auditoria da tecnologia da informação.

A auditoria de programas do governo é responsável pelo acompanhamento, exame e avaliação da execução de programas e projetos governamentais. A auditoria administrativa engloba o plano da organização, seus procedimentos, diretrizes e documentos de suporte à tomada de decisão. A auditoria contábil é relativa à fidedignidade das contas da instituição. Esta tem como finalidade fornecer a garantia de que as operações e o acesso aos ativos se efetuem de acordo com as devidas autorizações. A auditoria financeira, ou auditoria das contas, analisa as contas, a situação financeira, a legalidade e regularidade das operações e aspectos contábeis, financeiros, orçamentários e patrimoniais. Verificando assim, se todas as operações foram corretamente autorizadas liquidadas, ordenadas, pagas e registradas. A auditoria operacional consiste em analisar todos os níveis de gestão nas fases de programação, execução e supervisão, sob a ótica da economia, eficiência e eficácia. Além disso, a auditoria operacional também analisa a execução das decisões em vigor e verifica até que ponto os resultados pretendidos foram

alcançados. Já a auditoria da tecnologia da informação, que é essencialmente operacional, analisa especificamente os controles internos, o ambiente, os sistemas e a segurança da informação da entidade fiscalizada, identificando assim, seus pontos fortes e deficiências.

O enfoque deste trabalho será sobre a auditoria operacional e a auditoria da tecnologia da informação. Estas auditorias dedicam-se a realizar suas análises sob a ótica dos processos de negócio da empresa auditada.

## **2.2 Controles internos**

Segundo ATTIE (2009, p. 152 e 153), os controles internos são divididos em duas vertentes, os controles contábeis e os administrativos. Os controles contábeis compreendem o plano de organização e todos os métodos e procedimentos relacionados, principalmente com a salvaguarda do patrimônio e a fidedignidade dos registros contábeis. Geralmente incluem os seguintes controles: sistemas de autorização e aprovação; separação das funções de escrituração e elaboração dos relatórios contábeis daquelas ligadas às operações ou custódia dos valores; e controles físicos sobre estes valores. Já os controles administrativos compreendem o plano de organização e todos os métodos e procedimentos que dizem respeito à eficiência operacional e a decisão política traçada pela administração. Normalmente, se relacionam de forma indireta aos registros de forma indireta aos registros financeiros. Com frequência abrangem análises estatísticas, estudos de tempo e movimentos, relatórios de desempenho e controles de qualidade.

Tendo isto em vista, iremos nos concentrar nos controles internos administrativos, os quais são o foco da auditoria interna de processos devido principalmente à grande preocupação com o âmbito operacional das empresas.

## **2.3 Técnicas de auditoria interna**

Com o crescimento da utilização de sistemas informacionais, cada vez mais dados são armazenados em bases de dados. Na ótica de processos de negócios, compreendemos este cenário como o armazenamento dos *logs* de transações provenientes da interação

entre atividades de um processo da empresa. Para compreender e analisar a dinâmica desses logs de processos, GOMES (2000, p. 114) comenta que os auditores internos utilizam ferramentas informatizadas de investigação como o ACL<sup>3</sup>, IDEA<sup>4</sup> e o SAS<sup>5</sup> independente do porte da empresa auditada.

Um dos softwares mais comumente utilizados nesta tarefa no Brasil é o ACL (Audit Command Language) ou Linguagem de Auditoria por Comandos. Criado por canadenses, o ACL é um software específico para fins de auditoria, pois permite a leitura e análise de uma grande massa de dados, que podem estar em diferentes formatos ou arquivos, sem que isso afete a velocidade de processamento. O IDEA (Interactive Data Extraction and Analysis) concorrente do ACL, também é um software voltado para extração e análise de dados muito utilizado no mercado. Já o SAS além de extrair e analisar dados, também é um software de trilha de auditoria, ou seja, permite ao auditor executar o acompanhamento de rotinas executadas em sistemas por funcionários ou usuários externos à empresa.

Ainda de acordo com GOMES (2000), todos estes softwares supracitados exigem do auditor um alto grau de conhecimento a respeito dos conceitos envolvendo banco de dados, trilha de auditoria, correlação de informações e outros aspectos que serão considerados vitais no cumprimento dos trabalhos de auditoria.

## **2.4 Problemas em testes de auditoria**

Segundo OLIVEIRA (1989), os testes em auditoria constituem o processo fundamental pelo qual o auditor reúne elementos comprobatórios. Estes testes podem ser aplicados em todas as transações ou, como mais frequentemente utilizado, em uma amostra representativa adequada. Neste procedimento o auditor consegue examinar uma quantidade menor que o total de um conjunto de dados, a fim de emitir uma conclusão sobre o mesmo.

Sendo a amostragem a técnica mais utilizada na auditoria, OLIVEIRA (1989) define o escopo do problema enfrentado pelo auditor na etapa:

---

<sup>3</sup> <https://techsupply.com.br/> último acesso: 20/05/2018

<sup>4</sup> <http://www.audtrack.com/> último acesso: 20/05/2018

<sup>5</sup> [https://support.sas.com/rnd/emi/APM\\_main/index.html](https://support.sas.com/rnd/emi/APM_main/index.html) último acesso: 20/05/2018

[...]o problema fundamental do auditor é a maneira de determinar a natureza e a extensão da verificação necessária: até onde se deve chegar, quanto é preciso investigar, quais são as variáveis a considerar para se ter a certeza de que as condições testadas se referem também às que não o foram e, com isso, formar uma opinião definitiva [...]. (p.18)

De acordo com OLIVEIRA (1989), os testes são classificados em “testes de observância” e “testes substantivos”.

a) Testes de observância

Os testes de observância são procedimentos empregados pelo auditor, afim de determinar se certos procedimentos de controle interno estabelecidos pelo sistema da empresa são cumpridos corretamente. Visa aprovar a credibilidade de tais controles e não o registro correto das transações. Como por exemplo, o auditor pode verificar se o encarregado de contas a pagar aprovou cada fatura para codificação de contas. Este teste relativo à aprovação constitui um teste de observância e não será levado em conta se a codificação das faturas é uma realidade concreta. O teste ideal para a correção da codificação seria um teste substantivo.

b) Testes substantivos

Os testes substantivos são aqueles empregados pelo auditor com a finalidade de obter provas suficientes e convincentes sobre as transações, saldos e divulgações nas demonstrações contábeis, que lhe proporcionem fundamentação razoável para a emissão do relatório.

Antes que os testes sejam iniciados, o auditor deve estabelecer um plano de amostragem e documentá-lo em papéis de trabalho (*working papers*), a partir deste, o auditor seguirá o processo de amostragem. De acordo com SCOTT (1982), este plano geralmente é desenvolvido em 13 etapas, sendo elas:

1. Definir os objetivos do teste;
2. Determinar os níveis de risco aceitáveis para o teste;
3. Especificar a população;
4. Calcular a amplitude da amostra;

5. Especificar a unidade da amostragem (dúzia, cento, etc.);
6. Determinar o método de amostragem que vai ser utilizado;
7. Selecionar a amostra;
8. Aplicar os procedimentos de auditoria à amostra;
9. Aplicar procedimentos alternativos onde os procedimentos de auditoria não podem ser aplicados;
10. Formar conclusão sobre cada item da amostra;
11. Avaliar os resultados da amostra (inferência);
12. Revisar o trabalho do teste;
13. Aplicar opinião sobre o total geral.

OLIVEIRA (1989) explica ainda que a amostragem é dividida em duas técnicas, a amostragem espontânea e a intencional. A amostragem espontânea visa analisar uma amostra sem tendenciosidades por parte do auditor. Esta não é uma técnica trivial, pois o auditor pode acabar escolhendo itens de acordo com sua fácil localização ou verificação. Por isso ele necessita atentar-se aos seguintes pontos: formar opinião somente das populações pertinentes às amostras tomadas; e deixar todo item da população ter uma igual ou conhecida chance de seleção ao: certificar-se de que nenhum modelo-padrão da população afetará a escolha da amostra feita ao acaso; e ao impedir que tendenciosidades pessoais afetem a seleção dos itens da amostra.

Em alguns casos não é possível escolher uma amostra imparcial ou a esmo, devido à natureza e a qualidade da população ou a amostra cujo tamanho tenha suporte estatístico pode não ser conveniente. Nestas circunstâncias, o auditor é forçado a utilizar a técnica de amostragem intencional, na qual faz-se uma seleção subjetiva de itens, ou examina-se um número restrito de itens. Embora a amostragem intencional seja uma técnica de auditoria reconhecida e aceitável, o auditor deve ter um cuidado especial ao emitir opinião projetada para a população a fim de aproximar-se do grau de confiança matemático da Amostragem Estatística. OLIVEIRA (1989) afirma que com a utilização desta técnica, torna-se muito difícil afirmar com certeza, que a amostra retirada de uma determinada população representa exatamente esta mesma população. Isto porque se trata de um processo de amostragem e não um censo completo (exame de todos os itens da população).

Tendo todos estes pontos em vista, pode ser concluído que se fossem empregadas técnicas de mineração de processos para auxiliar a etapa de testes em auditoria, teríamos

grandes vantagens. Visto que a complexidade das análises em cada teste seria minimizada, a precisão destas análises seria total e o tempo de resposta das mesmas seria reduzido.

## **2.5 Modelagem e Descoberta de Processos**

Segundo SANTOS (2014) a modelagem de processos de negócios consiste, de forma abstrata, na técnica de representar como o processo ocorre na prática, levando em consideração características intrínsecas, como os recursos, controles, papéis e responsabilidades. A modelagem de processos é interessante para a auditora interna de processos pois auxilia na fase de mapeamento de processos de uma empresa. Esses modelos podem ser categorizados como descritivos e prescritivos. O modelo descritivo tenta capturar os processos existentes sem ser normativo De acordo com LOPES (2015), um exemplo deste modelo seria um processo de um hospital cujo objetivo é reagir a uma situação urgente, contudo, a flexibilidade para agir de forma diferente do fluxo normal das ações é de extrema importância.

Já o modelo prescritivo, segundo LOPES:

[...] descreve a maneira que o processo pode ser executado. Em um Sistema de Workflow (WFM - WORKFLOW MODELS), modelos prescritivos são usados para reforçar um caminho particular de trabalho em sistemas de informação. Entretanto, analistas muitas vezes precisam desviar de seu fluxo normal em relação aos modelos mais rígidos dos sistemas WFM. Na maioria das situações, modelos prescritivos não são usados diretamente pelos sistemas de informação. (p. 10)

A fase do ciclo de processos correspondente à descoberta de processos é a coleta de informações sobre um processo existente e sua representação em um modelo. Na prática, a coleta de informações muitas vezes é demorada e complicada (DUMAS, et al., 2013), pois os processos de negócio são dinâmicos e complexos, exigindo técnicas mais sofisticadas do que simplesmente a modelagem.

As técnicas de mineração de processos podem ser utilizadas para extrair informações de logs de eventos de sistemas corporativos relacionados a processos, para fazer descoberta automática de modelos de processos (AALST, 2011). O log de eventos é o registro da execução do processo. O evento neste log refere-se especificamente a uma

atividade do processo de negócio, deve indicar a instância do processo (caso) e deve estar totalmente ordenado (ROZINAT e AALST, 2008). A principal razão pela qual a mineração de dados tem atraído muita atenção na indústria nos 11 últimos anos é a disponibilidade de grandes quantidades de dados e a necessidade de transformar esses dados em informações úteis e conhecimento (HAN e KAMBER, 2000).

## 2.6 Mineração de Processos

O objetivo da mineração de processos é descobrir, monitorar e melhorar processos reais, a fim de extrair informações e conhecimento de logs de eventos provenientes dos sistemas de informação (ERPs) de uma empresa, conforme é apresentado na Figura 1.

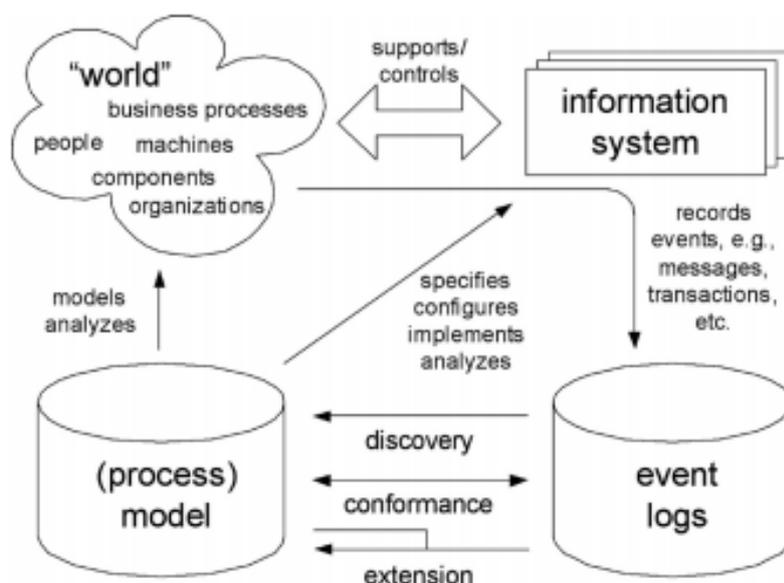
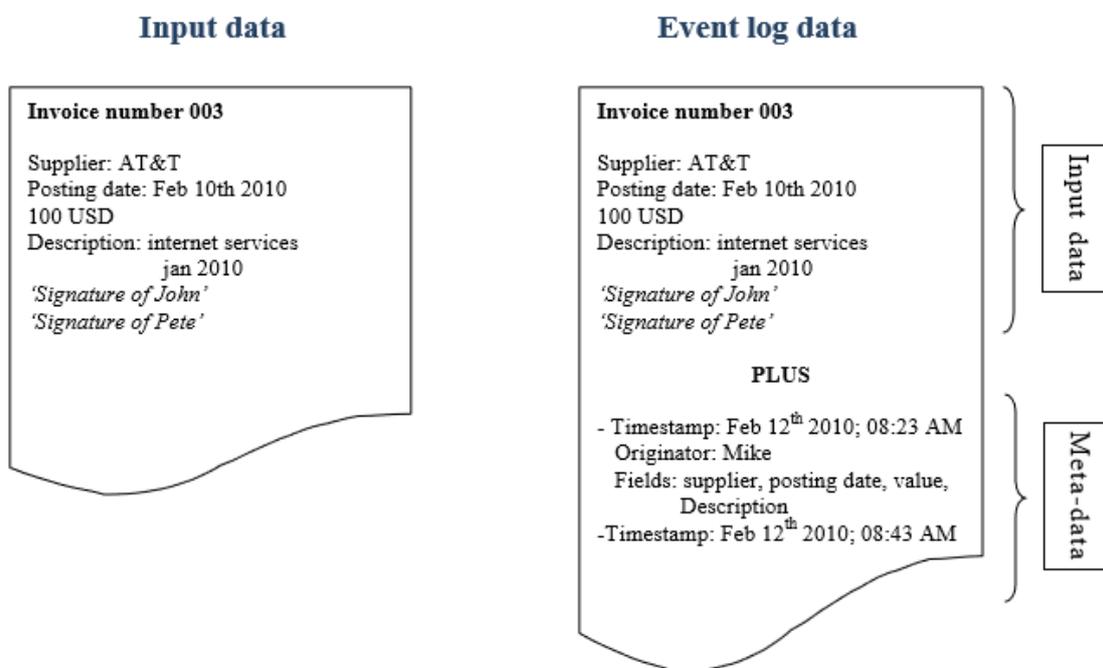


Figura 1 – Mineração de Processos

De acordo com a Figura 1, a mineração de processos nos permite: descobrir processos; extrair modelos de processo a partir de um log de eventos; verificar a conformidade entre um modelo de processo e sua execução no dia-a-dia; monitorar desvios de execução de um processo; descobrir automaticamente redes sociais ou organizacionais; estender um modelo; reparar modelos; realizar previsões de acontecimentos e recomendações com base no histórico de eventos (AALST, 2012). Como pode-se notar, todas as análises viabilizadas pela mineração de processos somente são possíveis através da

disponibilização e investigação de logs de eventos de um processo. O log de eventos compreende uma estrutura que contém os dados de uma atividade de um processo e seus respectivos dados históricos.



**Figura 2 – Exemplo de log de eventos**

A Figura 2 (JANS, ALLES e VASARHELYI, 2012) fornece um exemplo concreto dos dados que podem ser armazenados em um log de eventos sobre uma fatura. O lado esquerdo mostra os dados sobre a fatura inserida pela pessoa que faz a entrada de dados, que é o que seria por exemplo verificado no caso de uma auditoria. Em tempos passados, esses dados referem-se aos que foram introduzidos pelo contador da firma do cliente em seu livro-razão. Podemos chamá-los de "dados de entrada", pois são caracterizados pelo ato controlável de uma pessoa que insere os dados. Esses dados de entrada são o tipo de dados disponíveis no momento em que esses dados são armazenados, como o número da fatura, a data de lançamento e o fornecedor.

O lado direito da Figura 2 mostra os dados da mesma fatura armazenados no log de eventos. Formalmente, todos os dados de entrada também fazem parte do log de eventos, mas claramente são as entradas exclusivas do log de eventos que são de interesse particular para um auditor porque esses dados são registrados automaticamente pelo

sistema e não são inseridos pelo auditado. São essas características adicionais que fazem um log de eventos ter uma dimensão diferente de um conjunto de dados de entrada de um sistema. A instância, no exemplo da fatura, é identificada exclusivamente pelo número da fatura. Além do identificador de instância e dos dados de entrada, são adicionados todos os dados exclusivos de todas as atividades que foram realizadas nessa instância. Os logs de eventos podem ser gerados a partir de diferentes tipos de Sistemas de Informação, como por exemplo, Enterprise Resource Planning (ERP), Business Process Management Systems (BPMS), Customer Relationship Management (CRM) ou sistemas customizados.

Para que as ferramentas de mineração de processos possam mais facilmente trabalhar com estes logs, um padrão foi proposto. O padrão XES<sup>6</sup> define uma gramática para uma linguagem baseada em *tags*, conforme podemos ver nas Figuras 3 e 4. Seu objetivo é fornecer aos projetistas de sistemas de informação uma metodologia unificada e extensível para capturar comportamentos de sistemas por meio de logs de eventos e fluxos de eventos definidos no padrão XES. Além disso, este padrão inclui uma coleção básica dos chamados protótipos de extensão XES que fornecem semântica a certos atributos, conforme registrados no log ou fluxo de eventos.

---

<sup>6</sup> <http://www.xes-standard.org/start> último acesso: 24/06/2018



Figura 3 – Formato XES (parte 1)

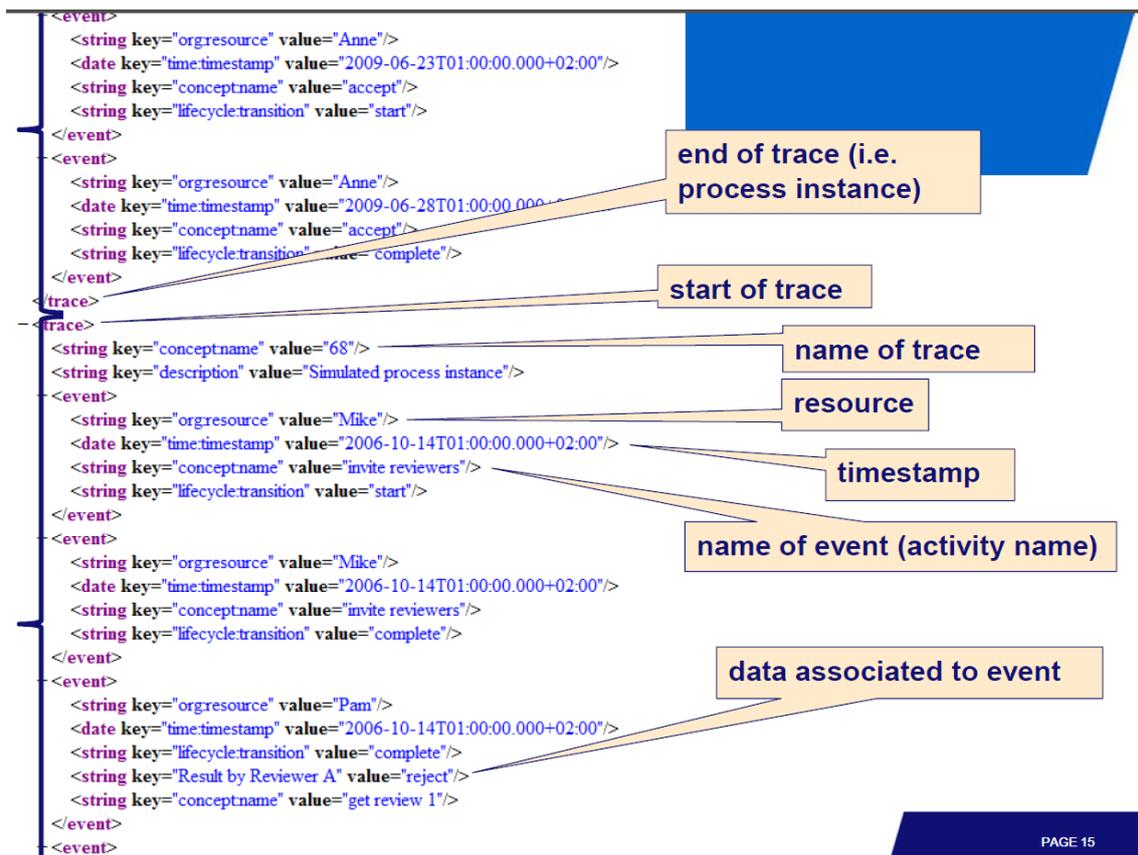


Figura 4 – Formato XES (parte 2)

O objetivo deste padrão é fornecer um formato do tipo XML (*eXtensible Markup Language*) geralmente reconhecido para o intercâmbio de dados de eventos entre sistemas de informação em muitos domínios de aplicativos, porém também permite o intercâmbio de informações entre ferramentas de análise de dados. Este padrão visa padronizar a sintaxe e a semântica dos dados do evento que, por exemplo, estão sendo transferidos de uma ferramenta que gera esses dados para uma outra que está analisando esses dados. Como resultado, se os dados do evento forem transferidos usando a sintaxe descrita por este padrão, sua semântica será bem entendida e clara nos dois programas. A seguir, serão descritas algumas ferramentas de mineração de processos desenvolvidas e disponíveis no mercado e academia.

## 2.7 DISCO

O DISCO<sup>7</sup> é uma solução completa de mineração de processos da Fluxicon que realiza descobertas automáticas de processos. O DISCO viabiliza a criação automática de mapas gráficos de processos diretamente de logs com dados brutos. Com isso, ele permite que este mesmo mapa de processo criado possa ser animado, ilustrando dinamicamente para o usuário o fluxo das atividades do processo gerado para auxiliar na identificação de caminhos entre atividades que estejam sendo executadas em um intervalo de tempo acima da expectativa inicial. Além disso, o DISCO disponibiliza uma grande variedade de filtros possibilitando assim que o usuário minere os dados de seu log a ponto de descobrir informações que até então não eram perceptíveis por uma análise superficial. Sua saída de dados (*output*) pode ser no formato XES, o qual é necessário para ser utilizado nas análises do programa ProM.

## 2.8 ProM

O ProM<sup>8</sup> é um framework gratuito implementado em Java que suporta uma grande variedade de técnicas de mineração de processos no formato de plug-ins com o objetivo de auxiliar seus usuários a obter informações valiosas a partir de logs com dados brutos provenientes de processos. No ProM existem vários tipos de plug-ins para realizar diferentes tipos de análises em um log. Dentre estes plug-ins podemos citar por exemplo os plug-ins *Alpha algorithm*, *Genetic mining* e *Multi-phase mining* para análises com finalidade em mineração de fluxo de controle; já o *Social Network miner* e o *Staff Assignment miner* para análises sob uma perspectiva organizacional; o *Decision miner* para a análise sob a perspectiva de dados; o *Fuzzy Miner* para análise de processos flexíveis e pouco estruturados; e o *LTL Checker* para mineração de regras em um log<sup>9</sup>.

---

<sup>7</sup> <https://fluxicon.com/disco/> último acesso: 10/06/2018

<sup>8</sup> <http://www.promtools.org/doku.php> último acesso: 11/06/2018

<sup>9</sup> <http://www.promtools.org/doku.php?id=prom52> último acesso: 12/06/2018

## 3 A Mineração de Processos na Auditoria de Processos

Conforme exposto anteriormente a mineração de processos extrai conhecimentos e informações de logs de eventos de um ERP. Até então, os dados contidos nestes logs de eventos eram raramente aproveitados para a execução de análises voltadas para processos. Sendo assim, a mineração de processos veio auxiliar nesta tarefa disponibilizando técnicas e ferramentas para descoberta de processos, controles, informações e estrutura organizacional e social a partir dos logs de eventos<sup>10</sup>.

### 3.1 Mecânica da auditoria antes da mineração de processos

Para entendermos melhor porque a mineração de processos traria benefícios significativos para a área de auditoria, é necessário realizar uma introdução breve sobre como processos costumavam a ser executados, como por exemplo os processos de contabilidade. De acordo com JANS, ALLES e VASARHELYI (2012) algumas décadas atrás, a realização da contabilidade em uma empresa significava preencher à mão todos os livros de registro da área, como por exemplo o Livro Razão. Neste mundo de contabilidade manual, os dados que os auditores dependiam para verificar quais transações a empresa do cliente havia feito e como esses mesmos dados haviam sido contabilizados para tais transações vinham inteiramente de livros baseados em papel. Estes livros também eram complementados por pedaços de papel como cartas de confirmação por escrito e faturas marcadas à mão como pagas.

O problema desta arcaica mecânica não era somente por conta da evidente escassez de tecnologia, mas também com as limitações que a dependência de procedimentos manuais representam para o processo de auditoria. Livros de contabilidade escritos à mão sofrem do que, segundo JANS, ALLES e VASARHELYI (2012), podemos chamar de “*what you see is what you get*” ou problema de WYSIWYG. Com isso podemos dizer

---

<sup>10</sup> <http://bpmcenter.org/> último acesso em: 17/06/2018

que a única informação que o auditor possuía era o que estava literalmente sendo observado a sua frente. Logo, o auditor não tinha como verificar quem efetuou as inserções dos dados contábeis e nem quando as fizeram. Se estas inserções tivessem sido falsificadas, apagadas e reescritas, adicionadas ou modificadas seja isto após o seu registro inicial, ou por uma pessoa não autorizada a realizar a manipulação destes dados, o auditor pode detectar esta movimentação suspeita somente por meio de um exame minucioso das evidências físicas dos livros.

### **3.2 Valor agregado da mineração de processos na auditoria**

JANS, ALLES e VASARHELYI (2012), utilizam a definição do termo WYSIWYG, pois assim é possível transmitir claramente a diferença fundamental que os logs de eventos podem fazer na auditoria. Um log de eventos não é somente um “registro cronológico de atividades de um computador”, como é o caso dos livros de contabilidade citados anteriormente. Além disso, os logs de evento são potencialmente valiosos para a auditoria não somente por fornecer ao auditor mais dados para analisar. Mas também porque seus dados adicionais são registrados automaticamente e independentemente da ação de uma pessoa cujo comportamento é o assunto da auditoria.

Acessando um log de eventos, o auditor não está mais restrito ao registro WYSIWYG de transações inseridas pelo auditado. Ele possui também um conjunto independente do que se descreve por “metadados” sobre as circunstâncias sob as quais o auditado fez essas entradas. Estes metadados englobam muito mais do que simples registros de data e hora das transações, pois aproveitando estes dados de rastreamento, o log de eventos permite que o auditor reproduza o histórico de qualquer transação. Sendo assim, o auditor é capaz de traçar o relacionamento entre uma entrada em particular e seu autor para todas as transações registradas anteriormente, além de traçar todos os caminhos em que os processos de negócio estão sendo executados na prática até então (JANS, ALLES e VASARHELYI, 2012). Por exemplo, através da mineração de processos, o auditor tem a capacidade de comparar como processos, como por exemplo o de “compras a pagar”, foram realmente conduzidos com relação a como eles deveriam ser, ou para determinar como a demissão de um funcionário-chave impactou a segregação dos controles de impostos. Tal visibilidade de processo de negócio é muito complicada de se analisar a partir de uma transação isolada. Porém, torna-se viável quando os dados

transacionais são completados pelos metadados e pelo histórico contidos nos pacotes de eventos e tornados visíveis pelas técnicas de mineração de processos.

### 3.3 A verificação de conformidade em mineração de processos

A mineração de processos aborda técnicas para solucionar os problemas relacionados à processos de negócios. Com relação ao problema-foco do presente estudo, a mineração de processos dispõe da técnica de verificação de conformidade para trata-lo. A verificação de conformidade refere-se à análise da detecção de inconsistências entre o modelo de um processo e o correspondente log de execução real, além de quantificar e qualificar o nível de conformidade através do uso de métricas estatísticas como por exemplo o *fitness* (ROZINAT, 2010). A classe de algoritmos de mineração de processos responsável pela análise de conformidade de processos é chamada de técnicas de verificação de conformidade, pois estes algoritmos verificam a conformidade entre algum modelo já existente e o que é executado no dia-a-dia (ROZINAT, 2010).

O *fitness* é uma métrica que visa quantificar o quanto o comportamento observado no log se encaixa no modelo de processos descoberto ou modelado (LOPES, 2015). Um *fitness* perfeito indica que todos os casos de teste foram executados com trajetos de decisões idênticos aos do processo modelo. Logo, no caso em que o *fitness* não é perfeito, podemos concluir que ocorreram alguns desvios de trajeto na execução de um ou mais casos de teste (ACCORSI e STOCKER, 2012). O *fitness* é calculado a partir da técnica de verificação de conformidade denominada *replay*. Esta técnica consiste em processar vários casos de um log de teste em um modelo de processo afim de verificar a ocorrência de uma possível falta de alinhamento da sequência de atividades do log de teste com o seu modelo.

### 3.4 Replay

A técnica de Replay utiliza o log de eventos e o modelo de processo para ser executada. O log de teste é forçado a ser executado através do modelo do processo, mesmo que o modelo e log de teste não sejam compatíveis, as atividades que faltam ou sobram durante a execução são contabilizadas e inseridas no cálculo do fitness a fim de obter um índice de conformidade entre o log e o modelo (LOPES, 2015).

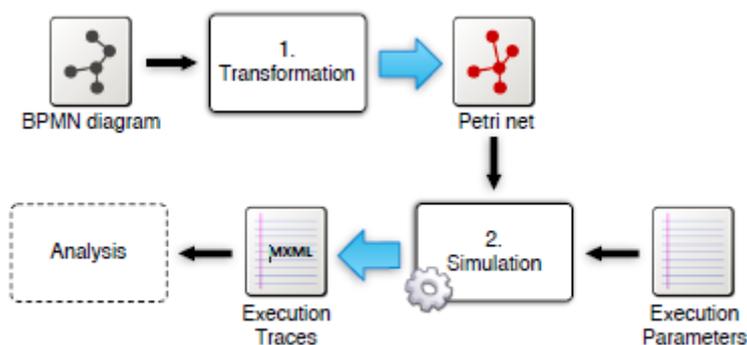
### 3.5 Trabalhos relacionados

Para mostrar a aplicabilidade da mineração de processos no âmbito de verificação de conformidade, a seguir será descrito um trabalho acadêmico que utiliza as técnicas de mineração de processos para auxiliar no âmbito da auditoria no trabalho de verificação da conformidade de processos de uma empresa do setor financeiro. Este trabalho busca validar um modelo de processo a-priori com a execução do mesmo no dia-a-dia. Nele é descrito um processo de solicitação de empréstimo ao banco.

Para manter a privacidade dos dados e indivíduos participantes deste processo, os autores ACCORSI e STOCKER (2012) basearam suas análises em informações que não foram provenientes do processo real. O log foi gerado através de uma simulação do processo em questão, a qual é controlada e parametrizada por condições padrões do negócio. Como por exemplo, respeitar o índice de aceitação de empréstimos, número de indivíduos envolvidos no processo e incluir os casos em que é necessário abortar o processo. O estudo simulou um período de seis meses de log, o qual inclui desvios do processo mencionado e contém rotas em que ocorrem violações do negócio. As análises realizadas pelos autores foram parametrizadas nas seguintes classes de requisitos de processos de negócio:

- *Autorização*: garante o controle de acesso para assegurar que somente usuários autorizados são permitidos para executar as atividades relacionadas ao processo.
- *Controle de uso*: garante restrições após o acesso a dados como retenção e limite de uso destes dados.
- *Separation of Duty* (SoD): são restrições associadas ao processo para limitar a possibilidade de agentes exercerem atividades e reduzir o risco de fraude. *Binding of Duties* (BoD) é relacionado à restrição em que um indivíduo pode atuar somente em um conjunto específico de atividades.
- *Conflict-of-interest* (CoI): previne a interação de informações sensíveis com organizações ou departamentos concorrentes para que os mesmos não façam parte do fluxo do processo.
- *Isolamento*: a execuções do processo em questão não interfere em outros.

Para executar estas regras foram feitas entrevistas não só com os funcionários do banco, mas também com auditores habituados com os requisitos acima. Após a coleta das informações necessárias e da consolidação destas em um processo modelo BPMN, foi iniciada a etapa de transformação deste BPMN em um modelo de rede de Petri, conforme é mostrado na Figura 6. Em seguida, foi utilizado nesta rede de Petri o software *Security Workflow Analysis Toolkit (SWAT)* a fim de simular um log semelhante ao processo real, incluindo casos de desvios relacionados às violações de regras de privacidade, as quais foram citadas acima. Como exemplo de uma violação, podemos citar a permissão da extrapolação de atrasos para a aprovação de um empréstimo em mais de 7 dias, este seria um caso de desvio no log simulado. Este log gerado foi armazenado no formato MXML para as análises de verificação de conformidade de processo no ProM.



**Figura 5 – Procedimento de confecção da simulação, (ACCORSI e STOCKER, 2012)**

Para executar as análises de verificação de conformidade e auxiliar nos procedimentos de auditoria de processos, foi utilizada a técnica de replay no log simulado. Esta técnica foi utilizada para encontrar desvios do fluxo padrão, como por exemplo, a falta da execução de alguma atividade do modelo. Podemos encontrar também pontos “outliers” como uma violação de um dos requisitos de segurança. Para a verificação de SoD, BoD, CoI e restrições de isolamento do processo foi utilizado o plug-in *LTL Checker* do ProM e para a verificação de tempo de execução de atividades foi utilizado o plug-in *SCIFF Checker*, que também pertence ao ProM. Por fim, para verificar as restrições de autorização da atuação de indivíduos no processo, foi necessário gerar um relatório que contém informações de qual usuário realizou qual tarefa e avalia-lo de forma manual.

Além do trabalho descrito anteriormente, há na literatura acadêmica mais exemplos de análises no âmbito da auditoria utilizando os testes de conformidade. AALST, HEE e VERDONK (2010), propuseram uma estrutura de auditoria que basicamente emprega a descoberta de processos e a verificação de conformidade e descreve alguns desafios da aplicação da mineração de processo à auditoria. Já na pesquisa de JANS et al. (2009), é salientado que o uso de técnicas de mineração de processo, utilizando o software *Internal Fraud Risk* (IFR2), pode agregar valor reduzindo o risco de fraude interna em empresas. Para verificar se um processo funciona de uma maneira que corresponda ao modelo projetado, JANS et al. (2009) emprega técnicas de descoberta de processo para reconstruir um modelo de processo que mostra o comportamento real do processo. Além de analisar as relações entre as atividades do processo e as pessoas envolvidas, eles utilizam verificação de conformidade para a verificação da segregação de restrições de trabalho. Este artigo é semelhante ao trabalho de JANS et al. (2011), que utilizam o processo de mineração para fins de auditoria interna de um ciclo de aquisições em uma instituição financeira, sendo que a principal diferença deste é o foco na segurança e privacidade dos dados.

### **3.6 Rede de Petri**

Redes de Petri é uma representação gráfica de processos em formato de diagrama. Ela utiliza um tipo especial de grafo que contém dois tipos de nós: posição e transição. Em notação gráfica, a posição é representada por um círculo, a transição é representada por um quadrado. A concatenação entre posições e transições é feita por uma estrutura chamada arco. Um arco liga somente uma posição em uma transição e vice-versa. Sendo assim, um arco entre duas posições ou duas transições é impossível (AALST e STHAL, 2011). Segue abaixo na Figura um exemplo de rede de Petri.

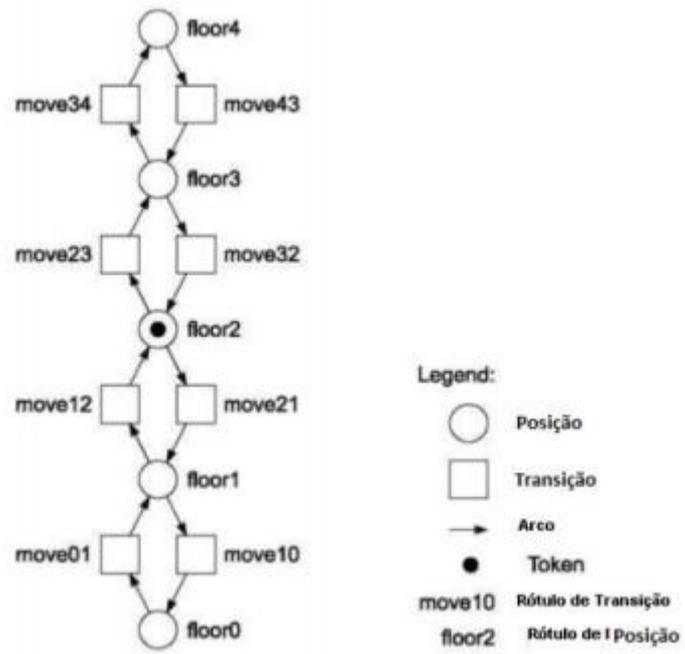


Figura 6 – Exemplo de uma rede Petri

## 4 Cenário de Aplicação

O presente projeto visa mostrar como algumas das técnicas de mineração de processos podem auxiliar na execução dos procedimentos da auditoria de processos trazendo simplicidade, agilidade e otimizações para a análise de conformidade de processos. Conforme já exposto anteriormente, a análise de conformidade é realizada a partir de dois logs de um mesmo processo. Sendo um deles o log de eventos referente ao modelo do processo em questão e o outro sendo o log da execução deste processo no dia a dia.

Logo, para mostrar tais benefícios, no primeiro momento foi necessário obter uma fonte que disponibilizasse informações sobre processos em formato de log de eventos. A fonte utilizada neste trabalho foi o log de eventos do sistema de informações de uma empresa participante do Desafio de Business Process Intelligence (BPI) deste ano (BPIC<sup>11</sup> - 2018).

O Desafio BPI de 2018 tem como contexto a dinâmica orçamentária agrícola da União Europeia. A União Europeia gasta uma grande fração do seu orçamento na política agrícola comum (CAP). Entre esses gastos estão os pagamentos diretos, que visam principalmente fornecer uma renda básica para os agricultores dissociados da produção. O resto do orçamento da CAP é gasto em despesas relacionadas com o mercado e desenvolvimento rural. Os processos que regem a distribuição destes fundos estão sujeitos a regulamentos complexos, capturados na legislação da UE e nacional. Por este motivo, os estados membros são obrigados a operar um sistema integrado de administração e controle (IACS), que inclui sistemas de TI para apoiar os complexos processos de distribuição de subsídios.

O processo considerado no desafio BPI 2018 abrange o tratamento de pedidos de pagamentos diretos da UE para agricultores alemães do Fundo Europeu Agrícola de Garantia. O processo se repete a cada ano com pequenas alterações devido a mudanças nos regulamentos da UE. Cerca de 10% dos casos estão sujeitos a uma inspeção mais rigorosa no local. Um log relacionado a este processo, *payment application*, foi

---

<sup>11</sup> Business Process Intelligence Challenge (BPIC), último acesso 07/07/2018: <http://www.win.tue.nl/bpi/doku.php?id=2018:challenge>

disponibilizado pelo Desafio com um prévio tratamento dos dados, sendo assim não foi necessário nenhum tipo de análise para identificar e retirar dados incorretos que poderiam impactar nos resultados das análises a serem executadas neste trabalho.

Contudo, o único tipo de log disponibilizado pelo Desafio foi o log do dia-a-dia da empresa relacionado ao seu processo de pagamento de contas. Sendo que para realizar o objetivo principal deste trabalho, verificar conformidade entre o modelo e a execução real de um processo, se faz necessário criar um modelo a partir do log disponibilizado para servir de base para esta análise de conformidade em log modelo e log de execução real do processo de pagamento de contas.

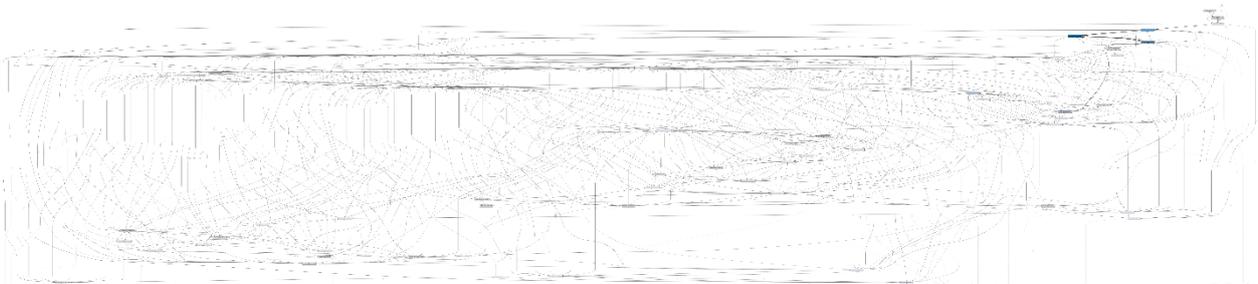
Para tal, os passos executados foram:

- Descobrir atividades do log;
- Analisar a dinâmica entre atividades;
- Identificar campos definidores do log;
- Idealização do log modelo e segregação do log original.

#### **4.1 Descobrir as atividades do log**

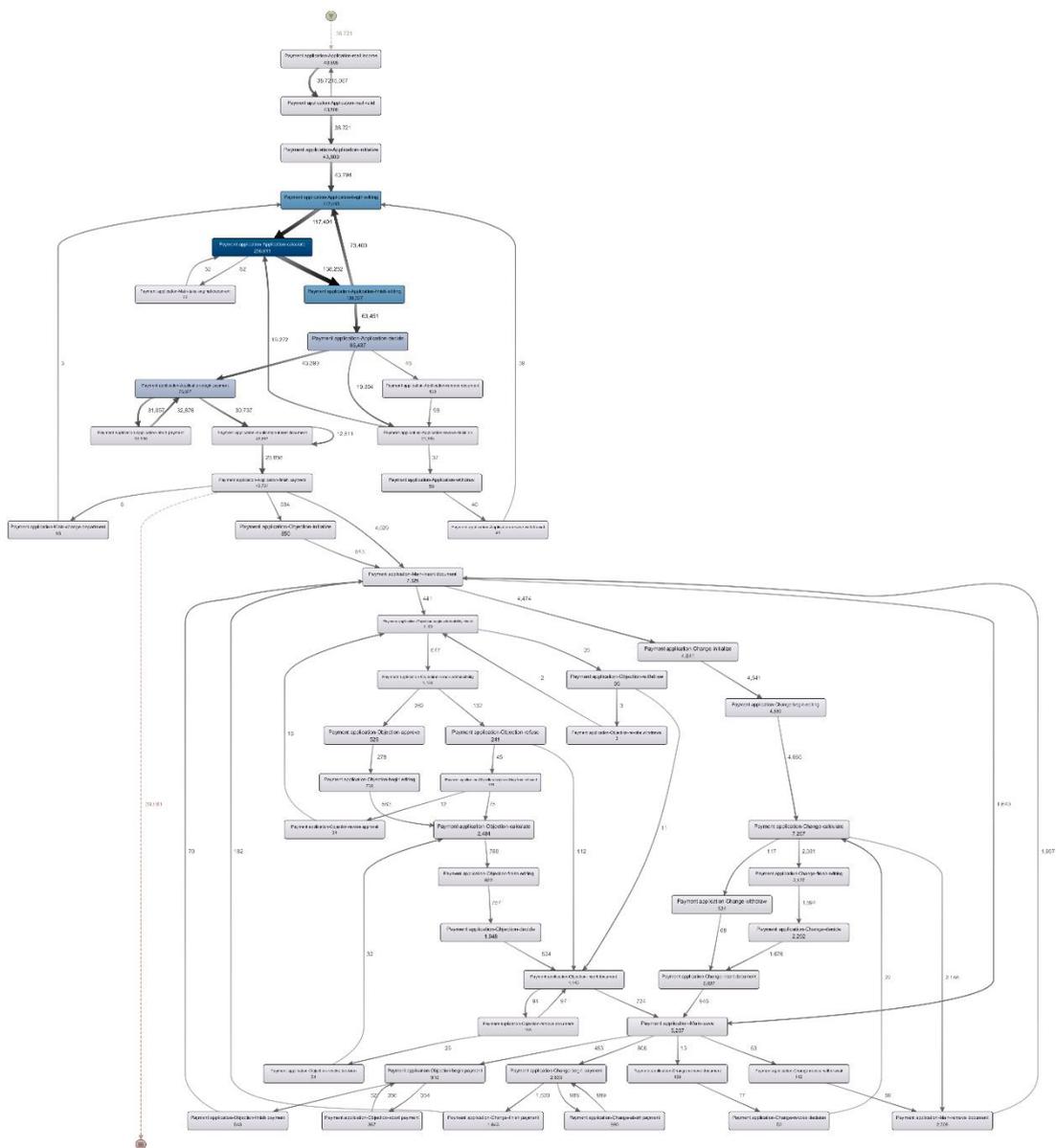
O primeiro passo para iniciar as análises dos dados provenientes do log do processo de pagamento de contas foi realizar a importação do log para o *software* Disco. Com ele, foi possível realizar a primeira etapa dos procedimentos de mineração de processos, a descoberta das atividades do processo. O Disco nos permite analisar os dados do log de eventos por meio de uma visão gráfica de um modelo de processos. Tal modelo dispõe de forma sequencial todas as atividades do processo e todos os relacionamentos entre cada uma destas atividades.

Ao importar o log de eventos para o Disco, um modelo BPM com todos os caminhos e atividades é gerado automaticamente, como é ilustrado na Figura 7.



**Figura 7 – Modelo com todos os caminhos e atividades do processo**

Devido à complexidade do log analisado, o primeiro modelo gerado pelo Disco não oferece muita informação sobre o processo de pagamento de contas. Porém, como nesta ferramenta o grau de detalhamento das atividades e trajetos não são determinísticos, foi possível manipular o modelo gerado a fim de estudar as suas atividades e as correlações destas. Sendo assim, reduzindo ao máximo a quantidade de caminhos alternativos e mantendo o número máximo de atividades do log de eventos, obteve-se como resultado o modelo mostrado na Figura 8 a seguir.



**Figura 8 – Modelo intermediário com o máximo de atividades e mínimo de caminhos**

Continuando a aplicar filtros na visão gráfica mantendo o mínimo de caminhos e agora reduzindo também ao máximo o número de atividades, obteve-se, através do Disco, o modelo representado na Figura 9.

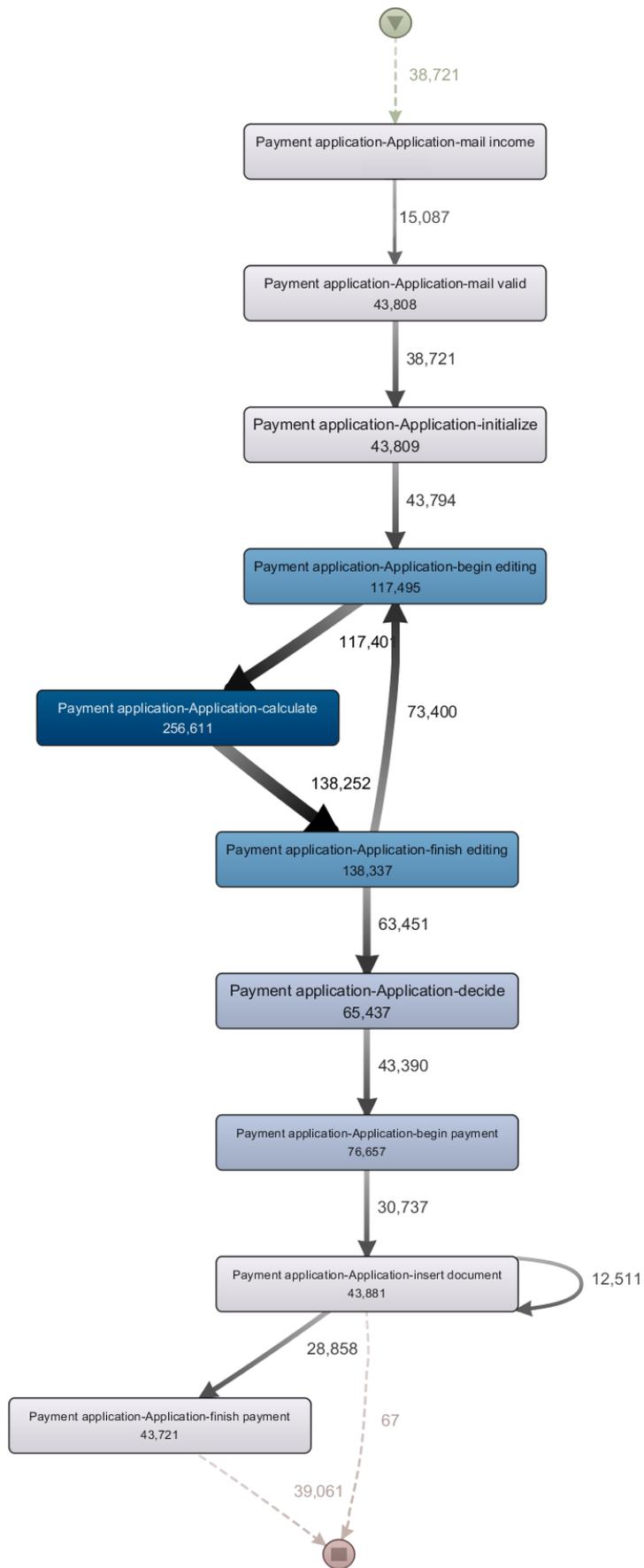


Figura 9 – Modelo simplificado com mínimo de atividades e caminhos

A partir deste modelo, evidenciamos as principais atividades executadas neste processo. Nele, nota-se a presença de atividades entre as cores cinza e azul escuro. Nos modelos gerados pela ferramenta Disco, as cores das atividades representam sua frequência de execução. Logo, as atividades em cinza são atividades pouco frequentes e as em azul escuro são atividades mais frequentes. Além das cores, é perceptível a presença de pesos em cada caminho do processo, que estão indicados como setas que interligam atividades. Este peso é referente ao tempo gasto na transição de cada atividade.

O log *Payment Application* é composto pelos sub processos *Application*, *Change*, *Main* e *Objection*. As atividades mencionadas anteriormente são referentes ao sub processo *Application*, o qual representa 94,26% dos casos, sendo assim o sub processo mais relevante do log de eventos. A distribuição das porcentagens entre os quatro sub processos é apresentada abaixo na Tabela 1.

**Tabela 1 – Frequência relativa dos sub processos**

Sub processo	Frequência relativa
Application	94,26%
Change	2,93%
Main	1,54%
Objection	1,28%

## 4.2 Analisar a dinâmica entre atividades

Após identificar as atividades constituintes do processo de pagamento de contas, a próxima etapa será analisar a dinâmica das interações entre estas atividades. Para que seja possível entender melhor o papel de cada atividade e saber quais são os insumos e consequências de cada uma delas. A seguir, discutimos a da dinâmica entre as principais atividades anteriormente encontradas.

Como é visto na Figura 9, para inicializar o processo, é preciso receber uma solicitação de pagamento por e-mail, por este motivo, a primeira atividade é o recebimento deste pedido, denominada *mail income*. A próxima etapa é a confirmação desta solicitação, *mail valid*. Quando o pedido é confirmado, efetua-se a atividade *initialize*, em que de fato ocorre o início do processo de pagamento. Em seguida é efetuada

a atividade *begin editing*, cuja finalidade é começar a editar os dados de entrada para uma formatação que possibilite dar sequência para a atividade *calculate*, a qual calcula o valor a ser recebido pelo agricultor alemão de forma automática.

Porém, como a etapa de edição é realizada manualmente, erros de digitação podem ocorrer e conseqüentemente o cálculo estará incorreto. Para evitar isto, após a atividade de cálculo do pagamento é realizada uma confirmação dos dados pela atividade *finish editing*. Se algo estiver incoerente nesta fase, será necessário eliminar as respectivas falhas/erros redirecionando a sequência de execução das atividades de volta para a atividade *begin editing*. Caso o cálculo corresponda ao valor previsto, a próxima atividade será a *decide*. Esta atividade é a responsável por decidir como será pago o valor devido, por exemplo, o número de parcelas. Imediatamente, o início do processo do pagamento é efetuado pela atividade *begin payment* que é seguida pela inserção de documentos pela atividade *insert document*.

Conforme exposto anteriormente, há alterações anuais no regulamento da União Europeia e isto também pode provocar mudanças no padrão dos documentos a serem anexados. Neste sentido, a inserção de documentos pode ocorrer inúmeras vezes seja pela motivação descrita ou por outra falha humana. Entretanto, após tais inserções há dois caminhos que podem ser seguidos: o fim deste processo por alguma interferência externa, caminho menos provável, ou seguir para a execução da última atividade *finish payment*, a qual é responsável por finalizar o processo de pagamento de maneira adequada antes do término do processo.

Em suma, o fluxo principal do processo de pagamento de contas pode ser resumido na seguinte sequência:

Inicializado

1. mail income (solicitação de pagamento por e-mail)
2. mail valid (validação do e-mail)
3. initialize (inicialização da análise de pagamento)
4. begin editing (começo da edição)
5. calculate (cálculo)
6. finish editing (finalização da edição)
7. decide (decisão)
8. begin payment (inicialização do pagamento)
9. insert document (inserção de documento)
10. finish payment (finalização do pagamento)

Finalizado

### 4.3 Identificar campos definidores do log

O passo seguinte foi executar uma análise no log em sua forma tabular. Nesta visão é possível observar os campos contidos no log como é exemplificado na Figura 10. Isto permitiu avaliar qual tipo de informação era necessária para o trabalho e quais não eram.

	Case ID	Activity	Resource	Complete Timestamp
1	-18008611495569447	Payment application-Application-mail income	0,n/a	2015/05/07 19:00:00.000
2	-18008611495569447	Payment application-Application-mail valid	0,n/a	2015/05/07 19:00:00.000
3	-18008611495569447	Payment application-Application-initialize	Document processing automaton	2015/11/06 08:21:27.376
4	-18008611495569447	Payment application-Application-begin editing	727350	2015/11/10 14:17:50.045
5	-18008611495569447	Payment application-Application-calculate	727350	2015/11/10 14:17:57.085
6	-18008611495569447	Payment application-Application-calculate	Processing automaton	2015/11/20 23:46:57.811
7	-18008611495569447	Payment application-Application-calculate	Processing automaton	2015/11/23 08:34:49.398
8	-18008611495569447	Payment application-Application-calculate	727350	2015/12/03 13:14:05.938
9	-18008611495569447	Payment application-Application-finish editing	727350	2015/12/03 13:14:07.318
10	-18008611495569447	Payment application-Application-decide	520882	2015/12/05 11:00:02.673
11	-18008611495569447	Payment application-Application-begin payment	DP-Z	2015/12/16 08:55:49.972
12	-18008611495569447	Payment application-Application-abort payment	DP-Z	2015/12/16 13:43:22.109
13	-18008611495569447	Payment application-Application-begin payment	DP-Z	2015/12/17 18:17:23.983
14	-18008611495569447	Payment application-Application-abort payment	DP-Z	2015/12/18 10:23:26.222
15	-18008611495569447	Payment application-Application-begin payment	DP-Z	2015/12/18 18:49:16.018
16	-18008611495569447	Payment application-Application-finish payment	DP-Z	2016/02/18 04:56:22.774

Figura 10 – Visão tabular do log

Neste log, as colunas utilizadas foram as seguintes:

- Case ID, coluna referente ao identificador único do caso;
- Activity, contém o nome do log, seguido do nome do sub processo e por fim o nome da atividade (processo-sub processo-atividade);
- Resource, contém o tipo de fonte da informação. Isto é, se a mesma é de origem manual, automática ou até mesmo proveniente de uma variável codificada, por questões de privacidade;
- Complete timestamp, coluna referente à variável de tempo no formato: YYYY/MM/DD HH:MM:SS:MMM.

Os demais campos continham dados irrelevantes para esta pesquisa. Sendo assim, foi aproveitada uma funcionalidade do Disco que permitiu a exclusão destes dados considerados irrelevantes, como foi o caso da coluna “Variant”, conforme é ilustrado abaixo na Figura 11.

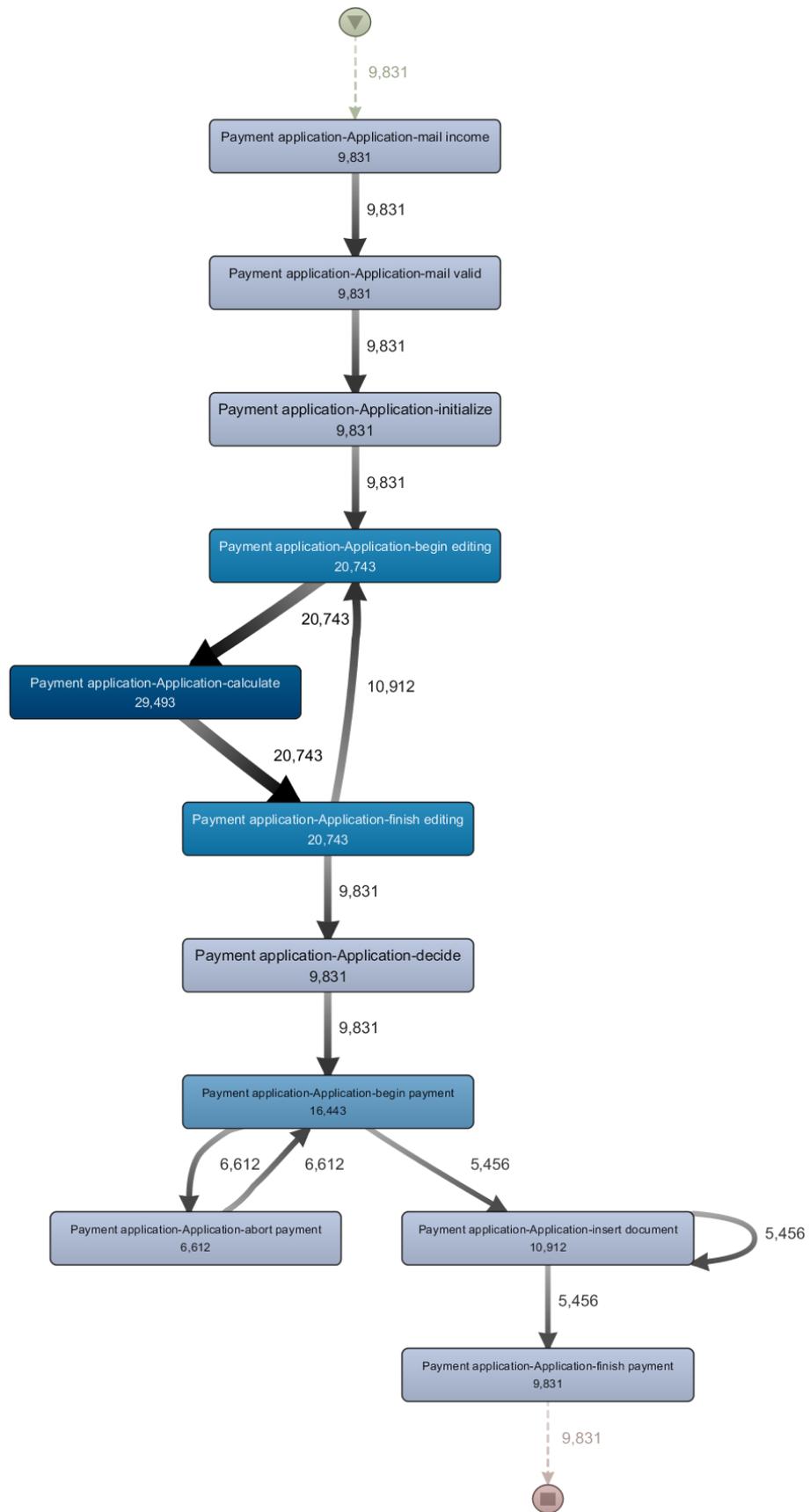
Case ID	Activity	Resource	Complete Timestamp	Variant	
1	-18008611495569447	Payment application-Application-mail income	0,n/a	2015/05/07 19:00:00.000	Variant 8
2	-18008611495569447	Payment application-Application-mail valid	0,n/a	2015/05/07 19:00:00.000	Variant 8
3	-18008611495569447	Payment application-Application-initialize	Document processing automaton	2015/11/06 08:21:27.376	Variant 8
4	-18008611495569447	Payment application-Application-begin editing	727350	2015/11/10 14:17:50.045	Variant 8
5	-18008611495569447	Payment application-Application-calculate	727350	2015/11/10 14:17:57.085	Variant 8
6	-18008611495569447	Payment application-Application-calculate	Processing automaton	2015/11/20 23:46:57.811	Variant 8
7	-18008611495569447	Payment application-Application-calculate	Processing automaton	2015/11/23 08:34:49.398	Variant 8
8	-18008611495569447	Payment application-Application-calculate	727350	2015/12/03 13:14:05.938	Variant 8
9	-18008611495569447	Payment application-Application-finish editing	727350	2015/12/03 13:14:07.318	Variant 8
10	-18008611495569447	Payment application-Application-decide	520882	2015/12/05 11:00:02.673	Variant 8
11	-18008611495569447	Payment application-Application-begin payment	DP-Z	2015/12/16 08:55:49.972	Variant 8
12	-18008611495569447	Payment application-Application-abort payment	DP-Z	2015/12/16 13:43:22.109	Variant 8
13	-18008611495569447	Payment application-Application-begin payment	DP-Z	2015/12/17 18:17:23.983	Variant 8
14	-18008611495569447	Payment application-Application-abort payment	DP-Z	2015/12/18 10:23:26.222	Variant 8
15	-18008611495569447	Payment application-Application-begin payment	DP-Z	2015/12/18 18:49:16.018	Variant 8
16	-18008611495569447	Payment application-Application-finish payment	DP-Z	2016/02/18 04:56:22.774	Variant 8

Figura 11 – Exclusão da coluna “Variant”

#### 4.4 Idealização do log modelo e segregação do log original

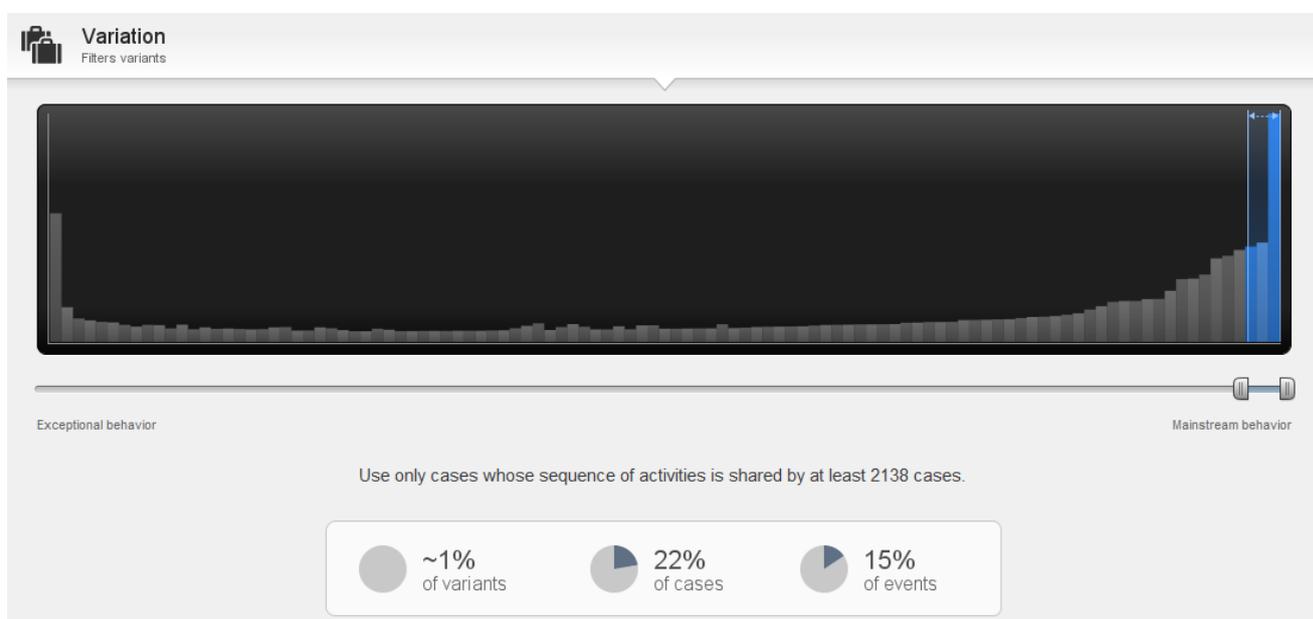
Após a compreensão da dinâmica entre as atividades do processo de pagamento de contas e a realização da retirada dos dados irrelevantes deste log, avançamos para a etapa de idealização do log modelo para posteriormente efetuar a segregação do log nos dois sub logs de eventos mencionados no início deste capítulo. Sendo que um deles representa o log relacionado ao modelo do processo de pagamento de contas e o outro representa o log de execução real do processo.

Na idealização do log modelo, foram utilizadas as atividades consideradas principais ilustradas na Figura 9, porém com uma modificação. Foi incluído um caminho alternativo após a execução da atividade *begin payment*. Sendo assim, após o início da análise do pagamento da conta, além dos caminhos mencionados anteriormente, o processo pode ser abortado na atividade *abort payment* conforme é mostrado na Figura 12, ilustração do modelo final.



**Figura 12 – Modelo de referência**

Com o modelo do processo de referência já idealizado, a etapa seguinte a ser realizada é a segregação do log original. Para isto, foi utilizado o filtro “Variation” da ferramenta Disco somente nos casos que caracterizavam o modelo idealizado, Figura 12. Este filtro, conforme ilustrado na Figura 13, serve para retirar partes de um log. Neste caso, foi retirado em torno de 20% dos casos do log modelo e exportado em formato XES. Após retirar o log modelo, os casos remanescentes do log original, correspondente ao restante dos casos somados aos 80% dos casos referentes ao modelo, foram exportados em formato XES representando o log de execução real da empresa, o qual passaremos a chamar de log de teste.



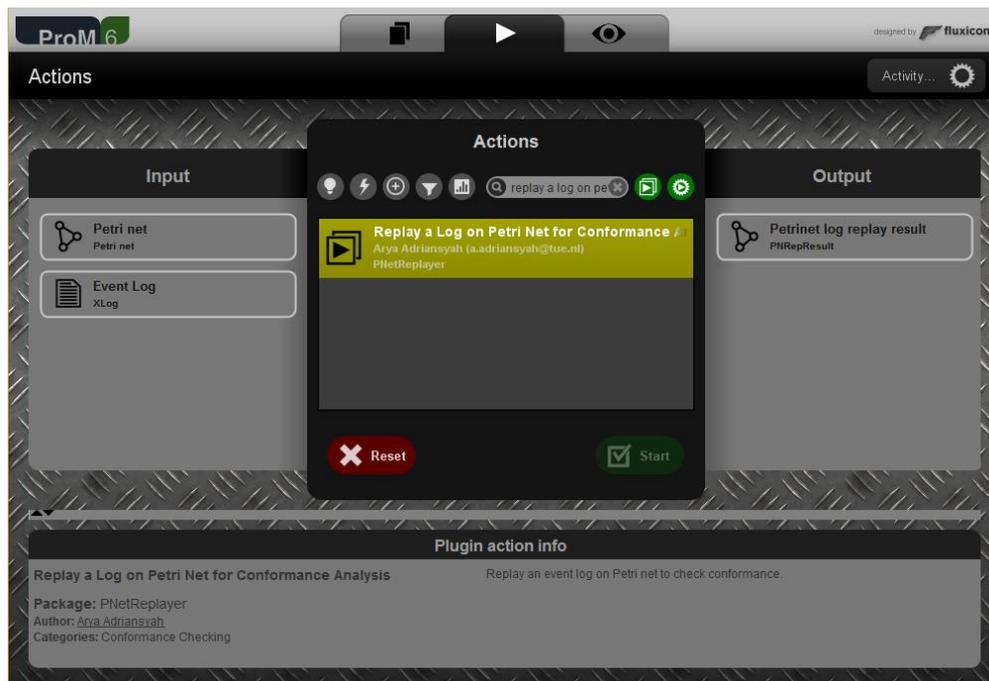
**Figura 13 – Filtro “Variation” considerando 20% dos casos do log modelo**

#### **4.5 Aplicar plug-in de verificação de conformidade**

Após a preparação dos logs modelo e do log de teste, é possível utilizá-los como inputs para a execução do plug-in de conformidade do ProM. A primeira etapa a ser executada foi importar os logs modelo e de teste para o ProM no formato “ProM log files (XESLite – MapDB (slow random access))”. O plug-in utilizado para a verificação de conformidade foi o “*Replay a Log on Petri Net for Conformance Analysis*”. Este plug-in visa, a partir da técnica de *replay*, comparar um log teste com um modelo para que assim possa ser gerada uma rede Petri com várias informações sobre como um processo está

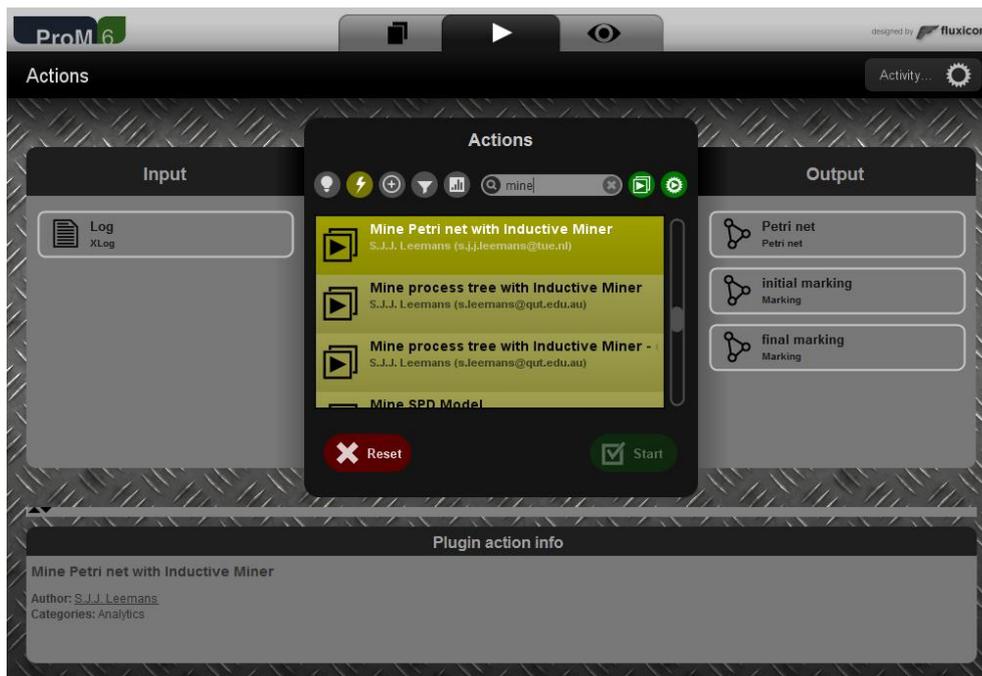
sendo executado com relação ao seu modelo de referência, tais como: o fitness entre o log e o modelo, quantificação e evidência dos caminhos alternativos ao modelo, pontos de sobrecarga de recursos e análise do desempenho individual das atividades modeladas.

Conforme mostrado na Figura 14, plug-in “*Replay a Log on Petri Net for Conformance Analysis*” recebe como input o log a ser testado em formato XES e o modelo a ser a base de comparação da análise no formato de modelo de rede de Petri.



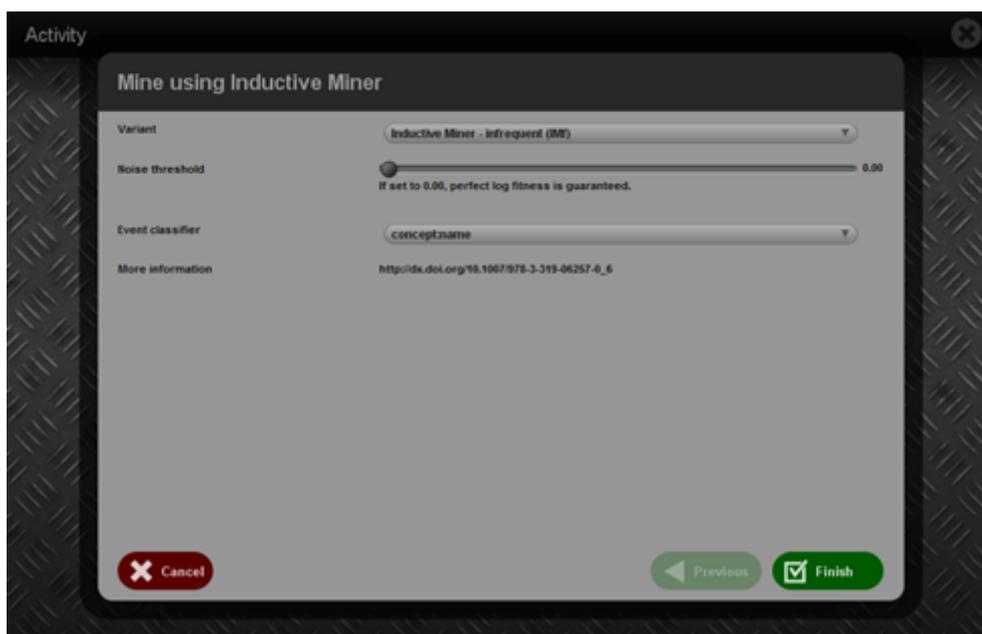
**Figura 14 – Input e Output do plug-in de verificação de conformidade**

Para gerar a rede Petri modelo, foi utilizado o plug-in “*Mine Petri net with Inductive Miner*”. Conforme mostrado na Figura 15, este plug-in possui como input um log XES, neste caso iremos utilizar o log modelo como input para gerar a rede de Petri.



**Figura 15 – Input e Output do plug-in gerador da rede de Petri**

Ao executar o plug-in acima, uma caixa de configurações é exibida na tela, como é ilustrado na Figura 16. Nesta etapa é permitido ao usuário executar determinados ajustes para evitar ruídos na produção da rede de Petri. Como pretendemos um fitness perfeito com o log modelo, o ruído foi parametrizado em zero para gerar a rede de Petri da Figura 17.

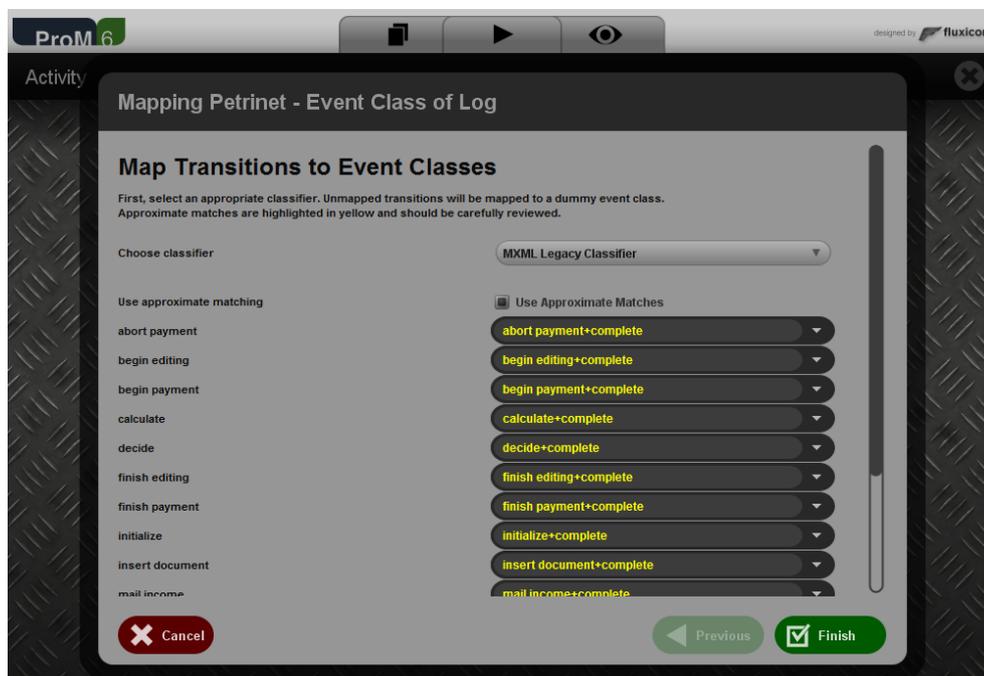


**Figura 16 – Configurações de produção da rede de Petri**



Neste plug-in, a rede de Petri é representada um pouco diferente da notação original. Como é visto na Figura 17, os retângulos representam as atividades do processo, os círculos representam transições entre atividades e os retângulos pretos representam mais de uma possibilidade de transição entre atividades. Como por exemplo, um loop ou uma decisão de caminho a ser tomado.

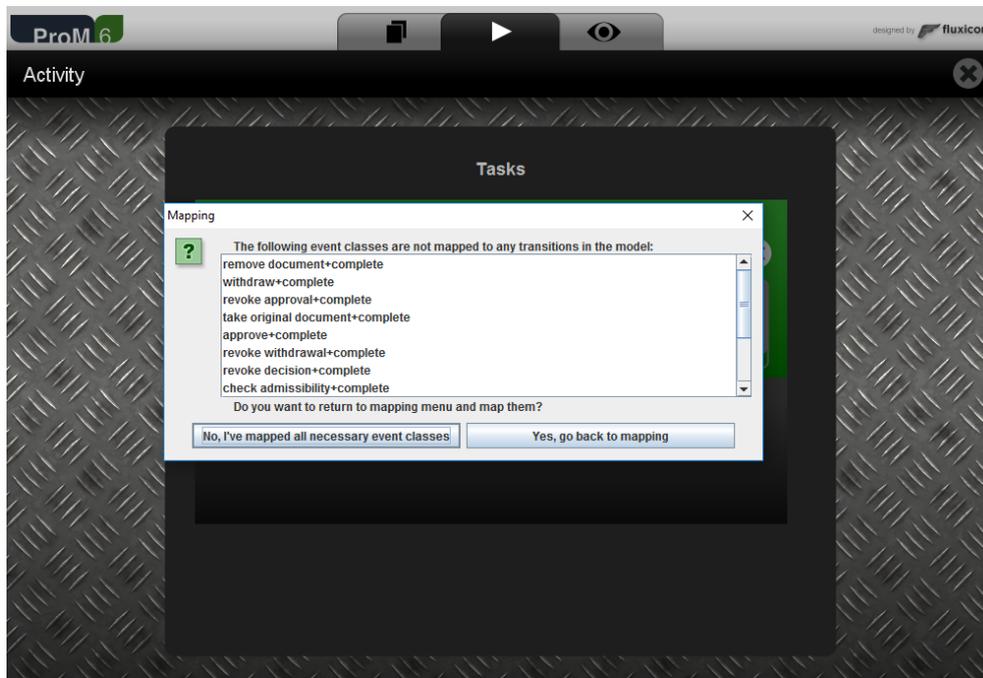
Com a rede de Petri gerada, podemos executar o plug-in de verificação de conformidade inserindo como input o log em formato XES de teste e a rede de Petri produzida anteriormente. Ao rodar o plug-in uma caixa de configurações é exibida. Conforme mostrado na Figura 18.



**Figura 18 – Configurações iniciais do plug-in de verificação de conformidade**

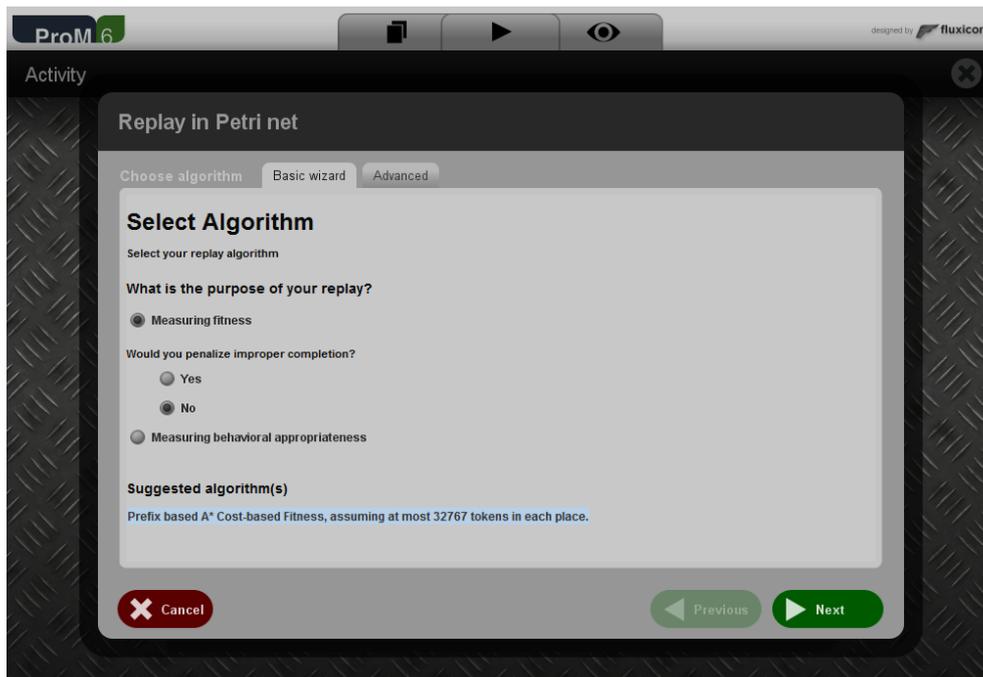
Nas configurações iniciais do plug-in de verificação de conformidade são mostradas todas as atividades do processo modelo e as atividades do log teste correspondentes a elas. Estas atividades estão marcadas de amarelo pois indicam resultados semelhantes às atividades do modelo. Em seguida, se o log de testes possuir atividades que não foram mapeadas no modelo é exibida uma caixa, como na Figura 19, informando as atividades que não foram mapeadas perguntando se o usuário gostaria de voltar à etapa anterior para mapeá-las de acordo com o modelo. Isto quer dizer que o log teste em questão possui atividades adicionais que não estão presentes no modelo. Estas atividades representarão as

atividades em não conformidade com o modelo, sendo assim não foi necessário mapear nenhuma atividade adicional.



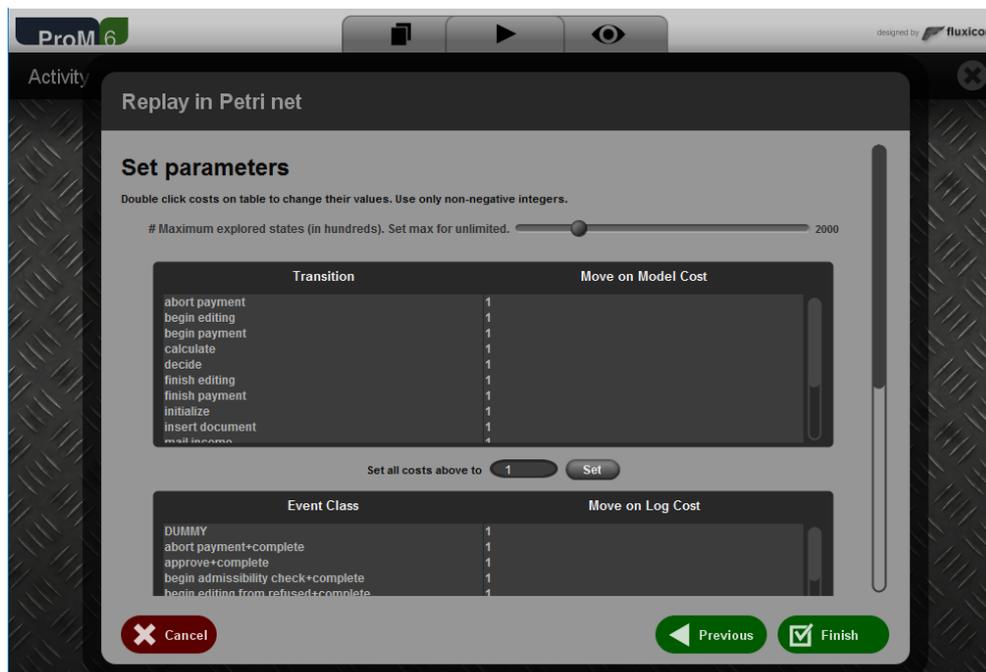
**Figura 19 – Caixa de diálogo informando atividades sem mapeamento com modelo**

Seguindo com as configurações do plug-in, o ProM permite ao usuário uma escolha mais intuitiva sobre o algoritmo de *Replay* a ser utilizado na análise de conformidade, conforme a Figura 20. A finalidade do presente trabalho é checar o grau de alinhamento da execução do fluxo do log de eventos teste com o modelo, em outras palavras, mensurar o *fitness* do processo. Com isto, foi selecionada a opção “*Measuring fitness*” e marcada a opção referente à penalização de trajetos completados impropriamente.



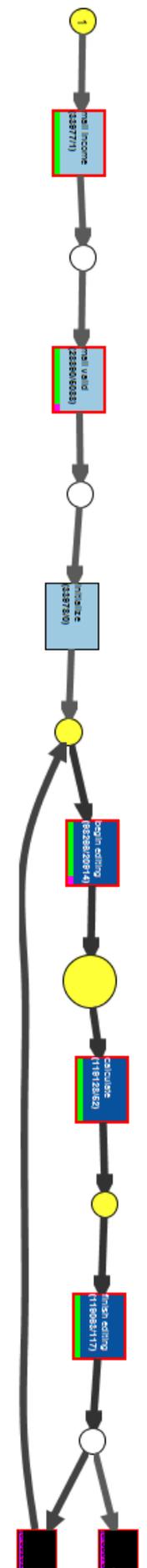
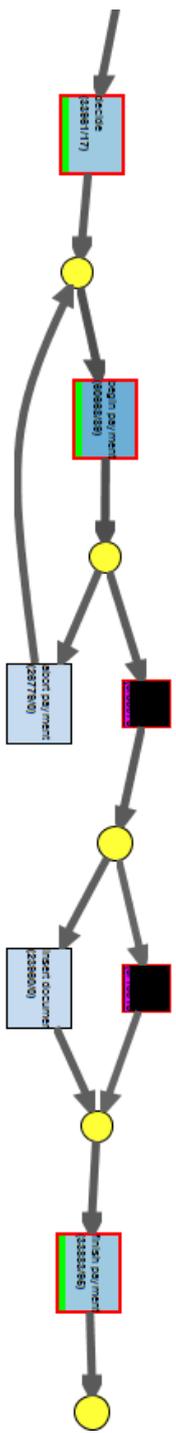
**Figura 20 – Configurações de finalidade das análises do plug-in**

Após estas configurações, o ProM permite ao usuário ajustar o custo de cada movimento entre atividades, conforme a Figura 21. Nesta análise inicial não serão adicionados custos diferenciados, pois o intuito primordial é avaliar o fitness puro entre o log de teste e o modelo.



**Figura 21 – Configurações de custo de movimentos**

Com todas as configurações anteriores devidamente estruturadas, o plug-in está pronto para ser executado. Ao final do processamento do plug-in, foi obtido como resultado uma rede de Petri com informações adicionais como cores e formas específicas, como pode ser visto na Figura 22.



**Figura 22 – Output do plug-in “*Replay a Log on Petri Net for Conformance Analysis*”**

#### **4.6 Informações fornecidas pelo plug-in de conformidade**

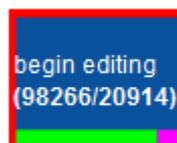
A seguir serão explicadas as informações adicionais empregadas pelo plug-in de conformidade na rede de Petri e também serão exemplificados alguns casos do output gerado.

#### **4.7 Cores das atividades**

As atividades mais escuras compreendem as atividades executadas com mais frequência. Já as atividades mais claras, são atividades executadas em um número menor de vezes. Seguindo esta linha de raciocínio, podemos observar na Figura 22 que as atividades relacionadas à edição de pagamento ocorrem com mais frequência do que as demais. Isto pode indicar algum problema no recebimento das informações de pagamento, ou até mesmo algum problema envolvido especificamente nestas atividades, como por exemplo, um retrabalho por falha na edição de um pagamento.



**Figura 23 – Atividade “*finish payment*”, output do plug-in de replay de conformidade**



**Figura 24 – Atividade “*begin editing*”, output do plug-in de replay de conformidade**

#### 4.8 Barra verde

As atividades mais escuras indicam a frequência de casos em que o log executou a tarefa em questão em sincronismo com o modelo. Com isto, a partir da Figura 23, podemos concluir que a frequência de casos conformes com o modelo é bastante alta.

#### 4.9 Barra lilás

Indica a frequência de casos com execuções divergentes em relação ao modelo. Na Figura 24, esta barra pode ser vista na parte inferior da atividade, indicando a presença de casos não conformes com o modelo. Já na Figura 23, os casos em que ocorrem divergências são tão irrisórios que a barra lilás não é vista.

#### 4.10 Atividades com borda vermelha

Sinaliza atividades em que o log teste não as executou de forma sincrônica ou correta com o modelo de referência. Outra forma de evidenciar esta informação é observando o interior das atividades como mostra a Figura 24. Nela podemos ver que em mais de 98.000 casos, esta atividade está alinhada com a execução do modelo. Em contrapartida, em mais de 20.000 casos esta atividade nem sequer foi executada. Ou seja, em mais de 20.000 casos a atividade de edição de pagamento não foi realizada.

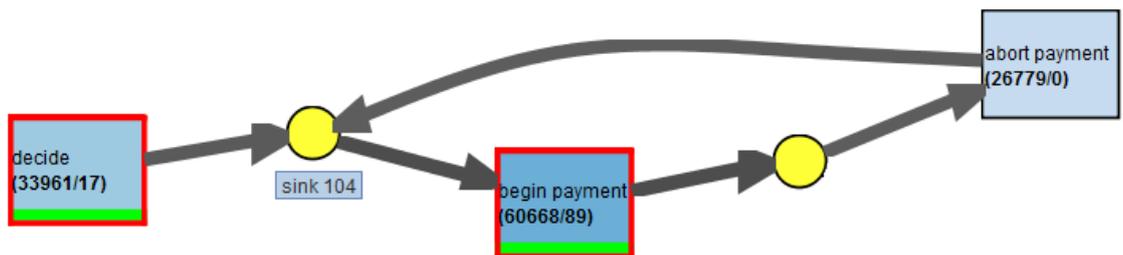


Figura 25– Ilustração do possível trajeto após atividade “decide”

▼ *Elements Statistics*

Selected elements\* sink 104

Marking	Move on Log	# (Freq)	#Traces
[sink 104]	approve+complete	106	106
	begin admissibility check+complete	185	120
	begin editing from refused+complete	24	18
	begin editing+complete	241	231
	begin payment+complete	240	239
	calculate+complete	695	281
	check admissibility+complete	183	120
	decide+complete	402	280

Figura 26 – Caminhos executados pelo log teste logo após execução da atividade “decide”

#### 4.11 Círculos brancos e amarelos

Os círculos brancos indicam trajetos que foram seguidos em conformidade com o modelo. Já círculos amarelos indicam ocorrências de movimentos fora do modelo, ou seja, não conformes. Como por exemplo na Figura 25, após a execução da atividade “decide” o modelo permite somente uma transição para a atividade “begin payment”. Porém existem casos no log de teste em que após a execução da atividade “decide”, ocorreram transações para outras atividades que não estão no modelo. Estas mudanças de trajetórias e suas respectivas frequências podem ser vistas na janela auxiliar do plug-in de conformidade, como mostrado na Figura 26.

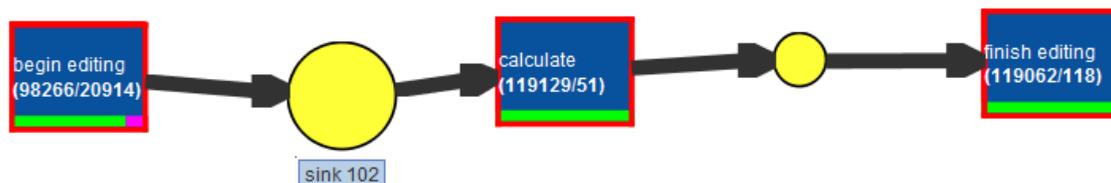


Figura 27 – Caminhos executados pelo log teste logo após execução da atividade “begin editing”

▼ Elements Statistics

Selected elements\* sink 102

Marking	Move on Log	# (Freq)	#Traces
	begin payment+co...	56	34
	calculate+complete	110597	30103
	check admissibilit...	43	23
	decide+complete	20891	20564
	finish editing+com...	77	63
	finish payment+co...	25	25
	initialize+complete	28	28
	insert document+...	350	212
	refuse+complete	18	18
	remove document...	152	130
	revoke decision+c...	19839	19143

Figura 28 – Caminhos executados pelo log teste após execução da atividade “begin editing”

#### 4.12 Tamanho dos círculos

Os círculos maiores indicam movimentações alternativas mais frequentes. Na Figura 27, podemos perceber a presença de um círculo maior do que os demais após a atividade “begin editing”. Este círculo sinaliza que, em relação à média de todos os trajetos alternativos executados no log, ocorreu uma frequência maior de trajetos alternativos exatamente após a execução da atividade “begin editing”, conforme é evidenciado na Figura 28.

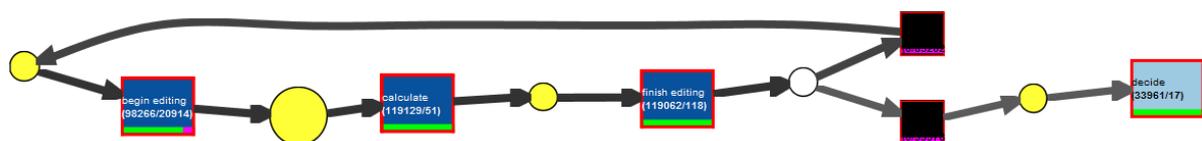


Figura 29 – Cada decisão de trajeto é representada por um retângulo preto

#### 4.13 Retângulos pretos

Indicam pontos de decisões de trajetórias a serem seguidas após a execução de uma atividade, podendo ser um loop em uma ou mais atividades, uma execução de

atividades em paralelo ou conforme é mostrado na Figura 29 uma decisão de caminho excludente (XOR).

#### 4.14 Comparação dos resultados

Neste tópico serão comparadas execuções do plug-in “*Replay a Log on Petri Net for Conformance Analysis*” com diferenças no ajuste de custos. Como referência comparativa, foi levado em consideração a execução sem filtros do plug-in, a qual já foi descrita nos tópicos 4.5 e 4.6. Já como variáveis comparativas, serão utilizadas uma execução do plug-in com alterações no custo das atividades do campo “*Move on Model Cost*” e uma execução do plug-in com alterações no custo das atividades do campo “*Move on Log Cost*”.

- **Comparação com alterações no campo “*Move on Model Cost*”**

A técnica de replay visa analisar o fluxo de execução das atividades do log em relação ao modelo de forma sequencias, com base na posição do enfileiramento das atividades. Ou seja, a leitura das atividades, a qual é denominada por “movimento”, ocorre baseada na posição das atividades. Por exemplo, em um modelo existem as atividades A, B, C, D e E, e segundo ele, elas devem ser executadas na seguinte sequência: ABCDE, porém ao analisar o log de teste, nota-se que foi executada a seguinte sequência: ACDE. Isto quer dizer que ocorreu um movimento somente no modelo, pois a atividade B não foi executada no log de teste, logo não foi possível executar a leitura, movimentação, da atividade relacionada ao log de teste.

Tendo isto posto, entende-se que o campo “*Move on Model Cost*” representa um filtro que permite ao usuário estabelecer um custo quando houver movimentação somente em uma atividade no modelo, conseqüentemente creditando uma penalização ao log de teste. Em outras palavras ocorrerá uma penalização quando ocorrer uma eventual ausência de uma determinada atividade do log em relação ao modelo analisado (AALST, ADRIANSYAH e DONGEN, 2012). Logo, se utilizarmos o filtro “*Move on Model Cost*” para o exemplo acima, o log seria penalizado referente a uma única atividade, B. O peso desta penalização, por sua vez, influenciaria no resultado do fitness do log.

Com relação à análise do processo de pagamento de contas, foi parametrizado um peso maior para atividades consideradas importantes a serem executadas. Com base nesta informação, as atividades “begin payment”, “calculate” e “finish payment” receberam peso 5 no filtro “Move on Model Cost”, enquanto as outras receberam peso 1, conforme mostrado na Figura 30.

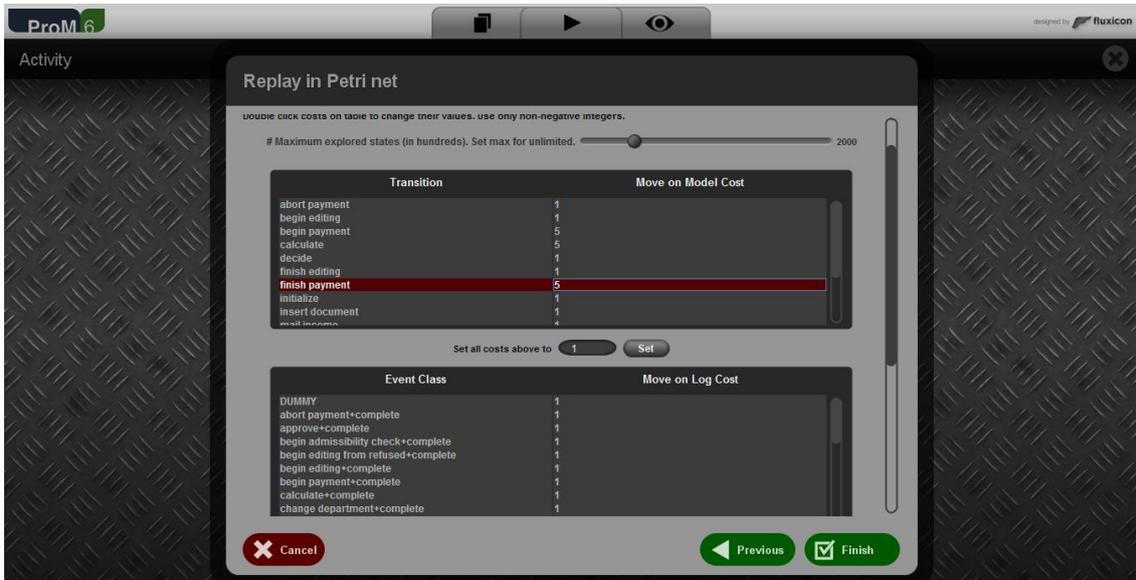


Figura 30 – Parametrização do campo “Move on Model Cost”

Ao executar o plug-in, obteve-se as seguintes informações globais:

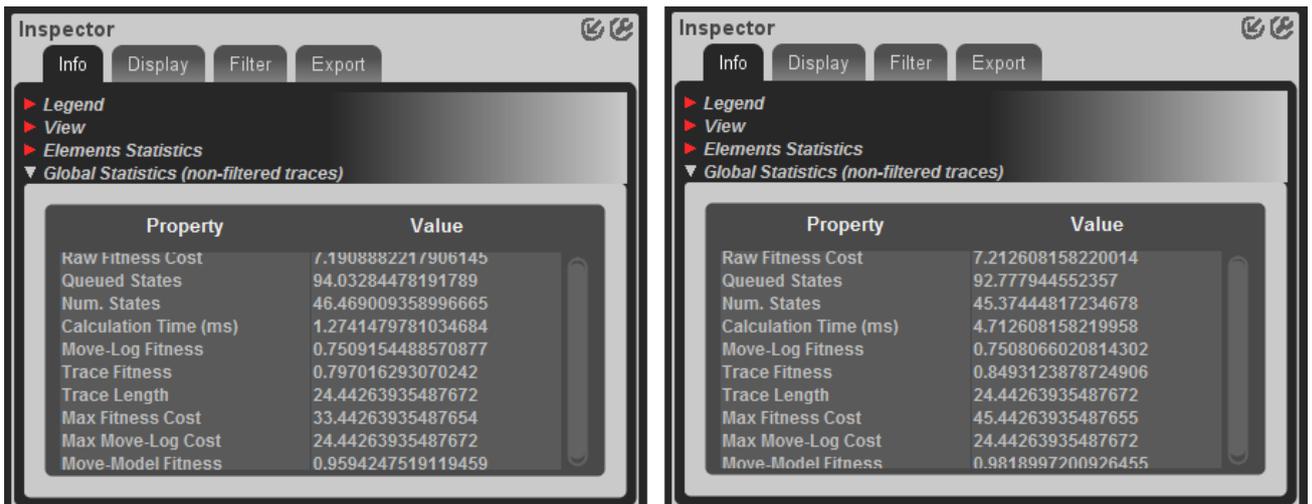


Figura 31 – Comparativo das estatísticas da execução de referência, ao lado esquerdo, com a execução com custo em “Move on Model Cost”, ao lado direito

Conforme pode-se notar na Figura 31, o fitness referente ao log representado pelo “*Trace Fitness*”, aumentou se compararmos ao fitness da execução sem filtro, lado esquerdo. Isto ocorreu pois, como é visto na Figura 22, as atividades “*begin payment*”, “*calculate*”, cuja importância foi considerada maior do que as outras atividades, possuem uma frequência maior de execução do que as demais atividades. Com isso, a penalização pela falta dessas atividades não foi expressiva, acarretando no aumento do *fitness*.

A variável “*Max Fitness Cost*” representa o pior cenário em que ocorre somente movimentos no log ou no modelo, nunca nos dois ao mesmo tempo. Esta variável equivale ao custo total de movimentos ao percorrer o log inteiro sem em nenhum momento realizar uma movimentação em conjunto com o modelo. Na Figura 31, podemos ver que custo máximo aumentou após a adição do peso no filtro “*Move on Model Cost*”, isto quer dizer que se o log executasse somente atividades fora do modelo, o custo total seria maior do que o custo da execução sem filtros.

- **Comparação com alterações no campo “*Move on Log Cost*”**

Já o campo “*Move on Log Cost*” é um filtro que permite ao usuário estabelecer um custo para uma movimentação que ocorre somente no log de teste. Ou seja, o mais em relação ao modelo analisado (AALST, ADRIANSYAH e DONGEN, 2012). Neste filtro, esta movimentação está sujeita a penalização. Neste filtro, com relação ao processo de pagamento de contas, inserimos um custo de 5 para todas as atividades que não fazem parte da execução do modelo, conforme mostrado na Figura 32. Com isto buscamos penalizar a execução de atividades pelo log que não estão presentes no modelo.

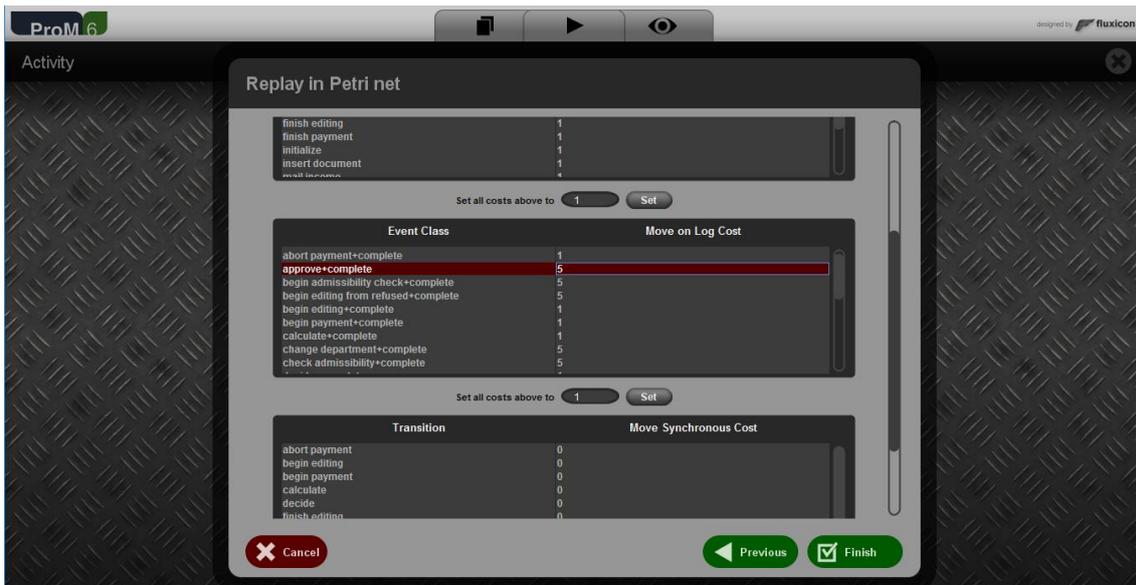


Figura 32 – Parametrização do campo “Move on Log Cost”

Ao executar o plug-in, obteve-se as seguintes informações globais:

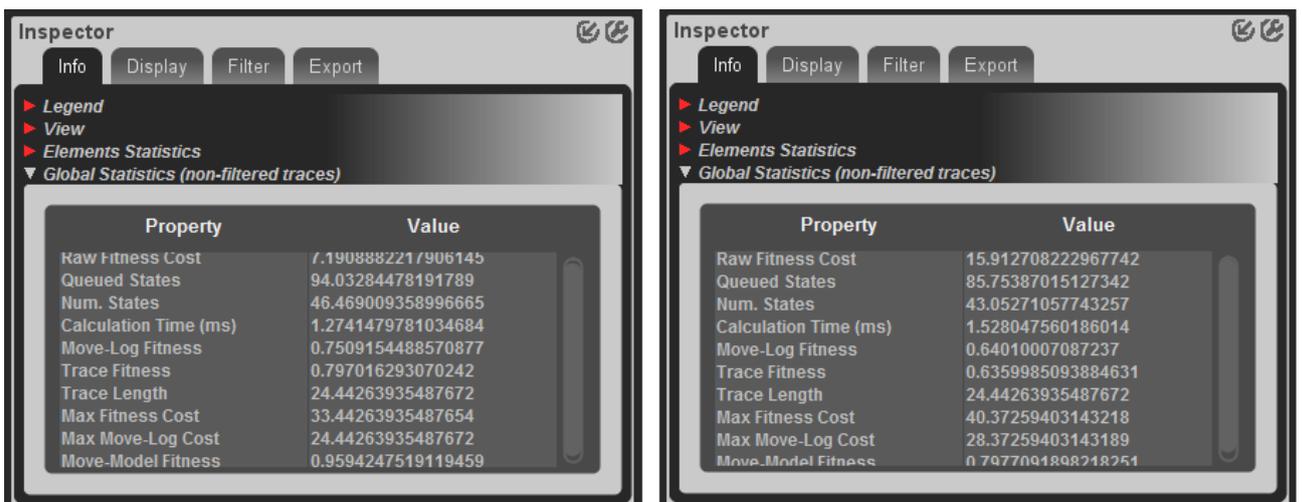


Figura 33 – Comparativo das estatísticas da execução de referência, ao lado esquerdo, com a execução com custo em “Move on Log Cost”, ao lado direito

Conforme pode-se notar na Figura 33, o *fitness* referente ao log representado pelo “Trace Fitness”, diminuiu se compararmos ao *fitness* da execução sem filtro, lado esquerdo. Com isso e a partir de AALST, ADRIANSYAH e DONGEN (2012, p. 9), podemos inferir que o log está executando uma quantidade expressiva de atividades fora do modelo. Sendo assim, a penalização pela execução dessas atividades pesou expressivamente no cálculo do *fitness*.

A variável “*Max Fitness Cost*” representa o pior cenário em que ocorre somente movimentos no log ou no modelo, nunca nos dois ao mesmo tempo. Esta variável equivale ao custo total de movimentos ao percorrer o log inteiro sem em nenhum momento realizar uma movimentação em conjunto com o modelo. Na Figura 33, podemos ver que custo máximo aumentou após a adição do peso no filtro “*Move on Model Cost*”, isto quer dizer que se o log executasse somente atividades fora do modelo, o custo total seria maior do que o custo da execução sem filtros.

## 5 Conclusão

Este trabalho apresentou e comparou técnicas de mineração de processos com a finalidade de aprimorar a qualidade e o tempo da análise dos processos e controles executadas pelo auditor de processos trazendo resultados precisos a partir de uma análise relativamente simples com recursos gráficos. Com o auxílio das ferramentas de mineração de processo Disco e ProM, foi mostrado como manipular, por meio de visões gráficas os logs de eventos provenientes de ERPs, a fim de descobrir modelos de processos e compará-los com o log da real execução destes processos. Com isso, foi possível gerar uma rede Petri com várias informações sobre como um processo está sendo executado com relação ao seu modelo, tais como: fitness entre o log e o modelo, quantificação e evidência dos caminhos alternativos ao modelo, pontos de sobrecarga de recursos e análise do desempenho individual das atividades modeladas.

Devido à escassez de material de referência especificamente com relação ao plug-in de verificação de conformidade “*Replay a Log on Petri Net for Conformance Analysis*”, seu poder de análise não foi totalmente explorado e utilizado. Sendo assim, algumas informações estatísticas não foram discutidas. Sendo que elas certamente permitem alcançar conclusões sobre o log em questão que não são notadas somente com a visão gráfica da rede de Petri resultante.

Como trabalho futuro, é interessante aprofundar um estudo sobre os passos da fase de seleção dos dados a serem utilizados na elaboração do modelo-referência de um processo. Para assim, evitar criar um modelo com atividades e/ou caminhos que possam representar erros na execução deste processo. Desta forma não haverá impactos nos resultados das análises de conformidade. Esta preocupação ocorre, pois, ao utilizar a técnica de replay tendo como input um log teste que possui essas mesmas atividades e/ou caminhos errôneos, não seria possível alarmar tais erros, uma vez que os mesmos fazem parte do modelo de referência.

Outros pontos interessantes a serem explorados são as outras 3 métricas de conformidade também abordadas no plug-in “*Replay a Log on Petri Net for Conformance Analysis*”, são elas: “*simplicity*”, “*precision*” e “*generalization*”. Pois a partir delas é possível identificar, por exemplo, quais são as atividades e/ou caminhos exatos do log

que ocasionam algum tipo de problema. Com isso, correlacionando com o presente trabalho, seria possível, por exemplo, explicar mais a fundo o porquê da variação do *fitness* ao aplicarmos diferentes pesos aos filtros que influenciam na geração da rede de Petri enriquecida com informações da técnica de replay.

## Referências Bibliográficas

ACCORSI, R. & STOCKER, T., 2012. On the Exploitation of Process Mining for Security Audits: The Conformance Checking Case. Germany.

ATTIE, William. Auditoria. 2. ed. São Paulo: Atlas, 2007.

AALST, W. M. P. v. d., 2011. Process Mining. Discovery, Conformance and Enhancement of Business Process. Berlin: Springer.

AALST, W. M. P. v. d., ADRIANSYAH A. & DONGEN, B. v. d., 2012. Replaying History on Process Models for Conformance Checking and Performance Analysis. Eindhoven: TU/E.

AALST, W. M. P. v. d & VERBEEK, H. M. W., 2006. Business process mining: An industrial application. Eindhoven: TU/E.

CUNHA, P. R., e HEIN, N., 2005. Procedimentos de auditoria utilizados pelas empresas de auditoria independente estabelecidas em Santa Catarina. Rio de Janeiro.

DIAS, S. V. S. Auditoria de processos organizacionais. 2. ed. Rio de Janeiro: atlas editora, 2006. v. 3000. 146p.

DUMAS, M., La Rosa, M., Mendling, J. & A. Reijers, H., 2013. Fundamentals of Business Process Management. Berlin: Springer Heidelberg.

FRANCO, Hilário, MARRA, Ernesto. Auditoria contábil. 3. ed. São Paulo: Atlas, 2000.

GOMES, Marcelo Alcides Carvalho. Uma Contribuição à prevenção de fraudes contra as empresas. Tese de Doutorado em Contabilidade. FEA/USP, São Paulo, 2000.

GONÇALVES, José Ernesto Lima. Reengenharia das empresas: passando a limpo. São Paulo: Atlas, 2005.

HAN, J. & KAMBER, M., 2000. Data Mining Concepts and Technics. Burnaby: Morgan Kaufmann Publishers. Eindhoven: TU/E.

JANS, M., ALLES, M. & VASARHELYI M., 2012. The case for process mining in auditing: Sources of value added and areas of application. Belgium & USA.

LOPES, G. N., 2015. Verificação e classificação de não conformidades de processos de negócios na indústria farmacêutica através de técnicas de mineração de processos, Rio de Janeiro: UNIRIO.

NETO, Abílio Bueno; SOLONCA, Davi., 2007. Auditoria de sistemas informatizados. Palhoça: UnisulVirtual.

OLIVEIRA, F. N., 1989. Técnicas de amostragem utilizadas pelos serviços de auditoria interna de empresas no Brasil – Um estudo de casos. Rio de Janeiro: Fundação Getúlio Vargas.

ROZINAT, A., 2010. Process Mining Conformance and extention. Eindhoven: TU/E.

ROZINAT, A. & AALST, W. M. v. d., 2008. Conformance checking of processes based on monitoring real behavior. pp. 64-95. Eindhoven: TU/E.

SANTOS, R. C. d., 2014. Um modelo de apoio à tomada de decisão para análise de similaridade e unificação de processos de negócio, Rio de Janeiro: Copper - UFRJ.

SCHEER, A. Agility & execution driven by aris Bussiness Process. Management. Rio de Janeiro, Anais, 2006.

SCOTT, Richard. et aliL Auditin &: A Systems Approach. Reston Publishing Company, Inc. A Prentice-Hall Company, Reston. Virgínia, 1982.