

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

ELISEU RAMOS

**CONTRIBUIÇÃO PARA UMA ABORDAGEM METODOLÓGICA
ESTRUTURADA NA ANÁLISE DE PROCESSOS DE NEGÓCIOS**

CURITIBA

2009

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

Ramos, Eliseu
R175c Contribuição para uma abordagem metodológica estruturada na análise de
2009 processos de negócios / Eliseu Ramos ; orientador, Eduardo Portela ; co-
orientador, Marco A. Buseti de Paula. -- 2009.
123 f. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná,
Curitiba, 2009
Bibliografia: f. 119-123

1. Negócios. 2. Fluxo de trabalho. I. Santos, Eduardo Alves Portela. II. Paula,
Marco Antonio Buseti de. III. Pontifícia Universidade Católica do Paraná.
Programa de Pós-Graduação em Engenharia de Produção e Sistemas.
IV. Título.

CDD 20. ed. – 670

ELISEU RAMOS

**CONTRIBUIÇÃO PARA UMA ABORDAGEM METODOLÓGICA
ESTRUTURADA NA ANÁLISE DE PROCESSOS DE NEGÓCIOS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção e Sistemas.

Área de Concentração: Integração, Avaliação e Automação de Sistemas

Orientador: Prof. Dr. Eduardo Portela
Co-Orientador: Prof. Dr. Marco A. Busetti de Paula

**CURITIBA
2009**

ELISEU RAMOS

**CONTRIBUIÇÃO PARA UMA ABORDAGEM METODOLÓGICA
ESTRUTURADA NA ANÁLISE DE PROCESSOS DE NEGÓCIOS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção e Sistemas.

COMISSÃO ORGANIZADORA

Prof. Dr. Eduardo Portela
Orientador

Prof. Dr. Marco A. Buseti de Paula
Co-Orientador

Prof. Dr. Eduardo Rocha Loures
Membro Interno

Prof. Dr. Júlio César Nievola
Membro Externo

Curitiba _____ de _____ de 2009.

“O *que* eu ouço, esqueço.
O que eu vejo, lembro.
O que eu faço, aprendo.”

Confúcio

AGRADECIMENTOS

Agradeço aos meus pais e irmãos por todo o apoio recebido, em especial ao meu irmão Jaime, pela idéia inicial e incentivo durante a realização do mestrado.

Ao professor Eduardo Portela, meu orientador, pelas idéias discutidas, orientações precisas e pelo valioso apoio, sem o qual esta pesquisa não se concretizaria.

Aos professores Marco Buseti e Eduardo Loures, também orientadores, pelos conselhos e pela disponibilidade presente em todos os momentos.

A Daniela e demais colaboradores da secretaria do Programa de Pós-Graduação em Engenharia de Produção e Sistemas, pelas orientações prestadas quanto aos trâmites administrativos desse trabalho, e pela boa vontade no atendimento sempre que necessário.

Aos meus amigos do mestrado, com quem pude dividir minhas ansiedades e obter apoio nos momentos críticos.

Aos demais amigos e familiares, principalmente meu filho Eduardo, que compreendeu a minha ausência especialmente no período final deste trabalho.

Finalmente, agradeço a Deus, por ter me dado disposição, saúde e capacidade, para superar mais um grande desafio.

RESUMO

Uma empresa é formada por um conjunto de processos de negócios, e a compreensão de seu funcionamento, suas características e interações são fundamentais para o seu sucesso. A identificação dos processos efetivamente realizados proporciona obter um maior entendimento da organização, racionalizando o fluxo de informações existente na empresa. Nesse sentido, insere-se a mineração de processos, que através do uso de ferramentas e técnicas específicas, possibilita analisar e aperfeiçoar os processos modelados. Atualmente as empresas utilizam sistemas de gestão empresarial para suportar seus processos de negócios, e muitas vezes esses processos formalizados e implementados, podem não atender todos os cenários possíveis e se tornarem desatualizados. Além disso, não acompanham o dinamismo presente no mundo dos negócios e a maneira como seus usuários os utilizam. Os sistemas de informação em operação, normalmente utilizam para fins de auditoria, um arquivo denominado *log*, com o registro de todas as atividades executadas por seus usuários. Esse arquivo é denominado *log* de eventos, se contiver informações relativas as atividades, data e hora e sobre os recursos envolvidos na sua execução. Um *log* de eventos é o ponto de partida para a mineração de processos de negócios. Esse trabalho se propõe a examinar a partir de uma ampla revisão bibliográfica, as principais questões envolvidas na aplicação da mineração de processos, demonstrar características, dimensões, e desenvolver uma abordagem metodológica estruturada, baseada nas técnicas de verificação de conformidade de processos, e na geração de *log* de eventos sintéticos. Com os resultados encontrados a partir de análise experimental realizada, verificou-se que a abordagem proposta pode ser empregada para auxiliar as organizações, na melhoria e racionalização de seus processos de negócios.

Palavras-chave: Processo de Negócio. Modelagem de Processos. Mineração de Processos. Log de Eventos.

ABSTRACT

A company is formed by a set of business processes, and understanding of its operation, characteristics and interactions are key to its success. The identification of these processes actually done, allow us to get a better understanding of the organization, and help us to rationalize the information flow within the company. By implementing the process mining, which through the use of specific tools and techniques, allows to analyze and improve the established processes. Currently, companies use business management systems to support their business processes. However, these processes often formalized and implemented, may not meet all possible scenarios and become outdated, nor they follow the dynamism present in the business world and how their users work with them. The currently information systems in use, usually maintain for audit purposes, a log file, that has a record of all activities performed by its users. This file is called the event log that contains information on activities, date and time and the resources used for that specific activity or activities. An event log is the starting point for the business process mining. The contribution presented in this paper, proposes to study starting from a broad bibliographical review, the main issues involved in implementing of the process mining. It will demonstrate features, dimensions, and develop a structured methodological approach, based on process compliance techniques, and the creation of synthetic event log. An experimental analysis has been elaborated and with the found results, it was verified that the proposed approach can be used to aid the organizations in the improvement and rationalization of their businesses processes.

Keywords: Business Process, Process Modelling, Process Mining, Event Log.

LISTA DE FIGURAS

Figura 1 - Estrutura da pesquisa	20
Figura 2 - Estrutura da revisão teórica	20
Figura 3 - Elementos básicos de Redes Petri	23
Figura 4 - Representação compacta de uma rede de Petri	24
Figura 5 - Marcação em rede de Petri - transação habilitada	25
Figura 6 - Marcação em rede de Petri - após o disparo da transição	25
Figura 7 - Exemplo de uma rede de Petri colorida	30
Figura 8 - Representação formal de um processo	33
Figura 11 - Exemplo de um processo de revenda	38
Figura 12 - Exemplo de um processo de compra	38
Figura 14 - Abordagens de um processo de negócio	40
Figura 15 - Ciclo de vida do BPM	42
Figura 16 - Ciclo de vida de um processo de negócio	44
Figura 18 - Interface de sistemas de Administração e Supervisão	47
Figura 21 - Meta modelo do formato MXML	54
Figura 22 - Formato XML de mineração de processos	55
Figura 23- Exemplo genérico de um log MXML visualizado a partir do XMLSpy	56
Figura 24 - Detalhamento XML do exemplo de log de eventos da figura 23	57
Figura 27 - Mineração de processo – técnica de conformidade	63
Figura 28 - Modelo M1 – Simplificação do processamento de uma reclamação de seguro	64
Figura 29 - Três logs de eventos para o processo descrito na Figura 28	66
Figura 30 - Modelo de processo M2 - Flower	67
Figura 31 - Modelo de processo M3	67
Figura 32 - Repetição de L2 e trilha $i=1$ no modelo M1	70
Figura 33 - Repetição de L2 para trilha $i=4$ no modelo de processo M1	72
Figura 34 - Diagnóstico nos contadores de fichas	74
Figura 35 - Modelo M4 - 100% compatível com o log de evento L2	75
Figura 36 - Modelo M5 que permite o desvio da atividade G	76
Figura 37 - Relações globais “Follow” derivadas para o modelo M5 e Log L2	79
Figura 38 - Diferenças entre atividades sucessoras e predecessoras	81
Figura 39 - Modelo contendo construções que podem “inflar” sua estrutura	82
Figura 40- Procedimentos metodológicos	89
Figura 41 - Ciclo de vida da abordagem metodológica estruturada proposta	91
Figura 42 - Visão Geral da ferramenta ProM	93
Figura 43 - Visão geral da área de trabalho	95
Figura 44 - Processo de criação de logs de eventos artificiais	97
Figura 45 - Agregação de logs de eventos	98
Figura 46 - <i>Framework</i> proposto	99
Figura 47 - Estrutura organizacional do processo formal	101
Figura 48 - Modelo de processo formal experimental	102
Figura 49 - Recorte do <i>log</i> de eventos LogTrataMultasReal.mxml	103
Figura 51 - <i>Fitness</i> entre modelo formal e real	105
Figura 52 - <i>Appropriateness</i> comportamental avançado	106
Figura 53- <i>Appropriateness</i> comportamental avançado	106
Figura 55 - Representação básica em RdPC do modelo real	108
Figura 56- Modelo de processo real melhorado em RdPC	110

Figura 57 - Ferramenta ProMImport.....	111
Figura 58 - Modelo de processo modificado obtido a partir da mineração do log ...	112
Figura 59 - <i>Fitness</i> entre modelo formal e modificado	112
Figura 60 - <i>Appropriateness</i> comportamental avançado.....	113
Figura 61 - <i>Appropriateness</i> estrutural avançada.....	113
Figura 62 – Nova configuração da estrutura organizacional do processo modificado	114

LISTA DE ABREVIATURAS E SIGLAS

B2B	– Business to Business
BPM	– Business Process Management
BPMI	– Business Process Management Initiative
BPMN	– Business Process Management Notation
CRM	– Customer Relationship Management
DTD	– Document Type Definition
EAI	– Enterprise Application Integration
EPC	– Event-Driven Process Chain
EPML	– EPC Markup Language
ERP	– Enterprise Resource Planning
ISO	– International Organization for Standardization
ML	– Modelling Language
MRP	– Material Requirement Planning
MXML	– Mining Extensible Markup Language
PNML	– Petri Net Markup Language
RdP	– Rede de Petri
RdPC	– Rede de Petri Colorida
SCM	– Supply Chain Management
UML	– Unified Modeling Language
WfMC	– Workflow Management Coalition
WfMS	– Workflow Management System
XML	– Extensible Markup Language

SUMÁRIO

1.	INTRODUÇÃO	13
1.1.	PROBLEMA DE PESQUISA.....	15
1.2.	JUSTIFICATIVA.....	15
1.3.	OBJETIVOS.....	17
1.3.1.	Objetivo geral	17
1.3.2.	Objetivos específicos	17
1.4.	METODOLOGIA	17
1.5.	Estrutura da pesquisa	20
2.	REVISÃO TEÓRICA.....	21
2.1.	Redes de Petri - RdP	22
2.1.1.	Clássicas	22
2.1.2.	Redes de Petri de alto nível - coloridas	27
2.2.	Processos	32
2.3.	<i>Workflow</i>	34
2.4.	Processos de negócios	36
2.5.	WfMC – Workflow Management Coalition.....	45
2.6.	<i>Log</i> de eventos	49
2.7.	Mineração de processos	59
2.8.	Técnica de conformidade.....	62
2.8.1.	Características	63
2.8.2.	Mapeamento	65
2.8.3.	As duas dimensões de conformidade	66
2.8.4.	Métrica <i>Fitness</i>	69
2.8.5.	Métrica <i>Appropriateness</i>	74
2.8.5.1.	<i>Appropriateness</i> comportamental.....	75
2.8.5.2.	<i>Appropriateness</i> comportamental simples.....	77
2.8.5.3.	<i>Appropriateness</i> comportamental avançado	80
2.8.5.4.	<i>Appropriateness</i> estrutural	82
2.8.5.5.	<i>Appropriateness</i> estrutural simples	84
2.8.5.6.	<i>Appropriateness</i> estrutural avançada	85
2.8.6.	Combinando <i>Fitness</i> e <i>Appropriateness</i>	86
2.8.7.	Considerações finais	87
3.	PROCEDIMENTOS METODOLÓGICOS	89
3.1.	Delimitação	89
3.2.	Ferramenta ProM.....	92
3.3.	Ferramenta CPN TOOLS.....	94
3.4.	Geração de <i>logs</i> artificiais.....	96
3.5.	Detalhamento da abordagem metodológica estruturada	98
3.6.	O Experimento	101
3.6.1.	Carregamento do <i>log</i> e modelo no ProM	104
3.6.2.	Verificação da conformidade	104
3.6.3.	Análise do processo encontrado	107
3.6.4.	Modificações no processo	108
3.6.5.	Geração e Agregação de <i>Logs</i>	111
3.6.6.	Análise do processo modificado	112
3.7.	Considerações sobre o Experimento.....	114

4.	CONCLUSÃO.....	116
4.1.	Análise dos objetivos propostos.....	116
4.2.	Limitações da pesquisa e sugestão de trabalhos futuros.....	117
4.3.	Considerações finais.....	118
	REFERÊNCIAS.....	119

1. INTRODUÇÃO

Para Jochem e Mertins (2005), a economia global altamente competitiva nos dias de hoje, voltada para a demanda por produtos manufaturados de alta qualidade, com baixo custo e ciclos de vida cada vez menores, tem forçado as empresas a empregarem novas estratégias de projetos, produção e gerenciamento de produto.

Thom *et al* (2006), cita que nos últimos anos as empresas para se manterem competitivas no mercado, tem explorado as técnicas de gestão por processos, principalmente pelas exigências cada vez mais freqüentes de certificação à norma ISO 9001:2000, a qual define que uma empresa deve ser retratada por seus processos de negócio principais e não pelo seu organograma.

Davenport (1993) define que o conceito de processos de negócio, é um conjunto de tarefas relacionadas logicamente, executadas a fim de obter um resultado de negócio definido para um cliente particular ou para o mercado.

Hammer e Champy (1993) apresentam essencialmente a mesma definição, só que neste caso, focalizam o cliente no centro dos processos de negócio: uma coleção de atividades onde por meio de uma ou mais entradas, são criadas saídas que tem valor para o consumidor.

Para Smith e Fingar (2003), a evolução da gestão de processos de negócios que vem ocorrendo, pode ser classificada por três grandes movimentos denominados de ondas. A primeira onda consistiu da modelagem de processos e simulação, baseada no Taylorismo, o foco era a construção e análise de processos, onde predominava a grande divisão do trabalho, o total controle gerencial do ambiente de trabalho, e uma contabilidade dos custos baseada em um sistemático estudo de tempos e movimentos.

A segunda onda caracterizou-se pelo início da reengenharia dos processos, direcionamento para a integração de aplicações empresariais – EAI, e implementação de sistemas específicos baseados em MRP, ERP e soluções customizadas. Observou-se que os processos eram monitorados ao mesmo tempo em que existiam modelos descrevendo estes processos.

Na terceira onda ocorreu integração em larga escala, onde o pensamento dominante foi a facilitação dos processos de mudanças, motivados pelas aquisições

e fusões, formação de novos mercados e produtos, e mudanças na Legislação. É a busca do conhecimento através da extração dos processos de negócios a partir dos sistemas em execução.

Uma empresa é formada por um conjunto de processos de negócios, e a compreensão de seu funcionamento, suas características e interações são fundamentais para o seu sucesso. Nesse sentido, insere-se a mineração de processos, que através do uso de ferramentas e técnicas específicas, possibilita analisar e aperfeiçoar os processos modelados.

Para Rosinat e Aalst (2005), as técnicas de mineração de processos têm provado serem ferramentas valiosas na obtenção do conhecimento de como os processos são organizados dentro das organizações.

Enquanto os modelos de referência de processos descrevem como um sistema deve ser utilizado, arquivos denominados *log* de transação, gerados pelos sistemas transacionais, armazenam todos os eventos realizados por seus usuários. Assim é possível avaliar se o processo real (representado pelos eventos gravados no *log*) e o modelo de processo (usado para configurar o sistema) estão alinhados.

A partir da leitura do *log* de eventos, as técnicas de mineração de processos podem ser usadas de duas diferentes formas: no processo de descoberta e na verificação de conformidade.

O processo de descoberta pode ser usado para construir automaticamente um modelo de processo, refletindo o comportamento que foi observado e gravado no *log* de eventos.

A verificação de conformidade pode ser usada para comparar o comportamento gravado, com alguns modelos de processo já existentes, detectando possíveis desvios.

Ambas as maneiras servem de entrada para o projeto e melhoria de processos de negócios, isto é, a verificação de conformidade pode ser usada para encontrar problemas em processos existentes; e o processo de descoberta pode ser usado como ponto de partida para análise de processos e configuração de sistema.

Dentre as pesquisas existentes sobre mineração de processos, destacam-se os trabalhos propostos por Wil van der Aalst, Ana Karla Alves de Medeiros, Anne Rozinat, Christian Günther e outros.

1.1. PROBLEMA DE PESQUISA

Colocado em um grau de abstração elevado, o principal problema observado ao propor esta pesquisa está relacionado a uma pergunta bastante recorrente nas organizações:

“Os processos de negócios formalmente estabelecidos são os processos que estão sendo executados pela organização”

Nos trabalhos realizados por Wil van der Aalst, Ana Karla Alves de Medeiros, Anne Rozinat, Christian Günther e outros pesquisadores da Universidade de Tecnologia de Eindhoven, Holanda, constata-se que a técnica de mineração de processos auxilia nessa resposta, ao fornecer a partir da leitura dos arquivos *log* de eventos gerados pelos sistemas transacionais, modelos de processos que estão sendo executados.

Logs de eventos reais gerados a partir de sistemas transacionais frequentemente são incompletos e contém ruídos, que impactam diretamente na qualidade dos processos minerados.

Desse modo, esta pesquisa busca propor uma contribuição para as organizações que necessitam aprimorar seus processos de negócios, que consiste em utilizar um ambiente de simulação baseado em redes de Petri coloridas e na técnica de mineração de processos, pelo método da conformidade, avaliando se com as alterações propostas num modelo de processo e no seu *log* de eventos,

“É possível melhorar a conformidade entre os processos de negócios formalizados e os processos efetivamente realizados”

1.2. JUSTIFICATIVA

Observa-se que as empresas têm uma necessidade de sistematizar suas práticas de trabalho e em automatizar seus processos, tamanha a velocidade com que as mudanças nos negócios ocorrem.

Nesse sentido, a atividade de revisão, correção e introdução de aprimoramentos nos processos devem ser praticadas regularmente, pois resulta em maior agilidade e competitividade aos produtos e serviços.

Além disso, os sistemas de auditoria de processos de negócio devem buscar melhoria face às exigências da legislação societária americana (Sarbanes-Oxley); pela importância crescente da governança corporativa e pela certificação à norma ISO 9001:2000.

Os processos de negócio são componentes-chave para organizar as atividades e para melhorar o nível de conhecimento do inter-relacionamento entre elas. As atividades dos processos de negócios podem ser realizadas manualmente ou com o auxílio de sistemas informatizados.

A identificação dos processos efetivamente realizados pela empresa proporciona obter uma maior compreensão da organização, fornecendo subsídios para racionalizar o fluxo de informações existente na mesma.

Uma empresa pode atingir seus objetivos empresariais de maneira eficiente e efetiva quando seus recursos (humanos, equipamentos, sistemas de informação) estão operando em harmonia. Os processos de negócios desempenham uma importante tarefa de facilitar essa efetiva colaboração, reduzindo o *gap* entre os aspectos organizacionais e a tecnologia.

Para Vernadat (1996), as organizações devem se adaptar rapidamente as mudanças de cenário, causadas pelas alterações constantes nas conformações geopolíticas, nas condições de mercado e na tecnologia. Na disputa pelo espaço de mercado global, a maioria das empresas deve reestruturar seus processos (administrativos, técnicos ou processos de suporte), melhorar a administração destes processos e encontrar uma forma de torná-los mais integrados.

Para a integração e controle de processos de negócios, é necessária uma formalização através de técnicas e ferramentas apropriadas, que abrange não apenas os objetos que eles utilizam ou processam, mas também as informações acessadas ou geradas, recursos demandados, responsabilidades e atribuições aplicáveis.

Aaslt *et al* (2003) afirmam que a criação e projetos de processos de negócios, são atividades complexas e demoradas e normalmente exibem distorções, entre o processo de negócio executado e o processo formalmente estabelecido pelas organizações. É nesse contexto que se insere a mineração de processos de negócios.

1.3. OBJETIVOS

1.3.1. Objetivo geral

Esta dissertação tem como objetivo geral propor uma abordagem metodológica estruturada de análise de processos de negócios, através do emprego da técnica de conformidade e do ambiente de simulação de redes de Petri colorida.

1.3.2 Objetivos específicos

Os objetivos específicos para este trabalho compreendem:

- a) Identificar no referencial teórico, definições e conceitos sobre processos, gerenciamento de processos de negócios, *log* de eventos, mineração de processos, análise de conformidade em processos de negócios, redes de Petri clássica e colorida;
- b) Propor modificações no processo de negócio e no seu *log* de eventos e avaliar através da abordagem metodológica estruturada se houve melhoria na conformidade entre o processo formal e o processo real;
- c) Analisar os resultados obtidos e elaborar conclusões.

1.4. METODOLOGIA

Pesquisa é todo processo sistemático de formação do conhecimento que tem por princípio gerar novos conhecimentos e/ou corroborar ou refutar algum conhecimento pré-existente.

Consiste no processo de aprendizagem tanto do pesquisador que a desenvolve quanto da sociedade na qual a pesquisa está inserida. Ander-Egg (1978) cita que pesquisa é um procedimento formal, ordenado, através do qual é possível conhecer a realidade e descobrir novos fatos ou dados, em qualquer área do conhecimento.

De acordo com Marconi e Lakatos (2007), a obtenção de dados por parte do pesquisador pode dar-se por meio de dois processos: documentação direta ou indireta. Na documentação direta, faz-se o levantamento de dados no próprio local onde os fenômenos ocorrem (pode ser através de pesquisa de campo ou de laboratório) e na indireta, utilizam-se fontes de dados coletados por outras pessoas (pode se constituir em material já elaborado).

Santos (2002) define como pesquisa exploratória aquela efetuada através de consulta a livros, artigos, teses, dissertações, publicações científicas e revistas especializadas, com a finalidade de conhecer fatos e fenômenos relacionados com o tema e recuperar as informações disponíveis.

Com um levantamento bibliográfico e documental é possível contextualizar a pesquisa, obter uma maior consistência com os dados expostos, e a partir daí, relacionar características, evidenciar sua relevância, demonstrar resultados esperados e dificuldades encontradas pela técnica de conformidade, na mineração de processos, a partir da leitura de arquivos de *log* de eventos.

Segundo Silva e Menezes (2001), esta pesquisa em relação à abordagem do problema foi classificada como qualitativa, ao considerar que há uma relação dinâmica entre o mundo real e o teórico, e também quantitativa, por outro lado, ao levar em conta as métricas adotadas na conformidade de processos, o que significa traduzir em números, as opiniões e as informações para classificá-las e analisá-las.

Quanto aos procedimentos técnicos adotados, segundo Gil (1991, p.54), trata-se de uma pesquisa experimental, pois “se determina um objeto de estudo, selecionam-se as variáveis que seriam capazes de influenciá-lo, definem-se as formas de controle e de observação dos efeitos que a variável produz no objeto”.

Metodologia e método são conceitos distintos. Para Marconi & Lakatos (2007, p.83), método “é o conjunto das atividades sistemáticas e racionais que, com maior segurança e economia, permite alcançar o objetivo, traçando o caminho a ser seguido”.

Já metodologia é a explicação minuciosa, detalhada, rigorosa e exata de toda ação desenvolvida no método (caminho) do trabalho de pesquisa. É a explicação do tipo de pesquisa, do instrumental utilizado (questionário, entrevista, etc), do tempo previsto, da equipe de pesquisadores e da divisão do trabalho, das formas de tabulação e tratamento dos dados, enfim, de tudo aquilo que se utilizou no trabalho de pesquisa.

Estorilio (2003), de forma semelhante descreve metodologia como sendo conceitos que abrangem um conjunto de conhecimentos, que proporcionam a criação de embasamento teórico e fornecem parâmetros para implantação de métodos, ferramentas, sistemas e outras soluções. Por sua vez, método é descrito como um procedimento composto de conhecimentos menos abrangentes e que normalmente são estruturados em passos relacionados com algo específico, visando alcançar um determinado objetivo.

Dessa forma, método pode ser entendido como um conjunto de procedimentos a serem seguidos para que se atinja um determinado resultado e metodologia se refere à lógica ou parte da lógica que estuda os métodos, ou conjunto de procedimentos metódicos de uma ou mais ciência.

Ainda segundo Marconi e Lakatos (2007), o método científico pode ser classificado quanto à classe de métodos mais amplos em:

- a) Indutivo – trata da busca da solução, que parte de constatações particulares e visa chegar a conclusões genéricas;
- b) Dedutivo – parte-se das teorias, leis e postulados gerais, que em muitas vezes predizem a ocorrência de fenômenos particulares;
- c) Dialético – a busca da solução é feita no mundo dos fenômenos por meio da ação recíproca, da contradição inerente ao fenômeno e da mudança dialética que ocorre na sociedade e na natureza.
- d) Hipotético-dedutivo – trata da análise das percepções de pontos falhos no conhecimento, a respeito do qual se estabelecem hipóteses, e pelo processo de inferência dedutiva, testam-se as possibilidades de ocorrência de fenômenos citados pela hipótese.

Quanto ao método utilizado nesta pesquisa, aplicou-se o método hipotético-dedutivo, conforme Popper (1975), que parte da definição de um problema para o estabelecimento de conjecturas ou proposições passíveis de teste. Esses testes são tentativas de refutação, por meio de observação e experimentação.

Semelhante definição para este método é dada por Bunge (1974 *apud* Marconi e Lakatos, 2007), que divide a abordagem em cinco etapas:

- a) Definição: que consiste no reconhecimento dos fatos e observação de lacunas no conhecimento existente e colocação do problema;
- b) Construção: de um modelo teórico a partir de suposições e hipóteses;

- c) Dedução: das conseqüências particulares, com elaboração de predições tendo por base o modelo teórico e dados empíricos;
- d) Teste: das hipóteses, através de verificação dos experimentos realizados, análise e interpretação dos resultados obtidos
- e) Adição ou introdução: de comparação das conclusões com as predições estabelecidas para o modelo hipotético.

Já para Vergara (2009), o método hipotético-dedutivo dá grande importância à técnica e à quantificação, e a causalidade é a base científica para justificar as hipóteses formuladas. As soluções para um problema devem ser testadas através da experimentação e observação, confirmando ou rejeitando as hipóteses estabelecidas.

1.5. Estrutura da pesquisa

Considerando a proposta de Marconi e Lakatos (2007) para a elaboração de pesquisa científica, este trabalho é estruturado em capítulos, conforme mostrado na figura 1.

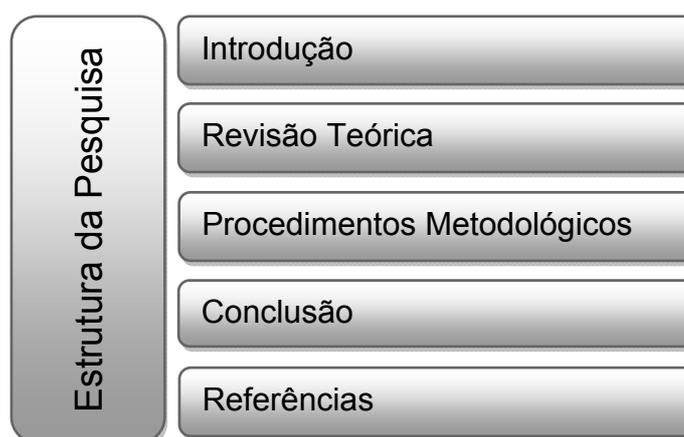


Figura 1 - Estrutura da pesquisa

2. REVISÃO TEÓRICA

Na revisão teórica é apresentado um embasamento conceitual que permitirá a compreensão das principais áreas de estudo desta pesquisa. Este referencial teórico é importante, uma vez que auxilia o esclarecimento e justifica o alcance dos objetivos citados anteriormente.

São abordados inicialmente conceitos e o formalismo das redes de Petri clássicas e coloridas, fundamental para compreensão das técnicas de mineração e simulação aplicadas nesta pesquisa.

A seguir são apresentadas definições de diferentes autores, relacionados a processos, *workflow*, processos de negócio, gerenciamento de processos de negócio, WfMC- Workflow Management Coalition e *log* de eventos.

Para finalizar, é abordada a mineração de processos, suas características, perspectivas, e é apresentada a técnica de conformidade e suas dimensões de análise. A organização deste capítulo é demonstrada na figura 2.



Figura 2 - Estrutura da revisão teórica

2.1. Redes de Petri - RdP

Neste subcapítulo são apresentadas as redes de Petri clássicas, que segundo Aasit e Hee (2002), constituem um excelente ponto de partida para o formalismo e a validação de um sistema de gerenciamento de processos de negócios. Os modelos elaborados com Redes de Petri são considerados modelos executáveis, uma vez que permitem simular a seqüência da execução das atividades, de acordo com alguns critérios pré-estabelecidos

2.1.1. Clássicas

A rede de Petri – RdP segundo Murata (1989), é uma ferramenta de modelagem gráfica e matemática, que surgiu em 1962, na apresentação da tese de doutorado *Kommunikatin mit Automaten* defendida por Carl Adam Petri na Faculdade de Matemática e Física da Universidade de Darmstadt, Alemanha.

Como ferramenta gráfica pode ser usada como ajuda visual de comunicação, similarmente ao fluxograma, diagrama de blocos ou redes, na análise de modelos e estudo de sistemas, com aplicação em diferentes áreas tais como sistemas administrativos, de manufatura, desenvolvimento de softwares, etc.

Ela proporciona também um ambiente uniforme para a análise formal, modelagem e simulação de sistemas a eventos discretos, obtendo uma visualização simultânea da sua estrutura e comportamento.

Ela é formada de estruturas dinâmicas com dois tipos de elementos: a transição, elemento ativo correspondente a alguma ação realizada dentro da rede representada, e o lugar, elemento passivo e relacionado a alguma variável de estado da rede.

A execução das ações no âmbito da rede está relacionada a pré-condições ou condições das variáveis de estado do sistema, isto é, existe associação entre transições e lugares que permitem a execução de determinada ação. De igual maneira, as informações de alguns lugares são modificadas, após a conclusão de determinada ação (pós-condição).

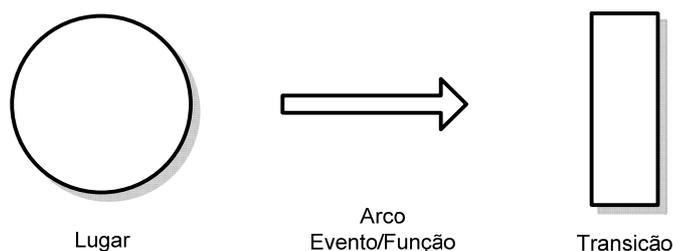


Figura 3 - Elementos básicos de Redes Petri

Fonte: Desenvolvido pelo autor.

Os lugares são representados graficamente por círculos e as transições, por traços ou barras (Figura 3) que são vértices do grafo associado às redes de Petri e interligam-se por arcos direcionados.

Arcos podem ligar lugares às transições e transições a lugares, representando as relações entre condições verdadeiras que viabilizam em algum momento a execução de ações dentro da rede, ou representando as relações entre ações e condições que se tornam verdadeiras com a execução das ações. As transições são habilitadas na medida em que são colocadas através dos arcos, marcas nos lugares de entrada. Esse disparo é controlado pelo número e pela distribuição das marcas nos lugares. Uma transição só dispara se houver marcas suficientes para todos os arcos de entrada. Pode-se associar um peso a cada arco e assim a quantidade de marcações deverá ser igual ao peso do arco.

A definição formal para uma rede de Petri sem marcação é dada por uma quádrupla $RP = (P, T, F, I, O)$, onde:

- $P = \{ p_1, p_2, \dots, p_m \}$ representa o conjunto finito de lugares da rede;
- $T = \{ t_1, t_2, \dots, t_n \}$ representa o conjunto finito de transições da rede;
- $I : (P, T)$ representa o conjunto finito de arcos de entrada em relação às transições;
- $O : (T, P)$ representa o conjunto finito de arcos de saída em relação às transições;
- $m \in \mathbb{N}$ e $n \in \mathbb{N}$

Na representação gráfica da Figura 4 temos um exemplo compacto da rede sem marcação, com lugares de entrada (p_1) e saída (p_2, p_3) conectados através de arcos à transição (t_1).

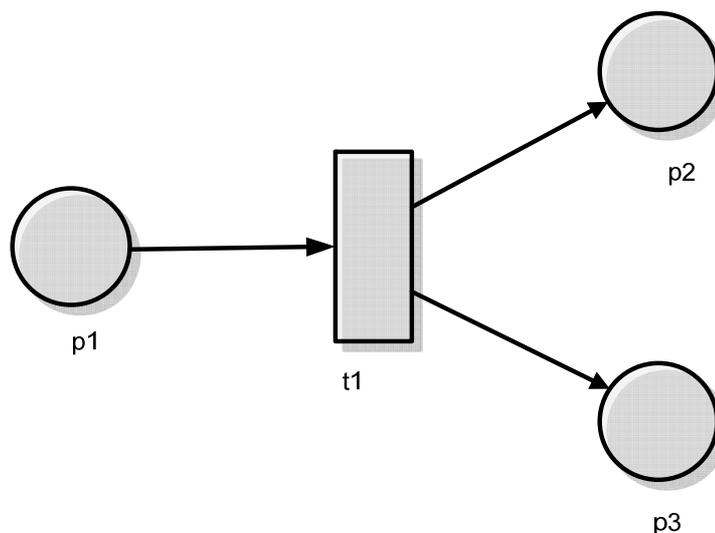


Figura 4 - Representação compacta de uma rede de Petri

Fonte: Desenvolvido pelo autor.

Uma estrutura com marcações em seus lugares de entrada e saída é denominada de Rede de Petri Marcada, cuja definição formal apresentada por Murata (1989) é dada por uma quintupla $RP = (P, T, F, W, M_0)$, onde:

- $P = \{ p_1, p_2, \dots, p_m \}$ representa o conjunto finito de lugares;
- $T = \{ t_1, t_2, \dots, t_n \}$ representa o conjunto finito de transições;
- $F \subseteq (P \times T) \cup (T \times P)$ representa o conjunto finito de arcos ;
- $W: F \rightarrow \{ 1, 2, 3, \dots \}$ representa a função de peso;
- $M_0: P \rightarrow \{ 0, 1, 2, 3, \dots \}$ representa a marcação inicial;
- $P \cap T = \emptyset$ e $P \cup T \neq \emptyset$.

A representação gráfica das marcações é indicada através de pontos pretos no interior dos lugares, onde uma determinada distribuição desses pontos configura um estado da rede. A figura 5 ilustra uma rede onde as marcações habilitam a transição t1, obedecendo as regras de disparo da transição e da quantidade de marcações no arco.

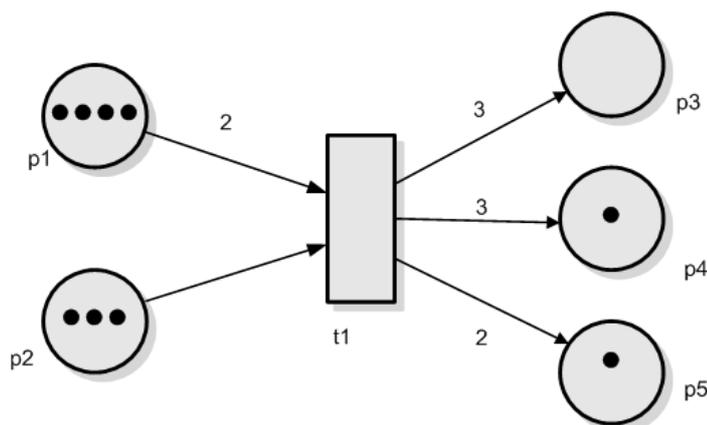


Figura 5 - Marcação em rede de Petri - transação habilitada

Fonte: Desenvolvido pelo autor.

A figura 6 ilustra uma rede de Petri após o disparo da transição \$t_1\$, simulando o comportamento dinâmico do sistema. O disparo da transição habilitada removeu de cada lugar de entrada, o número de marcas igual ao peso do arco que conecta a transição, transferindo para o lugar de saída, um número de marcações igual ao peso do arco de saída.

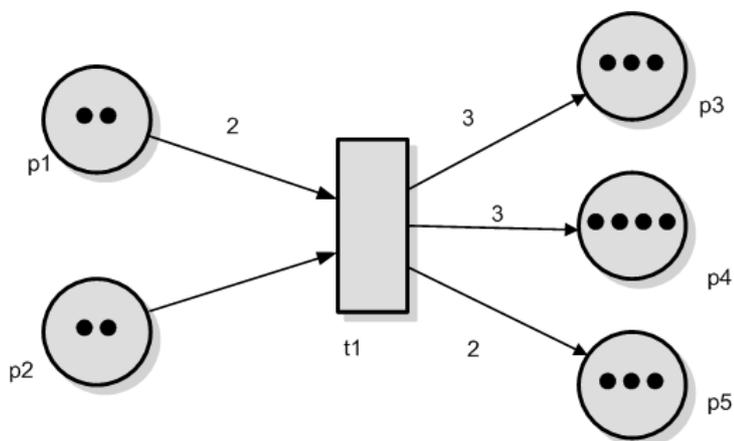


Figura 6 - Marcação em rede de Petri - após o disparo da transição

Fonte: Desenvolvido pelo autor.

As redes de Petri foram amplamente documentadas e suas propriedades conforme são dependentes ou não de marcação inicial, são classificadas em comportamentais ou estruturais. Nas propriedades comportamentais, Murata (1989) define:

- a) Alcançabilidade: indica a possibilidade de atingir um determinado conjunto de marcações após disparos de transições a partir de uma marcação inicial;

- b) Vivacidade: é quando uma rede é livre de impasses (*deadlocks*). Uma transição morta é uma transição que não pode mais ser disparada. Uma rede morta se encontra em impasse total, ou seja, todas as transições encontram-se mortas;
- c) Reversibilidade: quando existe a possibilidade de retorno à marcação inicial ou outra marcação qualquer;
- d) Limitabilidade: uma rede pode ser dita como k-limitada. Em uma rede 5-limitada, nenhum dos lugares pode exceder o número de marcações definida por k, neste caso, cinco;
- e) Persistência: quando em qualquer par de transições habilitadas, o disparo de uma não desabilita o disparo da outra.
- f) Cobertura: quando uma determinada marcação pode ser obtida através de outra marcação;
- g) Distância sincrônica: é uma métrica fortemente relacionada com o grau de dependência mútua entre dois eventos;
- h) Justiça: diz respeito à quantidade de disparos de uma transição em relação à outra.

Quanto às propriedades estruturais, destacam-se como de especial interesse:

- a) Vivacidade estrutural: se existir uma marcação inicial viva para a rede;
- b) Controlabilidade: uma rede é dita completamente controlada quando é qualquer marcação é alcançável pelas outras marcações;
- c) Limitabilidade estrutural: quando a rede possui restrições quanto sua estrutura, para qualquer marcação inicial;
- d) Conservabilidade: sem levar em conta os disparos das transições existentes na rede, o número de marcações não se altera;
- e) Repetibilidade: uma rede é repetitiva se para uma determinada marcação e um conjunto de disparos de transições, houver o disparo ilimitado de todas as transições da rede;
- f) Consistência: uma rede é consistente se for possível voltar ao estado inicial M_0 disparando pelo menos uma vez todas as transições da rede;

2.1.2. Redes de Petri de alto nível - coloridas

A rede de Petri colorida – RdPC é uma linguagem gráfica, e que vem sendo extensivamente utilizada para modelagem e análise de sistemas distribuídos, com elementos de concorrência, comunicação e sincronização.

Com uma semântica formalmente estabelecida, a qual permite que o seu comportamento seja definido sem ambigüidades, ela fornece uma boa plataforma para a modelagem e validação de sistemas de informação orientados a processos, projeto de protocolos de comunicação, análises de *workflow*, sistemas embutidos, e sistemas distribuídos.

De acordo com Jensen (1997), a RdPC combina os fundamentos de notação gráfica e princípios básicos para modelagem concorrencial, comunicação e sincronização da rede de Petri clássica, com as funcionalidades da linguagem de programação Standard ML, princípios para definição de tipos de dados e descrição da manipulação dos dados.

Um modelo RdPC de um sistema é um modelo executável representando os estados do sistema e suas transições (eventos), que permitem ou ocasionam mudanças no mesmo. A linguagem RdPC torna possível organizar um modelo como um conjunto de módulos, com a inserção do conceito de tempo, refletindo o tempo gasto para a execução de eventos no sistema modelado.

Ao simular uma rede RdPC é possível investigar diferentes cenários e explorar cada comportamento do sistema. Para análises de desempenho, é possível instrumentar o modelo com pontos de verificação e coletar os dados relativos às medidas de desempenho.

As redes de Petri coloridas são assim chamadas por permitirem que valores de dados sejam atribuídos às fichas e que as mesmas sejam diferenciadas umas das outras através de cores, em contraste com as fichas das redes de Petri clássicas, que por convenção, são representadas na cor preta.

Comparando as redes RdPC com as clássicas, algumas características são apresentadas:

- a) Cada rede RdPC pode ser transformada em uma rede Petri equivalente e vice-versa;

- b) Uma vez que o poder expressivo dos dois formalismos são os mesmos, não existe ganho teórico ao utilizar redes RdPC, entretanto, na prática, as redes RdPC são muito mais compactas e convenientes na modelagem que as redes clássicas;
- c) O formato de representação gráfica é de fácil compreensão, o que permite uma melhor aceitação pelos analistas de sistemas. O formato se assemelha aos diagramas de fluxo ou à representação física dos sistemas modelados;
- d) O grafismo utilizado para representar os lugares, estados, ações e fluxo de informação das Redes de Petri clássicas também é adotado pelas redes RdPC.

Ainda segundo Jensen (1992), uma rede RdPC é uma 8-tupla. Sua definição formal é adotada para evitar definições ambíguas de sua semântica, bem como para formular definições gerais e provar teoremas. Uma rede RdPC na prática sempre deverá ser descrita como um grafo:

RdPC = $\{\Sigma, P, T, A, N, C, G, E, I\}$ sendo:

- Σ : representa um conjunto finito de cores ou tipos, não nulo, denominado de *colour sets*;
- P: representa um conjunto finito de lugares;
- T: representa um conjunto finito de transições;
- A: representa um conjunto finito de arcos;
- N: $A \rightarrow (P \times T) \cup (T \times P)$ representa a função de nó (*node function*);
- C: $P \rightarrow \Sigma$ representa a função de cor (*colour function*);
- G: representa a função de controle (*guard function*);
- E: representa a função expressão de arco;
- I: é a função de inicialização definida a partir de P.

A figura 7 mostra um exemplo didático de rede de Petri colorida. Trata-se de um modelo de protocolo de comunicação simples, onde uma sequência de pacotes de dados é enviada a partir de um local emissor para um local receptor, através de uma rede não confiável, sujeita a erros e perdas de pacotes de dados durante a transmissão. Inicialmente são atribuídos os conjuntos de cores referentes aos dados que irão trafegar pela rede.

$$\Sigma = \{INT, DATA, INT \times DATA\}$$

- *INT*: Conjunto de cores responsável por armazenar o número seqüencial das mensagens a ser transmitida.
- *DATA*: Conjunto de cores necessário para armazenar o conteúdo da mensagem a ser transmitida e recebida
- *INTxDATA*: Conjunto composto pelo número da mensagem, e o dado correspondente.

As elipses e círculos representam os lugares. Elas descrevem o estado do sistema. Os retângulos são chamados de transições, que descrevem as ações. As setas indicam os arcos, cujas expressões descrevem como o estado da rede é alterado quando uma transição é habilitada. Cada lugar contém um conjunto de marcadores chamados de fichas, as quais podem conter valores de dados.

A rede de exemplo consiste em três partes: O Emissor com duas transições denominadas Envia Pacote e Confirma Recebimento. A parte Rede tem igualmente duas transições denominadas Transmite Pacote e Confirma Transmissão. Por último, na parte Receptor existe uma única transição chamada Recebe Pacote. No modelo exemplo, os lugares A, D e B, C consistem numa interface entre as três partes.

No lugar Envia, conforme a marcação inicial mostrada na figura 7, existem oito fichas do tipo *INTxDATA*. As informações com seus respectivos números seqüenciais são mostradas no quadro posicionado acima da rede. A notação *n'* *informação*, indica a existência de *n* fichas contendo a informação entre parênteses, no lugar especificado. Isto pode ser observado no lugar ProxEnvio, onde aparece a marcação 1'1, sinalizando a presença de uma ficha contendo informação do tipo 1, que no caso é a *string* "Modellin".

Desta maneira, constata-se que o lugar Envia possui oito pacotes que devem ser enviadas do emissor para o receptor de forma sequencial, (no lugar Recebidos inicialmente tem-se a *string* "") para que, ao final da transmissão, obtenha-se a mensagem "Modelling and Analysis by Means of Coloured Petri Nets". Convencionou-se que a *string* "#####" representa um valor de final de mensagem.

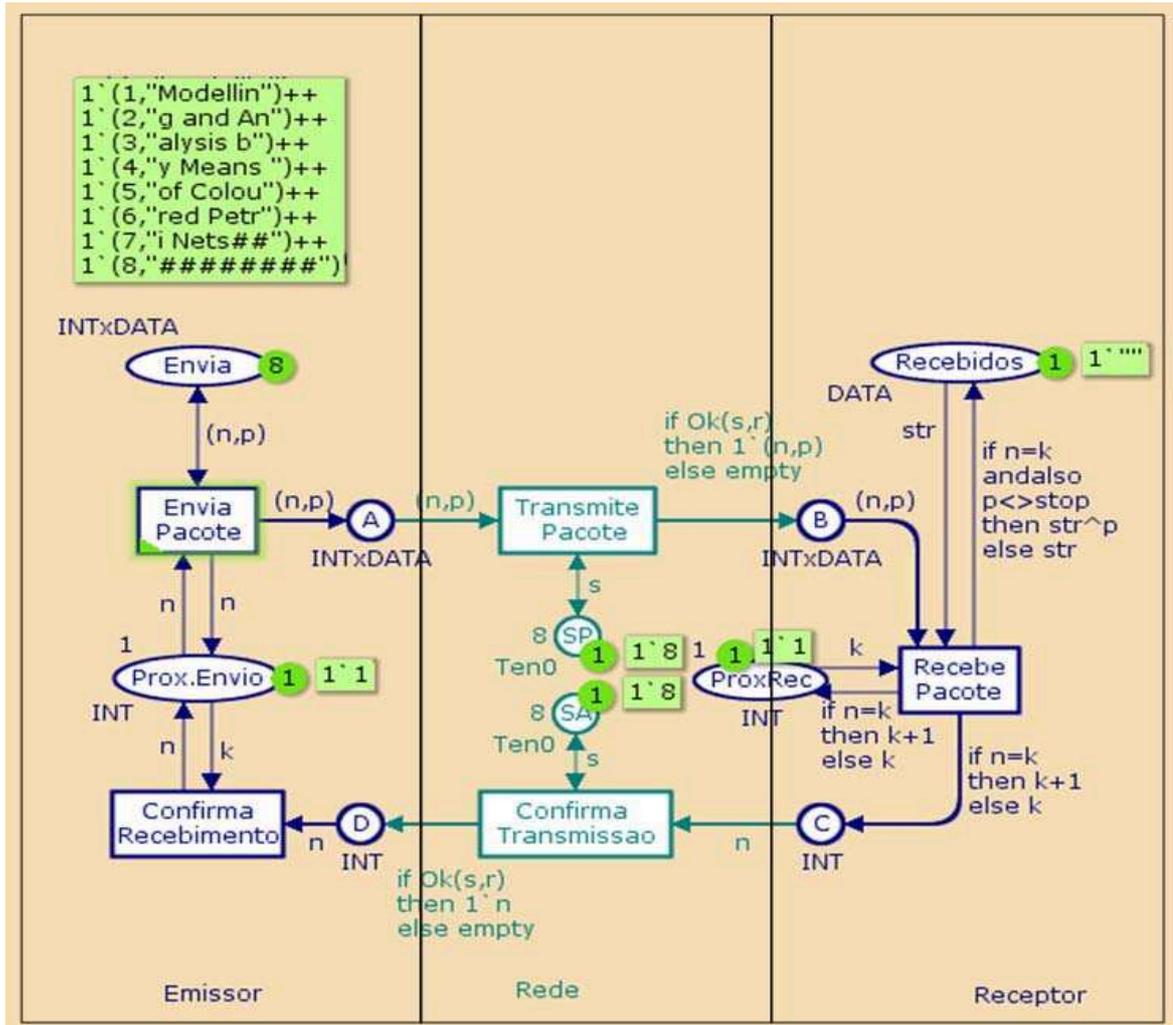


Figura 7 - Exemplo de uma rede de Petri colorida

Fonte: Adaptado de Jensen(1992).

A representação (n,p) , apresentada em alguns arcos (por exemplo, entre o lugar Envia e a transição Envia Pacote representa um fluxo de duas informações, uma do tipo INT e outra do tipo DATA, entre o lugar e a transição. Levando em conta o funcionamento dinâmico da rede, na transmissão do primeiro pacote a informação do tipo INTxDATA da primeira ficha (1, "Modellin") , flui para a transição Envia Pacote.

A informação contida no lugar ProxEnvio, que inicialmente contém uma ficha do tipo 1, garante que a mensagem que sairá do lugar Envio é a que possui o número 1. Isso se deve ao fato da variável n estar presente nos dois arcos precedentes à transição. Apesar de estarem disponíveis oito pacotes no lugar Envio, o lugar ProxEnvio consegue prover apenas uma ficha com valor 1 para o disparo da transição.

No instante que a transição Envia Pacote é disparada, surge uma ficha do tipo INT, que armazena informação referente ao número seqüencial da mensagem no lugar ProxEnvio, que corresponde a uma ficha igual a que foi consumida para permitir o disparo da transição. E, também, uma ficha com a informação completa do pacote surge no lugar A.

Nos lugares SP e AS, está representada uma ficha com o valor 8, com o objetivo de controlar o índice máximo de ordem para o pacote esperado neste modelo (que é o índice de número 8). Nesses dois lugares controla-se a probabilidade de erros na rede, proporcionando um mecanismo de simulação de ruído, onde o valor 8 representa 80% de chances de uma transmissão bem sucedida. Para outras taxas de erro, basta variar este índice entre 1 e 10.

Os arcos que ligam estes lugares às transições são bidirecionais, o que faz com que a marcação inicial deste lugar permaneça até o final da evolução da rede, pois para cada ficha consumida, é gerada uma ficha do mesmo tipo no referido lugar.

Existem também funções que são executadas no momento do disparo, e são colocadas sobre os arcos. Por exemplo, entre a transição Transmite Pacote e B existe uma função booleana chamada OK. A representação "If Ok (r,s) then 1'(n,p) else empty", será chamada no momento que ocorrer a transição. No caso de um retorno verdadeiro, surgirá no lugar B uma ficha com as informações (n, p). Caso contrário, nenhuma ficha surgirá no lugar B.

A rede evolui quando a transição Recebe Pacote é habilitada pela existência de uma ficha no lugar B. Observando os arcos que saem desta transição, constata-se que sempre ocorre um teste condicional (if n=k), indicando que o pacote recebido é realmente o pacote esperado pela rede.

Na ocorrência de um pacote não esperado, a rede evolui sem alteração na marcação dos lugares Recebidos e ProxRec. Nesse caso, ocorre apenas o surgimento de uma ficha com o número seqüencial da *string* esperada, para que seja enviada uma mensagem de reconhecimento, de forma que o emissor envie um novo pacote.

Quando uma *string* com o número seqüencial correspondente é recebida, ocorre a evolução com incremento do conteúdo da ficha presente no lugar ProxRec, e também no lugar C. A função no arco que liga a transição Pacote Recebido e o lugar Recebidos, determina se o pacote foi recebido corretamente (n=k), e o pacote

não é um terminador ($p \llcorner \text{stop}$). A *string* de conteúdo do mesmo será adicionada ao valor da variável *str*.

Na marcação inicial, indicada na figura 7, observa-se uma ficha com uma *string* de tamanho zero. Esta *string* é utilizada para o disparo da transição, e no caso de recebimento de um pacote válido, é retornada com a adição do pacote. O “*else*”, parte da expressão empregada no arco, indica que caso ocorra o recebimento de um pacote inválido, a ficha utilizada retorna ao lugar Recebido com seu valor inalterado.

Seguindo o fluxo de pacotes na rede, a partir do lugar C, observa-se o mesmo mecanismo usado na transição Transmite Pacote para a transição Confirma Transmissão, como forma de simulação de erros de transmissão de pacote. Agora a variável em uso é a *r*, que através de uma entrada do usuário, pode ser testada e assim disponibilizar uma ficha no local D. Por fim, com a habilitação da transição Confirma Transmissão e a ocorrência de uma ficha no local D, a transição Confirma Recebimento é sensibilizada. Esta transição atualizará o número do pacote que deverá ser transmitido, atualizando a ficha existente no local ProxEnvio.

Finalizando, após diversas simulações, a rede atinge a marcação de fim, com a mensagem sendo inteiramente transmitida no local Recebidos e um valor de número igual a 9, nos locais ProxEnvio e ProxRec.

Após a apresentação das redes de Petri e seu formalismo, a revisão teórica prossegue com definições importantes para o entendimento da linha de conhecimento demonstrada nesta pesquisa.

2.2. Processos

Num contexto genérico, processo é o conjunto sequencial e peculiar de ações que objetivam atingir uma meta. Do latim *procedere*, é o verbo que indica a ação ir para frente, de avançar (*pro+cedere*). Com uma grande diversidade de significados nos mais diferentes campos de conhecimento, para efeito desta pesquisa, a busca e a apresentação do termo ficou restrito a área de administração empresarial.

Valeriano (1998) esclarece que processo é um conjunto de recursos e atividades inter-relacionadas (os subprocessos) que transformam insumos em produtos ou resultados. Os insumos são genericamente chamados de entrada (“*input*”) e os produtos, de saída (“*output*”).

Para Davenport (1993), processo pode ser definido com um conjunto de atividades estruturadas e medidas destinadas a resultar em um produto específico para um determinado cliente ou mercado.

Segundo a BPMN - Business Process Management Notation (2006), processo é qualquer atividade, ou grupo de atividades concatenadas, executadas dentro de uma companhia ou organização, que transforma entradas em saídas.

A definição adotada para este trabalho é a proposta pela WfMC – Workflow Management Coalition (1995), para a qual um processo normalmente compreende um número de atividades discretas, associadas a operações humanas e/ou computadorizadas, e regras que normatizam a progressão do processo através das várias atividades. A definição de processo pode ser expressa na forma gráfica ou textual ou na forma de uma linguagem de notação formal.

Observa-se que apesar dos autores destacarem aspectos distintos em relação a processo, nota-se que a estruturação, serialização e objetivo são características comuns a todas as definições.

A figura 8 representa um conjunto ordenado de forma serial e paralela de oito atividades de um processo, conectadas com a finalidade de atingirem um objetivo comum.

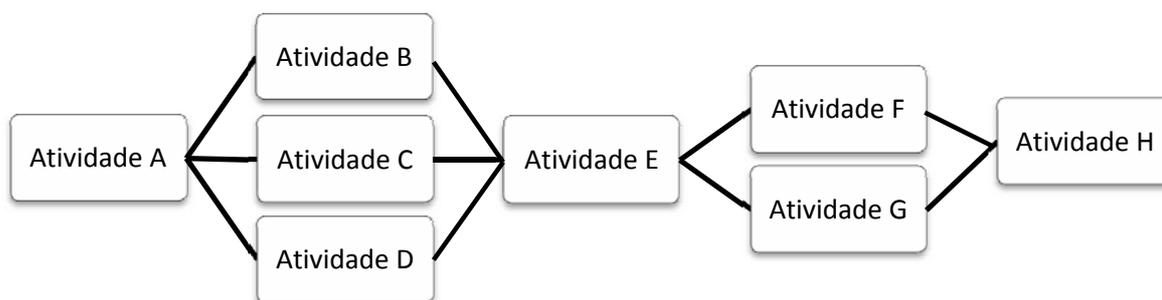


Figura 8 - Representação formal de um processo

Fonte: Desenvolvido pelo autor.

A esse conjunto de atividades estruturadas e formalizadas de um processo representado na figura acima, denomina-se fluxo de trabalho - *workflow*, cujo conceito é apresentado na próxima seção.

2.3. Workflow

Para a *Workflow Management Coalition* (1999), *workflow* é uma coleção de atividades estruturadas, de forma automatizada no todo ou em partes, para realizar um processo de negócios, como por exemplo, executar ações decisórias ou processar um pedido de compras. Um *workflow* define a sequência de execução das atividades e as condições em que cada atividade pode ser iniciada, assim como a sincronização e o fluxo de informações entre as mesmas.

Conforme Georgakopoulos e Hornick (1995), *workflow* é um grupo de tarefas estruturadas de maneira a realizar um processo de negócio. Sob o guarda-chuva do *workflow*, a expressão usualmente empregada, aplica-se a consultas a um processo de negócio, a especificação de um processo, ao *software* que executa e automatiza um processo ou ao *software* que suporta simplesmente a coordenação e colaboração das pessoas que executam um processo.

A figura 9 ilustra os diversos conceitos vinculados ao termo *workflow*, segundo o mesmo autor.

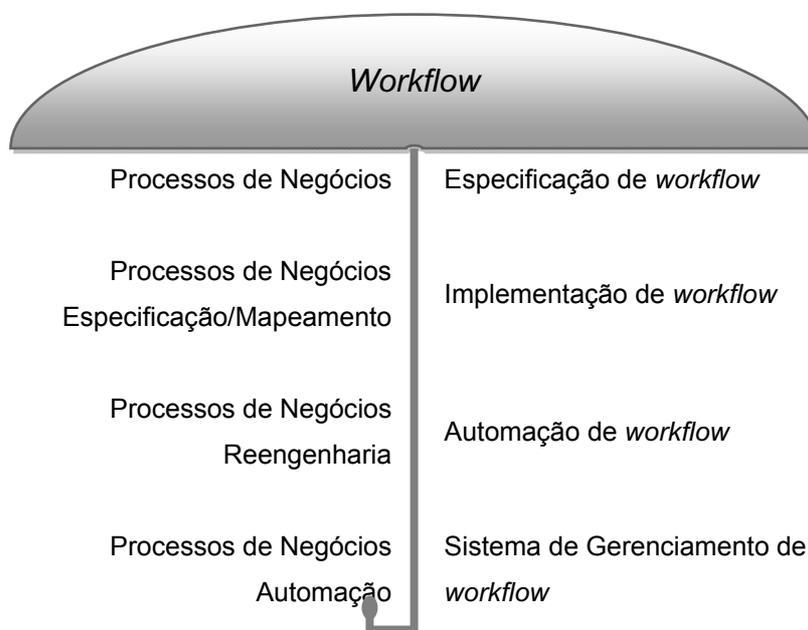


Figura 9 - Guarda-chuva de workflow

Fonte: Adaptado de Georgakopoulos e Hornick (1995).

Aalst *et al* (1994) citam que *workflow* é a descrição de um conjunto de trabalhos parcialmente ordenados, necessários para processar / executar uma ação, considerando um determinado conjunto de regras e procedimentos.

Um trabalho é um processo determinando a execução de uma sequência de atividades, conforme um dado procedimento.

Um procedimento, por sua vez, é um conjunto ordenado de atividades de controle, pares de tarefas e conjuntos de classes de recursos, e (sub) procedimentos.

Uma tarefa é uma parte da atividade que deve ser executada por um ou mais recursos em um intervalo de tempo pré-determinado. Um recurso pode executar uma tarefa isoladamente ou em conjunto com outros recursos.

A função de um par de tarefas com conjunto de classes de recursos representa o que deve ser feito (tarefa) e quais recursos devem ser alocados para a mesma. Uma classe de recursos é um conjunto de recursos. O controle de atividade tem por função determinar a rota das atividades no procedimento, bem como, a sincronização das tarefas no mesmo.

Um WfMS (*Workflow Management System*) é um sistema que define, gerencia e executa inteiramente *workflows* por meio da execução de *software* cuja sequência de atividades é determinada por uma representação da lógica do *workflow* no computador. Uma vez que um processo é definido, um WfMS certifica-se de que as atividades ocorram numa ordem própria e que os usuários sejam informados para que possam executar suas tarefas.

Para Aalst (2002), um sistema WfMS possui quatro perspectivas essenciais:

- a) Processo - refere-se às tarefas e ao roteamento dos cases
- b) Recursos - refere-se à estrutura da organização. Identifica quais os recursos, regras, perfis e grupos de pessoas que farão parte do processo de negócio e, conseqüentemente, possuirão acesso ao sistema;
- c) Informação – referem-se aos dados, instâncias de processos e seus atributos utilizados no sistema;
- d) Tarefas – referem-se às atividades do processo, suas funções e responsáveis pela execução.

A figura 10 demonstra como essas perspectivas são relacionadas em um sistema de gerenciamento de processos de negócios.

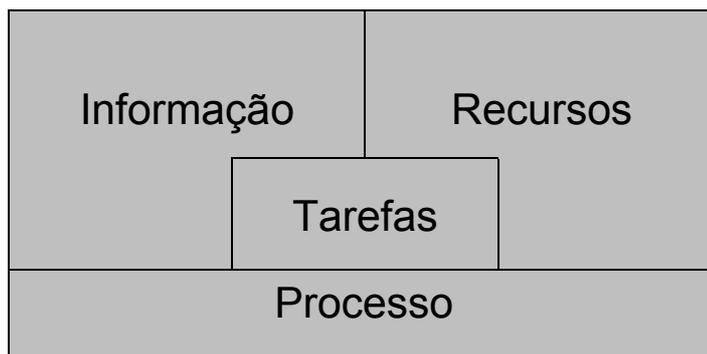


Figura 10 - Perspectivas de um WfMS

Fonte: Adaptado de Aalst (2002).

Com a adoção de um sistema de gerenciamento de workflow é possível obter melhorias distintas em:

- a) Eficiência: a automatização dos processos de negócio resulta na eliminação de muitas etapas desnecessárias;
- b) No controle de processo: o gerenciamento aprimorado dos processos de negócio é alcançado por técnicas de operação padronizadas;
- c) No serviço ao cliente: a consistência nos processos proporciona uma melhor previsibilidade nos níveis de resposta aos clientes;
- d) Flexibilidade: o controle do *software* sobre processos permite sua reformulação em linha com necessidades de mudanças de negócios.

No próximo subcapítulo, são apresentadas aspectos, características e definições de vários autores sobre processos de negócios.

2.4. Processos de negócios

Uma organização é formada por um conjunto de processos, que são as atividades de negócio realizadas pela mesma com a finalidade de agregar valor, atender as necessidades de seus clientes e gerar rendimento aos acionistas. A identificação e mapeamento dos processos internos são fundamentais para que a empresa se torne mais competitiva, racionalizando o tempo e obtendo melhores resultados.

De acordo com Silva (2002), no processo de negócio, as atividades manipulam as informações, utilizando os recursos e a organização da empresa, compreendendo uma unidade coesa voltada a um objetivo comum, normalmente direcionada a um determinado mercado ou cliente e com fornecedores definidos.

Já para Oliveira (2002), processos de negócio são os agregados de atividades que compõem os métodos de execução de um trabalho necessário para alcançar os objetivos da organização.

Rozenfeld (1996) define processo de negócio como sendo o conjunto de atividades realizadas no âmbito empresarial, que demandam recursos e estrutura organizacional. Através dos processos de negócio podem-se materializar as políticas gerenciais, os fluxos de documentos e informações, os procedimentos operacionais e os processos de manufatura.

Segundo o *Project Management Institute* (2004), os processos compreendem o conjunto de ações e atividades inter-relacionadas realizadas para obter um conjunto especificado de produtos, resultados ou serviços.

De acordo com a *Workflow Management Coalition* (1999), um processo de negócio compreende um conjunto de um ou mais procedimentos ou atividades relacionadas, as quais, coletivamente, realizam um objetivo de negócio no contexto de uma estrutura organizacional.

Davenport (1993) menciona que no gerenciamento dos processos de negócios, cada vez mais as organizações estão utilizando técnicas de *workflow*, como uma forma de melhorar a conscientização e o conhecimento acerca dos processos e para desconstruir a complexidade empresarial. A configuração e a instalação de um sistema de gerenciamento de *workflow* requerem modelos de processos que delimitem como os processos de negócios são gerenciados, modelos esses freqüentemente fornecidos pelos usuários do sistema.

Weske (2007), afirma de maneira mais abrangente que processos de negócios consistem no conjunto de atividades que são realizadas em coordenação num ambiente técnico organizacional.

Essas atividades em conjunto tornam possível atingir o objetivo do negócio, onde apesar dos processos serem conduzidos de forma exclusiva atendendo clientes internos dentro da organização, pode vir a atender clientes externos, interagindo com processos de negócios realizados por outras organizações.

Utilizando uma notação simplificada do padrão BPMN, a figura 11 representa as atividades que uma revenda realiza ao processar pedidos de compra.

O processo se inicia com o revendedor recebendo um pedido de compra. Em seguida, o fluxo das informações pode seguir dois caminhos concorrentes. No primeiro, o pedido é enviado e o pagamento é recebido. No segundo, os produtos são despachados. Quando as atividades de ambos os caminhos são executadas, os pedidos são arquivados e o processo é finalizado.

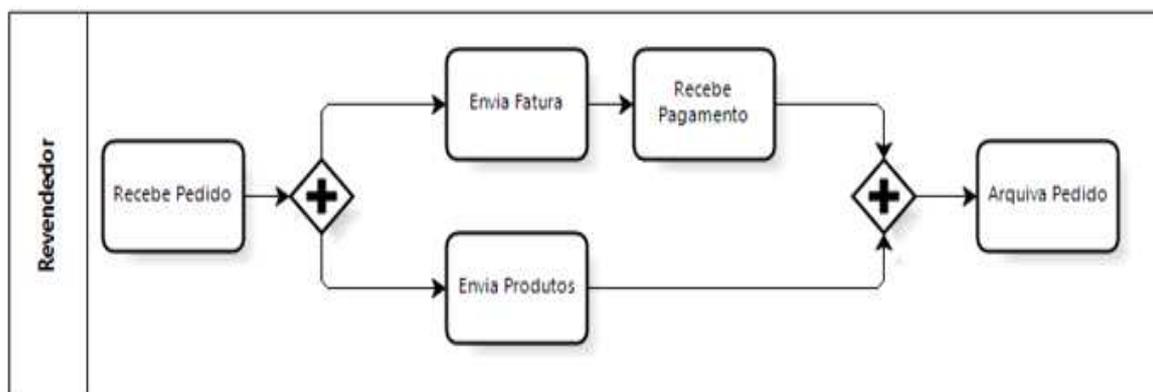


Figura 11 - Exemplo de um processo de revenda

Fonte: Adaptado de Weske (2007).

O processo de negócio mostrado na figura 12 mostra as atividades que um comprador executa ao processar um pedido de compra. O comprador envia um pedido de compra, recebe a informação para pagamento, liquida a fatura e recebe os produtos solicitados. Quando as atividades de ambos os caminhos são executadas, o processo é encerrado.

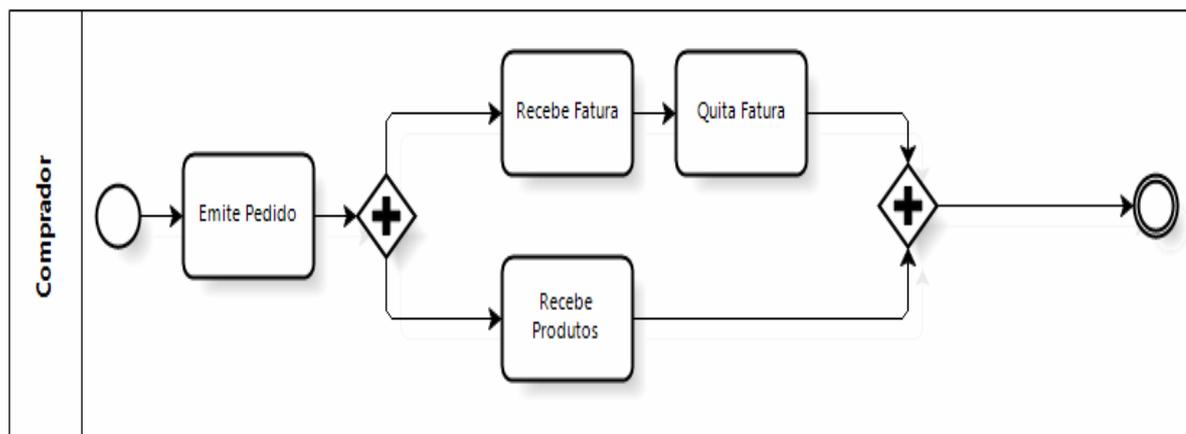


Figura 12 - Exemplo de um processo de compra

Fonte: Adaptado de Weske (2007).

Uma definição de Weske (2007) que se insere no contexto desta pesquisa, afirma que processos de negócios compreendem o conjunto de funções de negócios conectadas, onde as conexões são controladas por regras de negócio. Essas regras são específicas para cada empresa e para determinados intervalos de tempo.

Essa interação é mostrada na figura 13, onde ambos os processos de negócios, revendedor e comprador, são interligados através do fluxo de informações e de materiais, controlados por regras de negócios estabelecidas entre as partes.

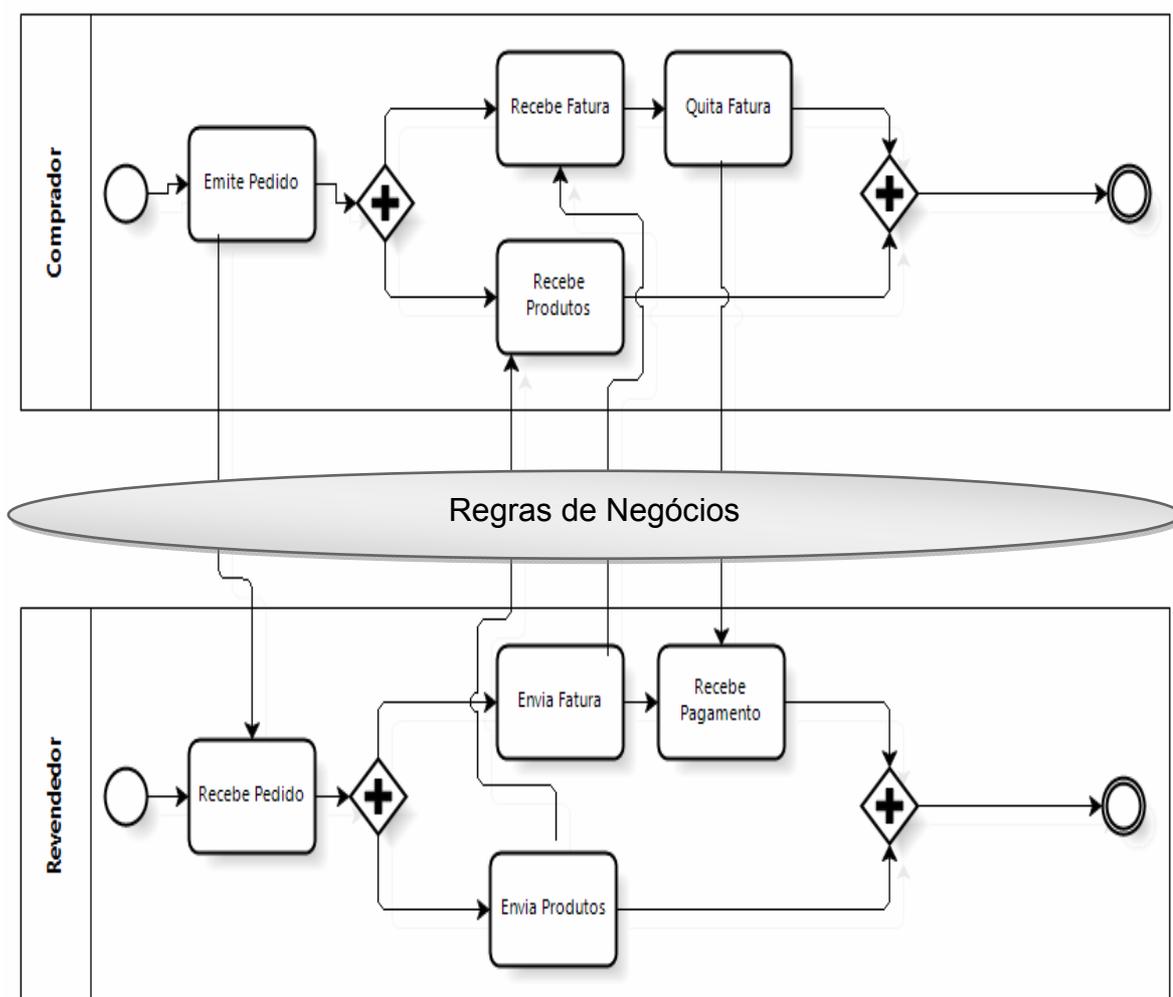


Figura 13 - Exemplo de interação entre processos de compra e venda

Fonte: Adaptado de Weske (2007).

A norma NBR ISO 9001:2000 define que um processo é um conjunto de tarefas ordenadas, interligadas e sucessivas, com início e fim definidos, através das quais são elaborados produtos e serviços. Define também que a organização deve ser retratada por seus processos de negócios principais e não pelo seu organograma. O princípio 4 da referida norma cita que um “resultado desejado é

mais eficientemente atingido quando os recursos e atividades são gerenciados como um processo”. (Associação Brasileira de Normas Técnicas, 2000).

A Figura 14 mostra do ponto de vista funcional, três diferentes abordagens para processos de negócios: enquanto na organização funcional as atividades são dirigidas pelas funções, na orientação a processos, os processos de negócios são estabelecidos, porém as funções ainda têm prevalência. Na orientação por processo, os processos de negócios dirigem todas as atividades organizacionais.

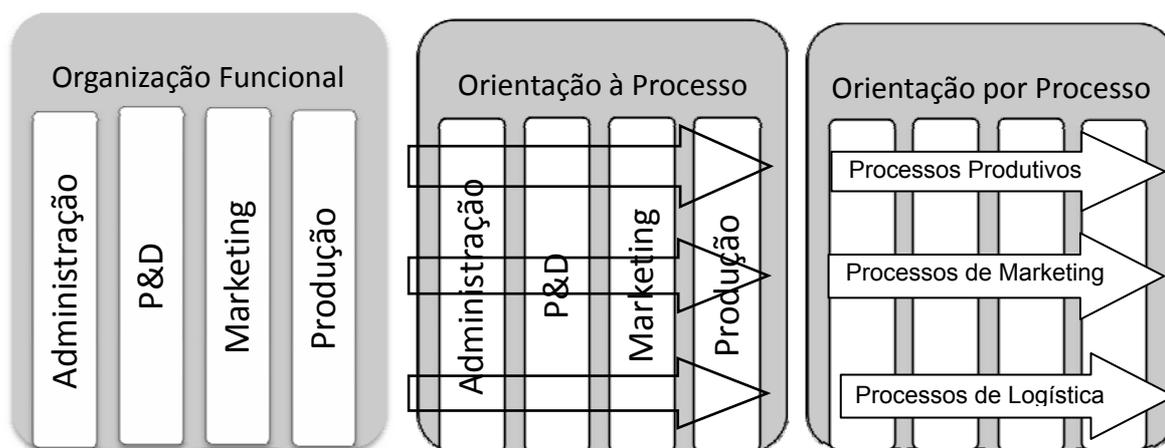


Figura 14 - Abordagens de um processo de negócio

Fonte: Adaptado de Abreu (1999).

Nessa abordagem é enfatizada a importância de entendimento e atendimento aos requisitos sistêmicos; da necessidade de considerar os processos em termos de valor agregado; da obtenção de resultados de desempenho e eficácia de processos; e da melhoria contínua de processos baseada em medições objetivas.

Medina-Mora *et al* (1992), categoriza os processos de uma organização em processos materiais, processos de informação e processos de negócios. O escopo de processos materiais abrange a montagem dos componentes e a entrega dos produtos físicos, relacionando tarefas humanas que são desempenhadas no mundo físico. Tais tarefas incluem a movimentação, armazenagem, transformação, medição e montagem de objetos físicos.

Os processos de informação são relacionados a tarefas automatizadas, (executadas por programas), e parcialmente automatizadas (quando realizadas por pessoas interagindo com computadores), que criam, processam, gerenciam e fornecem informação. Normalmente um processo de informação faz parte da estrutura organizacional e/ou dos sistemas de informação existentes.

Os processos de negócios são descrições das atividades de uma organização, implementadas como processos de informação e/ou processos materiais. Isto é, um processo é projetado para atender contratos de negócio ou satisfazer necessidades específicas do consumidor. Assim, a noção de processos de negócios está conceitualmente num nível mais alto do que a noção de processos de informação ou materiais.

Quando uma empresa faz uma leitura do seu negócio a partir dos processos de negócios, pode reavaliar cada processo com a finalidade de melhorar ou adaptá-los as mudanças exigidas pelo mercado. Essas melhorias buscam substancialmente reduzir custos, aumentar a qualidade dos produtos e serviços, a satisfação do consumidor e a eficiências das operações.

De acordo com Muehlen (2004), um processo é uma sequência de atividades que são necessárias para manipular um objeto economicamente relevante através de um objetivo específico.

Processos de negócios podem ser descritos a partir de diferentes níveis de abstração. Normalmente considerados processos de alto nível, são determinados pelas metas globais da empresa, os quais contêm interfaces com os consumidores, fornecedores ou terceiros. O gerenciamento de processos de negócios (BPM – *Business Process Management*) é considerado uma importante estratégia empresarial para criar e manter vantagens competitivas sustentáveis.

Para Jeston e Nelis (2006), BPM é a realização das estratégias corporativas de uma organização através da melhoria contínua, do gerenciamento e controle dos seus processos de negócios essenciais. De forma geral, o gerenciamento é uma função transversal, com uma abordagem estruturada que controla a utilização dos recursos e possibilita a orquestração adequada das atividades operacionais de uma empresa. As funções de gerenciamento seguem o ciclo de vida do planejamento, organização, suporte, administração e controle, e orçamento.

A Figura 15 mostra uma representação do ciclo de vida de uma estratégia de BPM. O diagrama apresenta um ciclo contínuo e ideal com sete fases, iniciando com as metas, especificações e a análise organizacional que define os parâmetros e as restrições para a modelagem do processo.

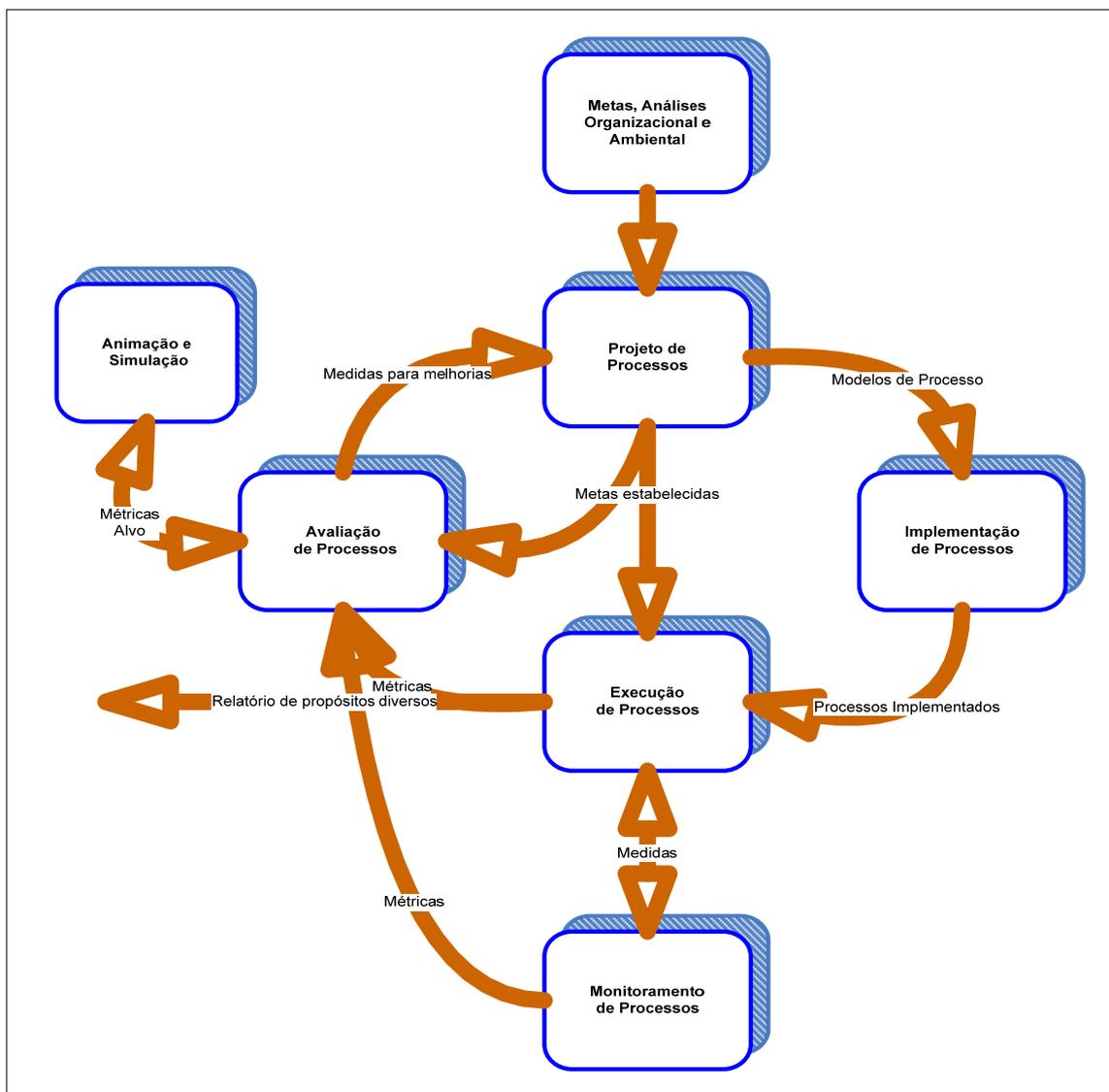


Figura 15 - Ciclo de vida do BPM

Fonte: Adaptado de Muehlen (2004).

Na fase de projeto, ocorre a identificação dos processos que a organização deseja analisar, modelar e/ou automatizar. A partir dos modelos de processo, durante a fase de implementação, processos específicos na forma manual (através de procedimentos ou manuais) ou automatizada (por meio de programas de *workflow* ou BPM), são transferidos e implementados em ambientes operacionais.

Com a execução dos processos, é possível monitorá-los em tempo real, e a avaliação e controle do processo é realizado através das trilhas de auditoria geradas pelos estágios de execução e monitoramento de processos. Com base nesse monitoramento e avaliação, é possível desenvolver planos de melhorias dos processos de negócios.

O *framework* proposto por Muehlen na figura acima, tem aderência a proposta de Vernadat (1996) que destaca as finalidades da modelagem dos processos de negócios:

- a) Melhorar a representação e o entendimento da forma como a empresa ou partes dela trabalham;
- b) Simular o comportamento de partes da empresa;
- c) Controlar, coordenar ou monitorar processos de determinadas áreas da empresa;
- d) Armazenar o conhecimento obtido e o *know-how* da empresa para uso futuro;
- e) Viabilizar a formação de uma base de dados para análises econômicas, organizacionais, quantitativas e qualitativas;
- f) Racionalizar e assegurar o fluxo de informações;
- g) Projetar ou re-projetar uma determinada área da organização, nos aspectos de natureza funcional, informacional, organizacional ou estrutural;
- h) Prover uma base de informações para tomada de decisões operacionais e organizacionais;

Ainda segundo Muehlen (2004), existem três razões pelas quais é importante compreender os diferentes aspectos de um processo de negócio: (a) avaliar o que aconteceu no passado; (b) entender o que está acontecendo no presente; e (c) construir um entendimento do que pode acontecer no futuro.

A primeira razão diz respeito ao controle do processo, baseado na análise posterior de processos de negócios completados. Pode envolver ou não a existência de representação formal pré-existente do processo de negócio em questão. Se não houver documentação do modelo do processo, ou se o escopo do processo abranger outros sistemas e domínios, o modelo pode ser gerado indutivamente através da mineração de processos. As pesquisas de ponta nesse sentido são conduzidas por Wil van der Aalst, através de um grupo de pesquisa da Universidade de Tecnologia de Eindhoven.

A segunda razão compreende o Monitoramento dos Processos de Negócio – BAM, atuando na supervisão em tempo real dos processos de negócios ativos e em execução. Por fim, na terceira razão trata-se da inteligência do processo, que

através de técnicas de planejamento de cenários e simulações dos dados dos processos de negócios, busca prever o comportamento futuro das organizações.

A figura 16 demonstra um ciclo de vida de um processo de negócios em quatro fases: (A) projeto de processos, (B) implementação / configuração de processos, (C) execução de processos, (D) diagnóstico de processos.

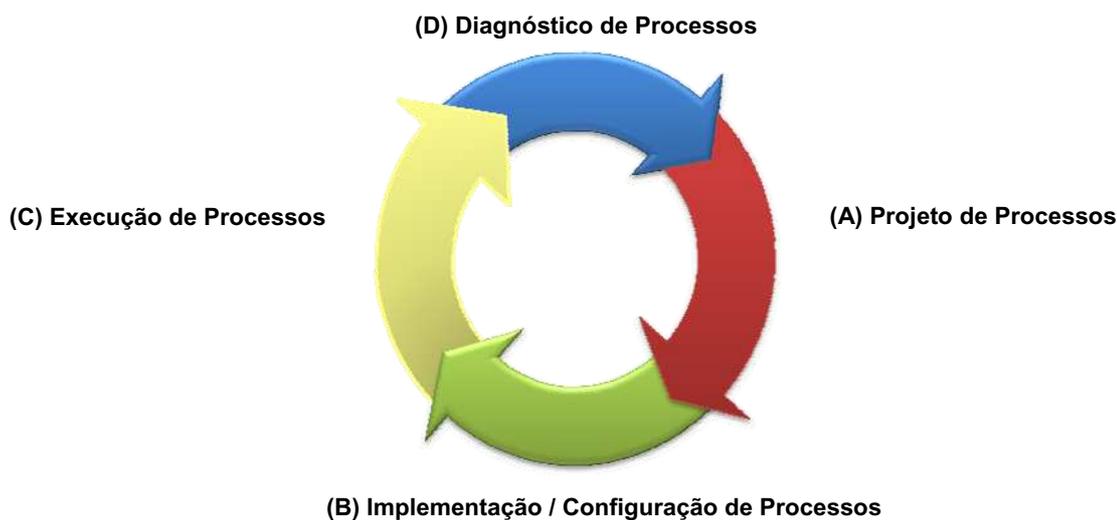


Figura 16 - Ciclo de vida de um processo de negócio

Fonte: Adaptado de Aalst *et al* (2003).

No processo tradicional de modelagem, a fase de projeto (A) é utilizada para a construção do modelo de *workflow*. Na sequência, esse modelo é implementado e configurado (B) através de um sistema, que é executado (C) conforme previsto na fase de projeto. Com base no *workflow* executado é possível coletar informações de diagnóstico que são analisadas na etapa seguinte (D). O resultado dessas análises serve para refinar o modelo de processo projetado. Na abordagem tradicional de modelagem, o foco está nas fases de projeto e implementação de processos.

Na abordagem através da mineração de processos, busca-se primeiro capturar o processo real a partir de uma execução real e depois formalizá-lo. Modelos de processo gerados podem ser utilizados para formalizar um processo que nunca foi projetado, para analisar as diferenças entre o processo especificado e o processo real, ou para implementar melhorias num modelo já conhecido. O foco está em entender de que forma o processo está sendo executado e fornecer evidências para o seu aperfeiçoamento.

Para Aalst *et al* (2003), apesar de todos os recursos e promessas na área de gestão de *workflow*, modelagem de processos de negócios é uma tarefa para

especialistas, que requer conhecimentos detalhados sobre processo em análise e sobre a linguagem de modelagem utilizada.

Outra questão não menos importante, é relacionada ao fato de que os sistemas de gestão de workflow buscam atender diferentes funcionalidades. Isso pode facilmente redundar em confusão com respeito ao que se espera de um WfMS, e desta forma a criação de padrões para a troca de informações entre os mesmos tornou-se necessária, que é o assunto abordado no subcapítulo seguinte.

2.5. WfMC – Workflow Management Coalition

Aalst e Hee (2002) afirmam que os sistemas de gerenciamento de *workflow* permitem a “extração” dos processos de gestão a partir dos programas aplicativos. Esses sistemas podem ser implementados com a finalidade de obter flexibilidade, integração de sistemas, otimização de processos, mudanças organizacionais, etc. Tudo isso significa dizer que nem sempre se sabe o que realmente pode ser esperado a respeito das funcionalidades dos sistemas de gerenciamento de *workflow*. Isso implica em um grande leque de produtos, cada um buscando atender determinadas capacidades funcionais, que sem uma padronização adequada, provoca o desenvolvimento de processos incompatíveis entre si.

Para dar uma resposta a este problema, uma organização sem fins lucrativos foi criada em 1993, com a finalidade de identificar as áreas funcionais e desenvolver especificações para serem implementadas nos sistemas de gerenciamento de *workflow*. Estas especificações destinam-se a permitir a interoperabilidade entre os diferentes sistemas de *workflow* e a melhorar a integração dos mesmos com outros serviços de TI. Nascia a WfMC - Workflow Management Coalition e seus membros originais incluíram a IBM, Hewlett Packard, Fujitsu, ICI, Staffware e em torno de 300 empresas das áreas de software e serviços.

Em 1995, a WfMC propôs um modelo de referência para especificar padrões na área de *workflow*, que é uma descrição geral da arquitetura de um sistema de gerenciamento de *workflow*, distribuídos em cinco interfaces de interoperabilidade e comunicação, que possibilitam a coexistência e interação dos

diferentes sistemas de *workflow* com o ambiente do usuário. A figura 17 ilustra esse modelo de referência.

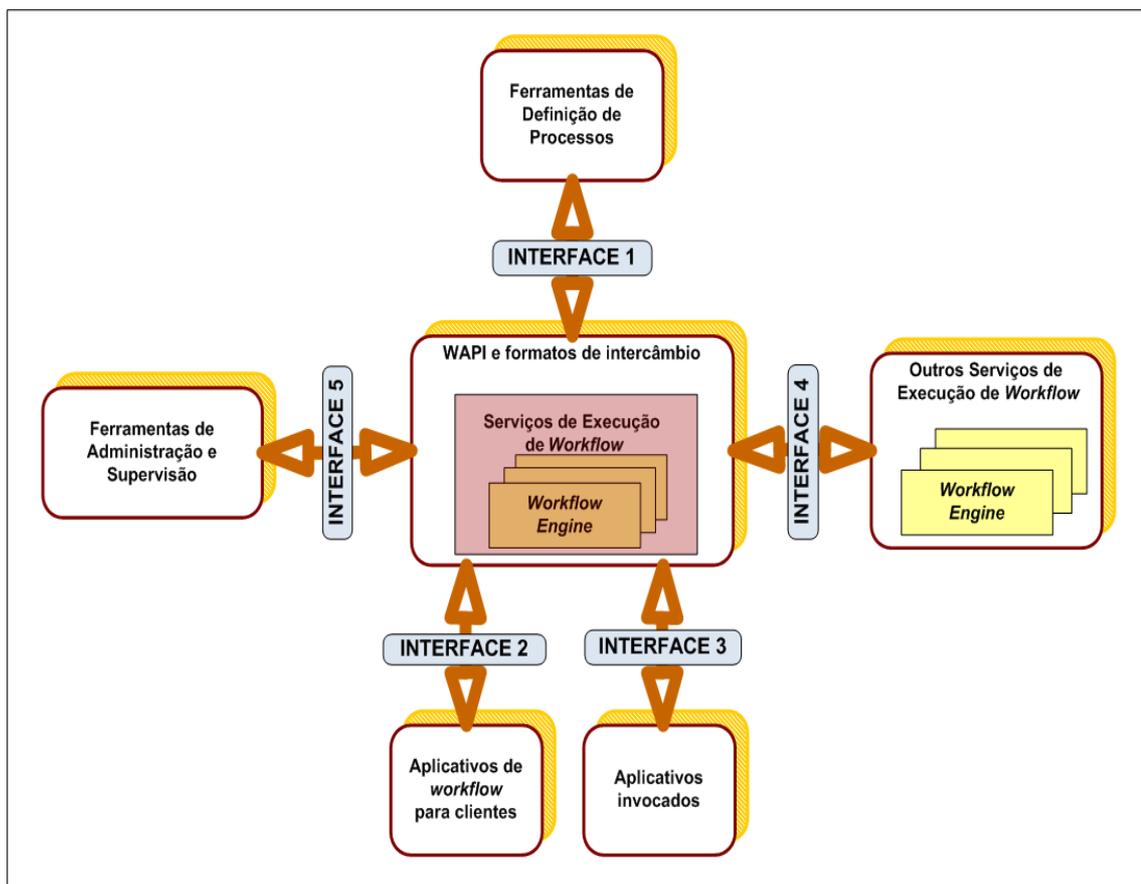


Figura 17 - Modelo de referência do Workflow - componentes e interfaces

Fonte: Workflow Management Coalition (1998).

Os cinco componentes que se comunicam, suas interligações e os formatos de intercâmbio de dados requeridos para sua interoperabilidade são detalhados como se segue:

- a) Interface 1 - Ferramentas de Definição de Processos: especifica um padrão de interface entre ferramentas de modelagem e definição de processos e os serviços de execução de *workflow*, para a importação e a exportação destas definições. Nota-se que a máquina de *workflow* é apenas um dos componentes do sistema de gerenciamento;
- b) Interface 2 - Aplicativos de *Workflow* para Clientes: define padrões para que o serviço de execução de *workflow* mantenha os itens de trabalho que os aplicativos de *workflow* oferecem ao usuário final. Através desta interface, o usuário consegue selecionar tarefas para execução, recuperar

detalhes da tarefa a ser cumprida e, por exemplo, invocar aplicações necessárias para a realização das tarefas;

- c) Interface 3 - Aplicativos Invocados: define um padrão de interface para permitir que o serviço de execução de *workflow* utilize uma variedade de aplicações que devem ser executadas para o cumprimento de determinadas tarefas.
- d) Interface 4 - Outros Serviços de Execução de *Workflow*: define uma variedade de modelos de interoperabilidade que determinam como ocorre a cooperação entre produtos de fabricantes diversos e os padrões de cada um;
- e) Interface 5 - Ferramentas de Administração e Supervisão: define um padrão para funções de monitoramento e controle, tornando possível a avaliação e extração de métricas do sistema, bem como estabelecer um modelo comum de auditoria de dados, e funções relativas à segurança, controles e autorizações dentro do sistema.

A Interface 5 é melhor detalhada, conforme mostra a figura 18, uma vez que tem relação direta com o contexto desta pesquisa.

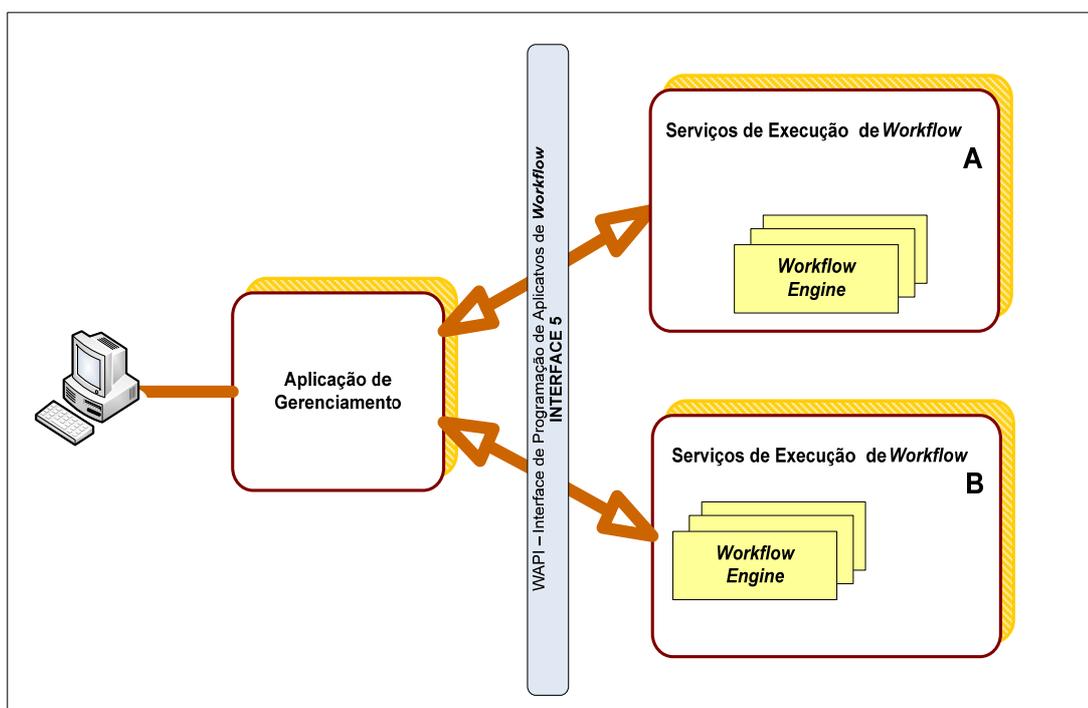


Figura 18 - Interface de sistemas de Administração e Supervisão

Fonte: WfMC (1998).

A interface mostra uma aplicação de gerenciamento independente interagindo com diferentes serviços de *workflow*, apesar de serem possíveis outros cenários alternativos de implementação; por exemplo, a aplicação de gerenciamento da aplicação pode ser parte integral de um serviço de execução, apesar de ser capaz de gerenciar várias funções através de outros domínios de *workflow*.

A especificação proposta pelo WfMC permite uma visão completa das condições do fluxo de trabalho dentro da organização, independentemente dos sistemas transacionais em uso. Compreende ainda um amplo conjunto de funções de propósito administrativo, que inclui detalhamento sobre segurança, controle e autorizações.

As áreas funcionais típicas para este padrão abrangem as seguintes operações:

- a) Gestão de Usuários: estabelecer/apagar/suspender/corrigir privilégios de usuários ou grupos de trabalho;
- b) Gestão de Funções: estabelecer/apagar/corrigir funções; relações entre participantes; configurar / desconfigurar funções de atributos;
- c) Gestão de Auditoria: operações de pesquisa, impressão, apagar trilhas de auditoria ou *logs* de eventos, etc;
- d) Controle de Recursos: adicionar, remover ou modificar as características de concorrência de uma atividade ou processo;
- e) Funções de Supervisão de Processos: mudar o estado de uma definição de processo de *workflow*, e/ou das suas instâncias; escolher entre as várias versões da definição de um processo; mudar o estado de todas as instâncias de processos ou atividades de certo tipo; fixar novos atributos a todas as instâncias de processos ou atividades de certo tipo; terminar todas as instâncias de processos;
- f) Funções de Auditoria de Processos: abrir ou fechar consultas a instâncias de atividades ou processos, com filtros opcionais; consulta de detalhes sobre instâncias de processos ou atividades, com filtros especificados.

Para Leymann (2000), utilizando ferramentas de administração e monitoração de processos, é possível armazenar em trilhas de auditoria, como uma entrada no histórico de execução, todos os eventos relevantes de um processo ou atividade, que é o assunto do próximo subcapítulo.

2.6. Log de eventos

Segundo Evans (2004), todo sistema operacional registra suas atividades em um ou mais arquivos de *log*, conhecidos como *log* de eventos. Não há um padrão industrial para a implementação do registro de eventos transacionais, que varia para os diferentes sistemas e aplicações. Para sistemas baseados em UNIX, por exemplo, a maioria dos eventos é gravado em um arquivo texto denominado Syslog, e da mesma forma, sistemas Windows mantêm múltiplos arquivos binários de log para diversas finalidades: segurança, sistema e aplicação.

Log de eventos são também conhecidos como trilhas de auditoria e podem ser empregados de diferentes finalidades, entre as quais se destacam a necessidade de satisfazer exigências legais e para obter informações estatísticas sobre os processos executados.

A figura 19 dá uma idéia geral da metodologia, onde a partir de modelos de processo que são executados em sistemas transacionais, são gerados registros eletrônicos denominados *log* de eventos.

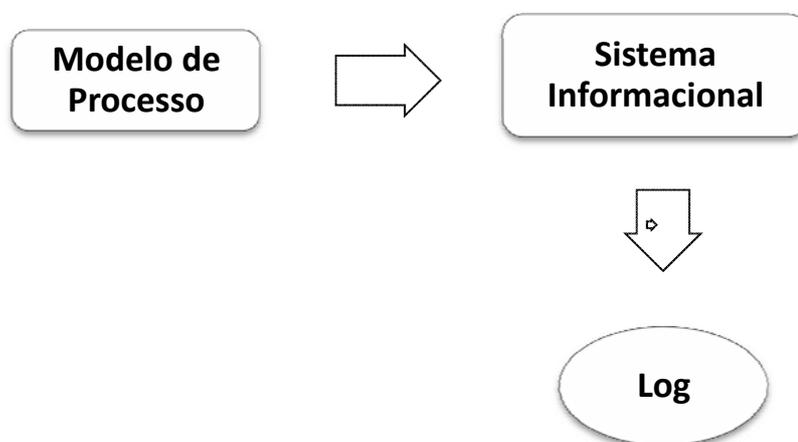


Figura 19 - Representação conceitual da mineração de processos
Fonte: Desenvolvido pelo autor.

Para Muehlen (2004), a maioria dos sistemas de *workflow* registra em um arquivo de *log* ou base de dados denominada de trilha de auditoria, os estados de transição de uma atividade bem como determinados eventos do sistema. A qualidade e o nível de detalhe das informações gravadas podem variar significativamente de sistema para sistema, e pode ter finalidades distintas tais como:

- a) Recuperação: em caso de falha no sistema, a trilha de auditoria pode ser utilizada para reiniciar ou recuperar o último estado do sistema;
- b) Execução: a trilha de auditoria pode ser empregada para determinar os recursos associados aos eventos ou atividades subsequentes de um fluxo de trabalho;
- c) Avaliação: em um contexto de controle e monitoramento de processos, a trilha de auditoria é usada por conter informações precisas e cronológicas a respeito da execução de um fluxo de trabalho ou instâncias de atividades.

Para Golani e Pinter (2003), o *log* de um fluxo de trabalho contém dados monitorados que se referem às execuções do processo, cujas atividades são ordenadas no tempo. Os autores citam também que os dados armazenados no *log* devem ser depurados antes de serem minerados, pois podem conter exceções que irão gerar instâncias de execução não compatíveis com o processo analisado.

Para Aalst *et al.*(2003), atualmente grande parte dos sistemas de informação (SCM, B2B, CRM e ERP) utilizados pelas empresas para a execução de seus processos de negócios, pode executar as tarefas envolvidas nos mesmos, sem necessariamente ter uma definição clara e explícita dos modelos de processos executados, ou ainda, fazer um acompanhamento apenas para fins de auditoria.

De qualquer modo, todos os sistemas transacionais suportam o registro rotineiro das atividades executadas na organização. Esse registro contém os dados acerca das instâncias de processos e o momento em que eles foram realizados na empresa, esperando que (a) o evento gravado seja correspondente a uma atividade do processo de negócio; (b) refira-se a uma instancia do processo; e (c) esteja totalmente ordenado.

Leymann e Roller (2000) mencionam que um *log* de eventos deve ser capaz de armazenar todos os eventos na vida de um processo, de tal forma que:

- a) Eventos tenham uma etiqueta de data e hora e sejam inteiramente ordenados;
- b) Cada evento também é referenciado a um originador (quem executa ou inicia a atividade);
- c) Cada evento se refira a uma atividade (i.e., um passo bem definido no processo);
- d) Cada evento se refere a um case (i.e., uma instância do processo).

Já Dongen e Aalst (2005) fazem uma distinção entre os eventos que são realizados e os eventos que são logados. A esses eventos registrados conhecidos como entradas de trilha de auditoria, para a finalidade de mineração de processos, são necessários que a informação armazenada atenda alguns requisitos:

- a) Cada entrada no *log* deve corresponder a um evento que ocorreu num certo tempo. Não se refere a um período de tempo em si, mas a marcação do momento de início ou do fim de uma atividade;
- b) Cada entrada no *log* deve se referir a uma única atividade, e cada atividade deve ter uma identificação exclusiva;
- c) Cada entrada no *log* deve conter a descrição do evento ocorrido com relação à atividade;
- d) Cada entrada no *log* deve se referir a uma instância específica do processo (*case*);
- e) Cada instância do processo deve pertencer a um processo específico.

Observa-se que os requisitos mínimos para composição de uma trilha de auditoria propostos por Dongen e Aalst (2005) são complementares aos citados por Leymann e Roller (2000).

Levando em conta que a informação presente em um *log* de eventos é altamente dependente da representação interna de dados existente em cada sistema transacional, e que esses sistemas atendem a diferentes propósitos em diferentes organizações, Aalst *et al* (2003) buscaram um formato de intercâmbio de dados de *logs* de eventos que pudesse ser comum a todas as plataformas.

A escolha recaiu sobre o formato XML (*Extensible Markup Language*), utilizado tanto pelos sistemas transacionais, tais como ERP, CRM e outros, como pelas ferramentas de mineração de processos. O XML é um formato para a criação de documentos com dados organizados de forma hierárquica, como encontrado em documentos de texto formatados, imagens vetoriais ou bancos de dados. Assim, um *log* de *workflow* é um documento estruturado e válido no padrão XML, com o elemento de topo denominado de *WorkFlow_log*.

A partir desse formato, criou-se uma definição de tipos de documento – DTD, que especifica a sintaxe para *logs* de *workflow*. A figura 20, obtida a partir do programa editor XMLSpy mostra que o *log* consiste de informações sobre o programa fonte e sobre um ou mais processos de *workflow*.

XML		version	1.0																																																	
		encoding	UTF-8																																																	
Comment		edited with XMLSpy v2009 sp1 (http://www.altova.com) by Eliseu Ramos (Student)																																																		
ENTER_NAME_OF_ROOT_ELEMENT...		EMPTY																																																		
Workflow_log sequence of																																																				
		source	0 or 1																																																	
		process	1 or more																																																	
source		EMPTY																																																		
source attribute list																																																				
	Att Name	Att Ty...	Att Values																																																	
1	program	Choice	<table border="1"> <thead> <tr> <th>Att Pres...</th> <th>Att Default</th> </tr> </thead> <tbody> <tr> <td>#REQUIRED</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Att Name</th> <th>Att Ty...</th> <th>Att Values</th> <th>Att Presence</th> <th>Att Default</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>staffware</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>inconcert</td> <td></td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>pnet</td> <td></td> <td></td> <td></td> </tr> <tr> <td>4</td> <td>IBM_MQ</td> <td></td> <td></td> <td></td> </tr> <tr> <td>5</td> <td>other</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Att Pres...	Att Default	#REQUIRED		Att Name	Att Ty...	Att Values	Att Presence	Att Default	1	staffware				2	inconcert				3	pnet				4	IBM_MQ				5	other																		
Att Pres...	Att Default																																																			
#REQUIRED																																																				
Att Name	Att Ty...	Att Values	Att Presence	Att Default																																																
1	staffware																																																			
2	inconcert																																																			
3	pnet																																																			
4	IBM_MQ																																																			
5	other																																																			
process sequence of																																																				
		case	0 or more																																																	
process attribute list																																																				
	Att Name	Att Ty...	Att Values	Att Presence	Att Default																																															
1	id	ID		#REQUIRED																																																
2	description	CDATA			none																																															
case sequence of																																																				
		log_line	0 or more																																																	
case attribute list																																																				
	Att Name	Att Ty...	Att Values	Att Presence	Att Default																																															
1	id	ID		#REQUIRED																																																
2	description	CDATA			none																																															
log_line sequence of																																																				
		task_name																																																		
		task_instance	0 or 1																																																	
		event	0 or 1																																																	
		date	0 or 1																																																	
		time	0 or 1																																																	
task_name		#PCDATA																																																		
task_instance		#PCDATA																																																		
event		EMPTY																																																		
event attribute list																																																				
	Att Name	Att Ty...	Att Values																																																	
1	kind	Choice	<table border="1"> <thead> <tr> <th>Att Pres...</th> <th>Att Default</th> </tr> </thead> <tbody> <tr> <td>#REQUIRED</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Att Name</th> <th>Att Ty...</th> <th>Att Values</th> <th>Att Presence</th> <th>Att Default</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>normal</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>schedule</td> <td></td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>start</td> <td></td> <td></td> <td></td> </tr> <tr> <td>4</td> <td>withdraw</td> <td></td> <td></td> <td></td> </tr> <tr> <td>5</td> <td>suspend</td> <td></td> <td></td> <td></td> </tr> <tr> <td>6</td> <td>resume</td> <td></td> <td></td> <td></td> </tr> <tr> <td>7</td> <td>abort</td> <td></td> <td></td> <td></td> </tr> <tr> <td>8</td> <td>complete</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Att Pres...	Att Default	#REQUIRED		Att Name	Att Ty...	Att Values	Att Presence	Att Default	1	normal				2	schedule				3	start				4	withdraw				5	suspend				6	resume				7	abort				8	complete			
Att Pres...	Att Default																																																			
#REQUIRED																																																				
Att Name	Att Ty...	Att Values	Att Presence	Att Default																																																
1	normal																																																			
2	schedule																																																			
3	start																																																			
4	withdraw																																																			
5	suspend																																																			
6	resume																																																			
7	abort																																																			
8	complete																																																			
date		#PCDATA																																																		
time		#PCDATA																																																		

Figura 20 - XML DTD – Definição de tipo de documento XML

Fonte: Desenvolvido pelo autor.

Cada processo de *workflow* (elemento de processo) consiste em uma sequência de *cases* (elemento *case*) e cada *case* consiste em uma sequência de linhas de *log* (elemento *log_line*). Tanto os processos como os *cases* possuem uma identificação e uma descrição.

Cada linha no *log* contém o nome da tarefa (elemento *task_name*). A linha ainda pode ter informação sobre a instância da tarefa (elemento *task_instance*), o tipo de evento (elemento *event*), a data (elemento *date*), e o horário do evento (elemento *time*).

É importante certificar-se que a descrição do processo e do *case* são únicas para cada processo e *case* respectivamente. O nome da tarefa deve ser o identificador único para a tarefa dentro do processo. Se existem duas ou mais tarefas no processo com o mesmo nome de tarefa, elas assumem o mesmo nome.

Apesar de assumir que tarefas possuem um único nome, existem múltiplas instâncias para uma mesma tarefa. Considere o exemplo de um loop que faz com que uma tarefa seja executada diversas vezes para um determinado *case*. Assim, é possível inserir o elemento *task_instance* na *log_line*. Este elemento é normalmente um número, isto é, se a tarefa A é executada pela quinta vez, o elemento *task_name* é "A" e o elemento *task_instance* é "5". A data e horário são elementos opcionais.

A semântica da maioria das construções é auto-explicativa, exceto para o elemento evento, isto é, o tipo de evento, que descreve os possíveis estados de uma tarefa, desde sua criação até a finalização. Esses estados são classificados como *normal*, *schedule*, *start*, *withdraw*, *suspend*, *resume*, *abort* e *complete*.

Dongen e Aalst (2005), cientes das dificuldades de se utilizar *log* de eventos, com semânticas e sintaxes diferenciadas, oriundos dos diversos sistemas transacionais, e dos requisitos mínimos de informações para a mineração de processos, desenvolveram um meta modelo de *log* de eventos adequado para mineração de processos (MXML).

Um meta modelo segundo Muehlen e Rosemann (1998) é o desenho estruturado que descreve um modelo básico de elementos e o relacionamento entre os elementos do modelo assim como sua semântica. Define também as regras para o uso desses elementos e o relacionamento entre eles.

A figura 21 ilustra o meta modelo desenvolvido, a partir de um diagrama de classe no formato UML.

Nota-se que para cada *log* de evento (campo *WorkflowLog*) executa um ou mais processos (campo *Processo*). Cada processo (campo *Processo*) possui uma ou mais instâncias de processo ou *cases* (campo *Instância do Processo*).

Da mesma maneira, cada instância do processo tem uma ou mais entradas de trilha de auditoria (campo *AuditTrailEntry*), que por sua vez, contem no mínimo uma descrição do evento que a originou (campo *WorkflowModelElement*).

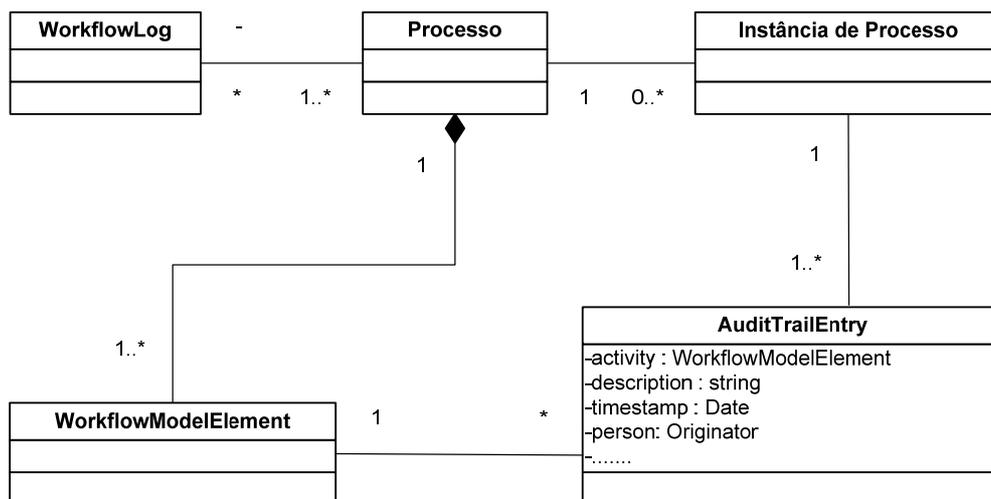


Figura 21 - Meta modelo do formato MXML

Fonte: Adaptado de Dongen e Aalst (2005).

Quando uma atividade é criada (*WorkflowModelElement*) ela pode assumir os seguintes comportamentos (tipos de eventos) que são suportados pelo sistema:

- a) *Schedule*: o controle das atividades pertence ao sistema de informação;
- b) *Reassign*: a atividade é atribuída a uma determinada pessoa ou grupo;
- c) *Start*: o usuário pode iniciar a atividade que foi atribuída a ele;
- d) *Resume*: o usuário pode reiniciar uma atividade atribuída a ele;
- e) *Suspend*: o usuário pode reiniciar uma atividade atribuída a ele;
- f) *Autoskip*: a atividade foi transferida automaticamente pelo sistema;
- g) *Manualskip*: a atividade foi transferida manualmente pelo usuário;
- h) *Withdraw*: a atividade não é realizada e foi retirada pelo usuário;
- i) *Complete*: uma atividade iniciada foi concluída;
- j) *ate_abort*: uma atividade iniciada não se completou e foi cancelada;

Traduzindo o meta modelo da figura 21 para o formato genérico XML adaptado para a armazenagem de *logs* de eventos e facilmente portátil entre as ferramentas de mineração de processos, chega-se ao esquema da figura 22, onde

as descrições são auto-explicativas exceto em duas situações, apontadas pelo círculo em destaque.

A primeira, no caso do elemento Data, é possível armazenar dados genéricos de texto e contém também uma lista de elementos de atributos. Conforme o nível de abstração pode guardar informações a respeito do ambiente em que o *log* foi gerado. A segunda situação é relativa ao elemento Source, que pode ser usado para armazenar informações a respeito do sistema que originou o log, ou ainda sobre as configurações desse mesmo sistema.

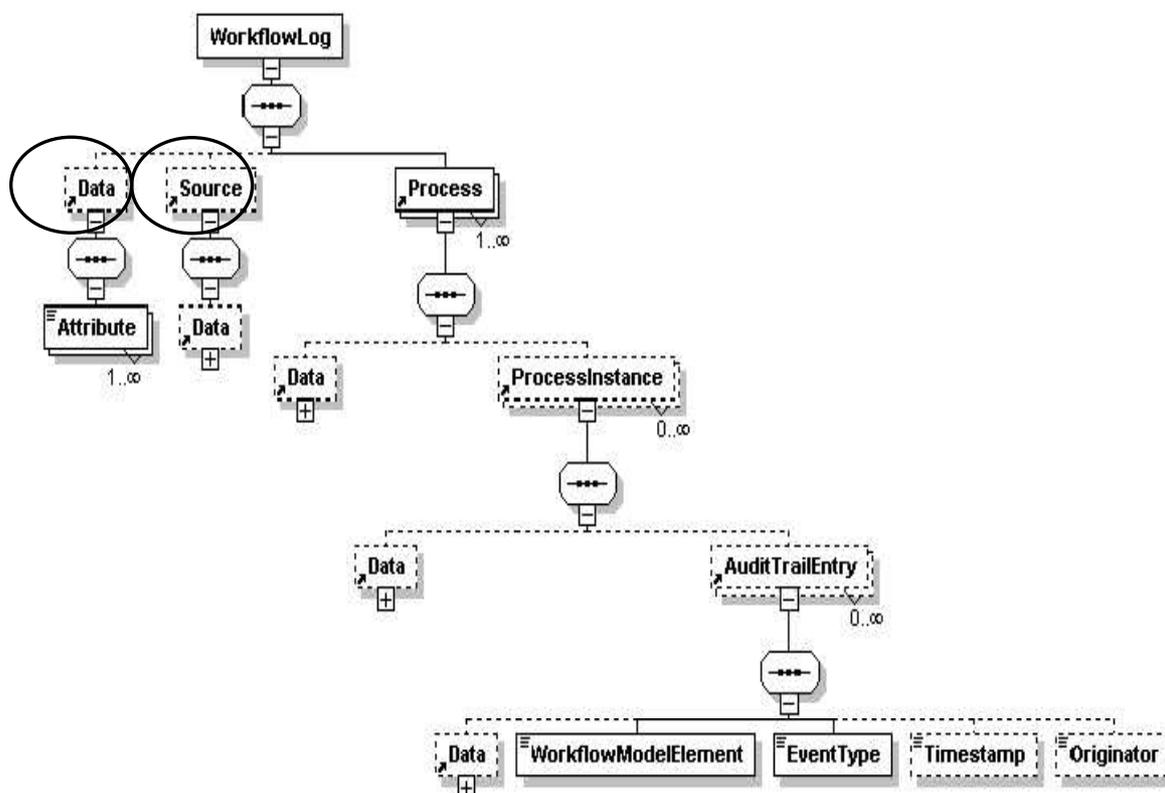


Figura 22 - Formato XML de mineração de processos

Fonte: Adaptado de Dongen e Aalst (2005).

As figuras 23 e 24 ilustram um exemplo genérico de um *log* de eventos no formato MXML, editado através do aplicativo editor XMLSpy. Ela mostra um *log* de eventos de um processo de negócio com as seguintes características:

- a) 5 cases;
- b) 31 eventos;
- c) 9 classes de eventos
- d) 1459 instâncias de processo;
- e) 1 tipo de evento;
- f) 1 originador;

The screenshot shows the XMLSpy interface for an MXML log. The tree view on the left shows the following structure:

- WorkflowLog
 - description: Test log for conformance analysis
 - Source
 - program=none
 - Data
 - Attribute
 - name: program
 - Text: name: , desc: , data: (program=none)
 - Process
 - id: 0
 - description:
 - ProcessInstances (5)
 - id: 1, description: 0
 - Data
 - Attribute
 - name: numSimilarInstances
 - Text: 1207
 - AuditTrailEntry (5)

WorkflowModelElement	EventType
1 A	complete
2 B	complete
3 D	complete
4 E	complete
5 I	complete
 - id: 2, description: 1
 - Data
 - Attribute
 - name: numSimilarInstances
 - Text: 145
 - AuditTrailEntry (7)

WorkflowModelElement	EventType
1 A	complete
2 C	complete
3 D	complete
4 G	complete
5 H	complete
6 F	complete
7 I	complete
 - id: 3, description: 2
 - Data
 - Attribute
 - name: numSimilarInstances
 - Text: 56
 - AuditTrailEntry (7)
 - id: 4, description: 3
 - Data
 - Attribute
 - name: numSimilarInstances
 - Text: 23
 - AuditTrailEntry (6)
 - id: 5, description: 4
 - Data
 - Attribute
 - name: numSimilarInstances
 - Text: 28
 - AuditTrailEntry (6)

Figura 23- Exemplo genérico de um log MXML visualizado a partir do XMLSpy

Fonte: Desenvolvido pelo autor.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" description="Test log for conformance analysis">
3  <Source program="name: , desc: , data: {program=none}">
4  <Data>
5  <Attribute name="program" name: , desc: , data: {program=none}</Attribute>
6  </Data>
7  </Source>
8  <Process id="0" description="">
9  <ProcessInstance id="0" description="">
10 <Data>
11 <Attribute name="numSimilarInstances">1207</Attribute>
12 </Data>
13 <AuditTrailEntry>
14 <WorkflowModelElement>A</WorkflowModelElement>
15 <EventType>complete</EventType>
16 </AuditTrailEntry>
17 <AuditTrailEntry>
18 <WorkflowModelElement>B</WorkflowModelElement>
19 <EventType>complete</EventType>
20 </AuditTrailEntry>
21 <AuditTrailEntry>
22 <WorkflowModelElement>D</WorkflowModelElement>
23 <EventType>complete</EventType>
24 </AuditTrailEntry>
25 <AuditTrailEntry>
26 <WorkflowModelElement>E</WorkflowModelElement>
27 <EventType>complete</EventType>
28 </AuditTrailEntry>
29 <AuditTrailEntry>
30 <WorkflowModelElement>I</WorkflowModelElement>
31 <EventType>complete</EventType>
32 </AuditTrailEntry>
33 </ProcessInstance>
34 <ProcessInstance id="1" description="">
35 <Data>
36 <Attribute name="numSimilarInstances">145</Attribute>
37 </Data>
38 <AuditTrailEntry>
39 <WorkflowModelElement>A</WorkflowModelElement>
40 <EventType>complete</EventType>
41 </AuditTrailEntry>
42 <AuditTrailEntry>
43 <WorkflowModelElement>C</WorkflowModelElement>
44 <EventType>complete</EventType>
45 </AuditTrailEntry>
46 <AuditTrailEntry>
47 <WorkflowModelElement>D</WorkflowModelElement>
48 <EventType>complete</EventType>
49 </AuditTrailEntry>
50 <AuditTrailEntry>
51 <WorkflowModelElement>G</WorkflowModelElement>
52 <EventType>complete</EventType>
53 </AuditTrailEntry>
54 <AuditTrailEntry>
55 <WorkflowModelElement>H</WorkflowModelElement>
56 <EventType>complete</EventType>
57 </AuditTrailEntry>
58 <AuditTrailEntry>
59 <WorkflowModelElement>F</WorkflowModelElement>
60 <EventType>complete</EventType>
61 </AuditTrailEntry>
62 <AuditTrailEntry>
63 <WorkflowModelElement>I</WorkflowModelElement>
64 <EventType>complete</EventType>
65 </AuditTrailEntry>
66 </ProcessInstance>
67 <ProcessInstance id="2" description="">
100 <ProcessInstance id="3" description="">
129 <ProcessInstance id="4" description="">
158 </Process>
159 </WorkflowLog>
160

```

Figura 24 - Detalhamento XML do exemplo de log de eventos da figura 23

Fonte: Desenvolvido pelo autor.

A figura 25 ilustra segundo Medeiros e Gunther (2005), as potencialidades de um *log* de eventos, onde a partir da informação encontrada pela mineração de processos, pode-se desenvolver novos sistemas que suportam a execução de processos de negócios, ou ser ferramenta de *feedback* no auxílio em auditoria, na análise e melhoria no desempenho de processos de negócios que já estejam implementados e segurança dos sistemas de informação.

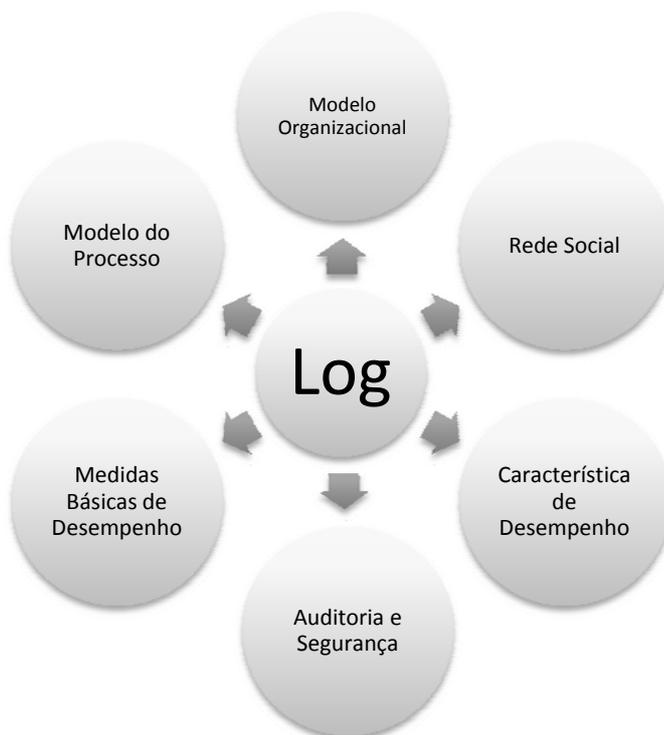


Figura 25 - Usos para log de eventos

Fonte: Adaptado de Medeiros e Gunther (2005).

Os *logs* de eventos como já citado, constituem o ponto de partida para a mineração de processos, que pode ser observada de três diferentes perspectivas:

- a) Perspectiva do processo: focaliza o fluxo de controle do processo. O principal objetivo desta perspectiva é encontrar uma boa caracterização de todos os caminhos possíveis, expressada em termos de uma rede de Petri ou EPC;
- b) Perspectiva da organização: focaliza o originador, ou seja, quais executores de atividades estão envolvidos e como eles se relacionam. Mostra a estrutura da organização, classificando as pessoas em termos de funções e unidades organizacionais;

- c) Perspectiva da instância do processo: *case* – o foco é nas propriedades dos *cases*, que podem ser caracterizados pelo caminho percorrido dentro do processo ou pelos originadores que executam atividades. Podem também ser caracterizados pelos valores dos elementos de dados correspondentes.

Conforme detalhado neste tópico, a leitura dos arquivos de *log* de eventos permite que se realize a mineração de processos, cujo objetivo em si é analisar processos de negócios.

Essa técnica é freqüentemente utilizada quando a descrição formal do processo não pode ser obtida por outros meios ou a qualidade da documentação existente é questionável e será mais bem aprofundada no subcapítulo a seguir.

2.7. Mineração de processos

As trilhas de auditoria de um sistema de gestão de *workflow*, os *logs* de transação de um sistema de gestão empresarial, os registros eletrônicos dos pacientes de um hospital, são exemplos que podem ser usados para descobrir os modelos, descrever processos, organizações e produtos. Além disso, os *logs* de eventos podem ser utilizados para comparar *logs* de eventos de modelos pré-existentes e avaliar se a realidade observada está de acordo com o modelo prescrito.

São três as técnicas básicas de mineração de processos, classificadas com base na existência ou não de modelos *a priori*:

- a) Descoberta: não existe modelo *a priori*, a construção de modelo é feita a partir da leitura do *log* de evento utilizando algoritmos;
- b) Conformidade: existe um modelo *a priori*, este modelo é comparado com o *log* de evento e as discrepâncias entre eles são analisadas. A verificação de conformidade pode ser utilizada para detectar desvios, localizar, explicar e medir a severidade destes desvios;
- c) Extensão: existe um modelo *a priori*, e este modelo é enriquecido com novos aspectos ou perspectivas. O objetivo não é avaliar a conformidade e sim aprimorar o modelo.

Rozinat e Aalst (2006) mencionam que as técnicas de mineração de processos têm demonstrado serem ferramentas valiosas na compreensão de como os processos de negócios são desenvolvidos nas empresas.

A partir da leitura de um arquivo contendo o registro real de todos os processos executados, essas técnicas permitem três oportunidades: a) conhecer o processo de negócio; b) testar a conformidade dos registros; c) entender os aspectos e perspectivas do modelo. Todas podem contribuir significativamente no projeto e na melhoria dos processos de negócios.

Wen *et al* (2006), citam entretanto que a construção de modelos de processos a partir de rascunhos é uma tarefa difícil, que consome muito tempo, sujeita a erros e exige o envolvimento de especialistas. Além disso, existem discrepâncias entre os modelos pré-definidos e os modelos realmente executados.

A mineração de processos fornece uma nova possibilidade na construção de modelos de processos, através da leitura e interpretação de *log* de eventos ou arquivos de auditoria, gerados pela maioria dos sistemas de informação transacionais, tais como SCM, B2B, CRM e ERP. Pode ainda ser utilizado para analisar e aperfeiçoar modelos predefinidos de processos.

Rosinat e Aalst (2006) exemplificam que enquanto o processo de descoberta aponta na extração automática de modelos de processo a partir do *log* de eventos de baixo nível, a verificação de conformidade está interessada na comparação entre um modelo de processo existente e seu correspondente *log* de eventos, tipicamente utilizada em sistemas de auditoria.

A figura 26 apresenta uma visão global da função de verificação de conformidade no contexto de mineração de processos, em relação aos métodos de descoberta e de extensão.

Na medida em que existe um alinhamento entre um processo e seu modelo obtido a partir do *log*, a técnica de extensão pode ser utilizada para projetar informações de diagnóstico derivadas do *log* de eventos do modelo, muito útil para visualizar gargalos de desempenho no modelo de processo.

O principal benefício das técnicas de mineração de processos é que a informação é objetivamente compilada, capturando o que de fato está ocorrendo de acordo com o *log* de eventos e não o que as pessoas pensam que está sendo realizado na organização.

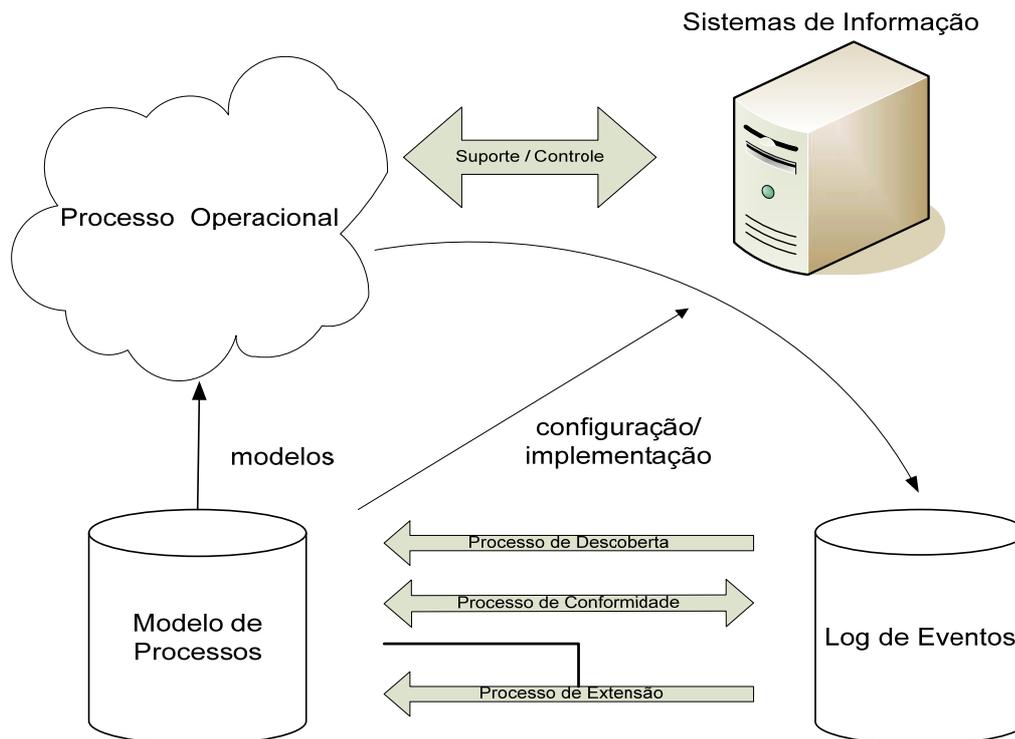


Figura 26 - A função de verificação no contexto de mineração de processos

Fonte: Adaptado de Rosinat & Aalst (2006).

O tipo de informação presente no *log* de eventos determina como citado no tópico anterior, quais perspectivas de mineração de processos pode empregada:

- a) a perspectiva do controle de fluxo pode ser minerada se o *log* contiver as tarefas que foram executadas no processo e for possível inferir sua ordem de execução e vincular essas tarefas com instâncias de processos individuais;
- b) a perspectiva organizacional, o *log* fornece informações sobre recursos - pessoas e sistemas que executam as tarefas. Baseada na transferência de trabalho ou parâmetros de alocação de recursos descobre-se também regras e unidades organizacionais;
- c) a perspectiva do case pode ser descoberta quando o *log* fornece informações detalhadas sobre as atividades, como valores dos campos de dados que são modificados pela execução das atividades. Todas as perspectivas são complementares e relevantes para a mineração de processos.

Ortogonalmente a essas três perspectivas (processos, organização e cases), o resultado de um esforço de mineração pode se referir às questões de ordem lógica

e/ou de desempenho. A mineração de processo pode, por exemplo, focalizar na questão de desempenho dos processos e analisar os tempos do fluxo das atividades, da utilização dos recursos e a frequência de execução.

2.8. Técnica de conformidade

A questão mais importante no contexto da análise de conformidade é se o processo de negócio real corresponde ao comportamento especificado, isto é, se o *log* de eventos corresponde ao modelo formal.

Segundo Medeiros e Weijters (2008), a técnica de mineração de processos por conformidade pode ser aplicada nas organizações para auxiliar os gerentes de processos a responderem questões relacionadas às três perspectivas (processos, organização e *cases*), entre elas:

- a) Qual é a compatibilidade dos *cases* (instâncias de processos) com os modelos de processos?
- b) Quais são as probabilidades de roteamento para cada atividade?
- c) Qual é o tempo médio/mínimo/máximo de atravessamento dos *cases*?
- d) Qual caminho toma mais tempo em média?
- e) Qual é o tempo médio para cada atividade?
- f) Quanto tempo é gasto entre duas atividades quaisquer no modelo de processo?
- g) Quais são as regras do negócio aplicáveis ao modelo de processo?

As respostas a essas questões podem ser obtidas com a utilização da ferramenta de mineração de processos denominada *framework* ProM.

A partir de um modelo de processo formalizado, *log* de eventos gerado de sistemas transacionais, e a utilização de ferramentas de mineração de processos, é possível verificar a conformidade entre os processos de negócio, conforme ilustra a figura 27.

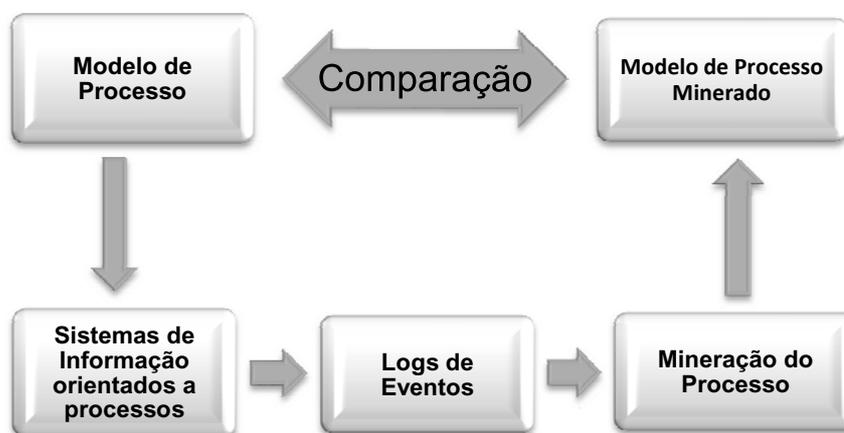


Figura 27 - Mineração de processo – técnica de conformidade

Fonte: Adaptado de Medeiros e Günther(2005).

2.8.1. Características

Para Rosinat e Aalst (2006), o *Fitness* é a característica mais dominante na técnica de conformidade, sendo possível quantificá-la numa escala de zero (pior caso) até um (melhor caso). Um bom *Fitness*, entretanto, não implica dizer que o modelo e o *log* de eventos possuem conformidade, tampouco significa ter obtido informações significativas sobre o mesmo. Ele deve ser analisado em conjunto com as demais dimensões de análise de conformidade.

Appropriateness é outra dimensão de análise desta técnica de mineração, abordada por Rosinat e Aalst (2006), que tenta capturar a idéia preconizada no princípio lógico denominado Navalha de Occam, centrada na idéia que o entendimento da existência de um fenômeno, deve levar em conta apenas as hipóteses necessárias à explicação do mesmo, eliminando todas as outras explicações que não causariam qualquer diferença aparente nas predições da hipótese ou teoria.

Esta dimensão, entretanto, não é facilmente quantificável como o *Fitness*. Rosinat e Aalst (2006) fazem uma distinção entre *Appropriateness* estrutural (se um modelo simples pode explicar um *log*, porque escolher um complexo) e *Appropriateness* comportamental (o modelo não pode ser tão genérico e permitir um comportamento diverso).

A figura 28 mostra a execução de uma rede de Petri, representando um modelo simplificado de processamento de uma reclamação de seguro. Este exemplo servirá de base para ilustrar o conceito de conformidade e detalhar os três conceitos básicos que são necessários para compreender a técnica de verificação de conformidade: um modelo baseado em redes de Petri; *logs* de eventos; e o mapeamento entre o modelo e o *log*.

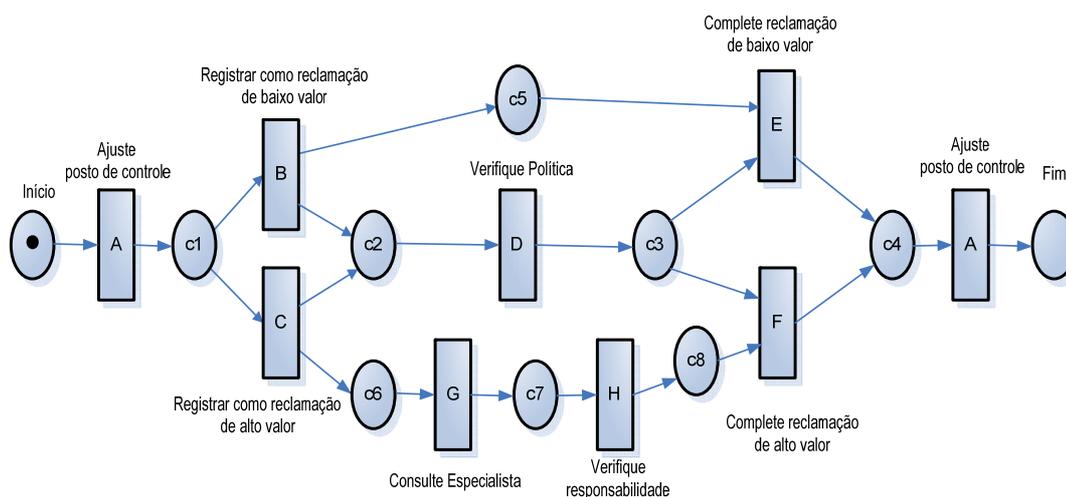


Figura 28 - Modelo M1 – Simplificação do processamento de uma reclamação de seguro

Fonte: Adaptado de Rosinat e Aalst (2006).

Na interpretação da rede de Petri mostrado na Figura 28, observa-se a existência de duas tarefas com a mesma etiqueta “Ajuste posto de controle” e que foi nomeada como transição A.

Pode-se imaginar que essa disposição corresponde a uma ação de *backup* automático dentro do contexto do sistema transacional, ou seja, a atividade A é executada no início para permitir um ponto de restauração, possibilitando atomicidade em todo o processo, e no final, para garantir durabilidade nos resultados.

Na sequência, o processo de negócio inicia-se com uma distinção entre reclamações de baixo e alto valor, que devem ser registradas de forma diferenciada (B ou C).

A política para o cliente é sempre verificada (D), mas em caso de reclamações de alto valor, adicionalmente um especialista é consultado (G) e assim, a responsabilidade da reclamação é verificada detalhadamente (H).

A realização das tarefas (E) e (F) podem ser tratadas como dois diferentes subprocessos envolvendo tomada de decisão e potencial pagamento, acontecendo

em outro departamento. Observe que as opções entre (E) e (F) são influenciadas pela escolha formal entre (B) e (C).

Esta é uma rede de Petri com as características de uma rede de *workflow* – WF Nets, onde processo é executado adequadamente, com propriedade de *soundness* e garantindo a ausência de *deadlocks* e *live-locks*.

2.8.2. Mapeamento

Rosinat e Aalst (2006) definem como pré-requisito para análise de conformidade, que as tarefas de um modelo de processo sejam associadas com o *log* de eventos, e recebam uma etiqueta denotando o tipo de *log* associado para cada tarefa do modelo.

Além do mapeamento simples de um-para-um, onde cada tarefa é associada exatamente com um tipo de evento no *log*, e nenhuma outra tarefa é associada com o mesmo tipo de evento, um mapeamento pode resultar nas seguintes construções:

- a) Tarefas Duplicadas: múltiplas tarefas no modelo são associadas com o mesmo tipo de *log* de eventos, o que significa que apesar de diferentes, sua ocorrência não pode ser distinguida no *log*. Observa-se que as tarefas duplicadas só podem aparecer no mapeamento, desde que as tarefas do modelo de processo sejam identificáveis (não por meio da etiqueta, mas por meio da identidade ou posição única no grafo). Na Figura 28, existem duas tarefas que receberam a mesma etiqueta “Ajuste do Posto de Controle”. Elas são exemplo de tarefas duplicadas;
- b) Tarefas Invisíveis: tarefas que não são executadas e, portanto não possuem nenhum evento associado no *log*. Isso acontece porque certos passos do processo podem não ser observados (exemplo: uma chamada telefônica). Tarefas invisíveis podem ser introduzidas com a finalidade de roteamento.

Observa-se que, apesar das atividades num cenário real de negócios serem freqüentemente executadas num nível mais depurado, no *log* é gravado informações relativas à programação, o início e a conclusão de uma atividade. Para manter a

abordagem analisada a mais universal possível, Rozinat e Aalst (2006) assumem que uma tarefa é geralmente associada a um tipo de *log* de evento. Considera-se também que antes de iniciar a análise, são excluídos todos os *log* de eventos que não estão associados com alguma tarefa.

A princípio, essa parece ser uma escolha razoável uma vez que o *log* pode estar em granularidade diferente do modelo analisado, isto é, conter eventos de baixo nível tais como de início, códigos de erro e mensagens de status do sistema.

2.8.3. As duas dimensões de conformidade

Para Rosinat e Aalst (2006), a questão mais importante no contexto da análise de conformidade é se o processo de negócio real corresponde ao comportamento especificado, isto é, se o *log* de eventos corresponde ao modelo.

Levando em conta o exemplo de modelo de processo M1 da Figura 28 e os *logs* de eventos mostrado na figura 29 a seguir, observa-se que o *log* de eventos L1 parece refletir melhor a realidade, uma vez que cada trilha do *log* pode ser associada com um caminho válido do Início ao Fim.

Em contraste, o *log* de eventos L2 não se aplica totalmente, uma vez que as trilhas ACHDFA e ACDHFA não contemplam a execução da atividade G; e o *log* L3 nem mesmo contém uma trilha que corresponda ao comportamento especificado. De alguma maneira, L3 parece ter um *Fitness* pior que L2.

(a) Log de evento L1		(b) Log de evento L2		(c) Log de evento L3	
No.de Instâncias	Trilhas do Log	No.de Instâncias	Trilhas do Log	No.de Instâncias	Trilhas do Log
4070	ABDEA	1207	ABDEA	24	BDE
245	ACDGHFA	145	ACDGHFA	7	AABHF
56	ACGDHFA	56	ACGDHFA	15	CHF
		23	ACHDFA	6	ADBE
		28	ACDHFA	1	ACBGDFAA
				8	ABEDA

Figura 29 - Três logs de eventos para o processo descrito na Figura 28

Fonte: Adaptado de Rosinat e Aalst (2006).

Existe outra dimensão de conformidade interessante e qualitativa, a qual pode ser ilustrada relacionando o *log* de evento L2 nos modelos de processos M2 e M3, mostrado na Figura 30 e Figura 31.

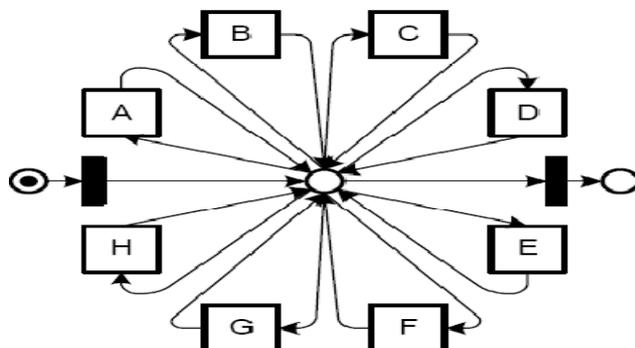


Figura 30 - Modelo de processo M2 - Flower

Fonte: Adaptado de Rosinat e Aalst (2006).

Apesar de o *log* ter um bom *Fitness* com relação a ambos os modelos, isto é, as sequências de eventos no *log* e no modelo são idênticas, eles não parecem adequados para descrever um processo de administração de reclamações de seguros.

O primeiro modelo M2 é genérico demais e abrange comportamento em excesso; permite sequências arbitrárias contendo as atividades A, B, C, D, E, F, G, ou H.

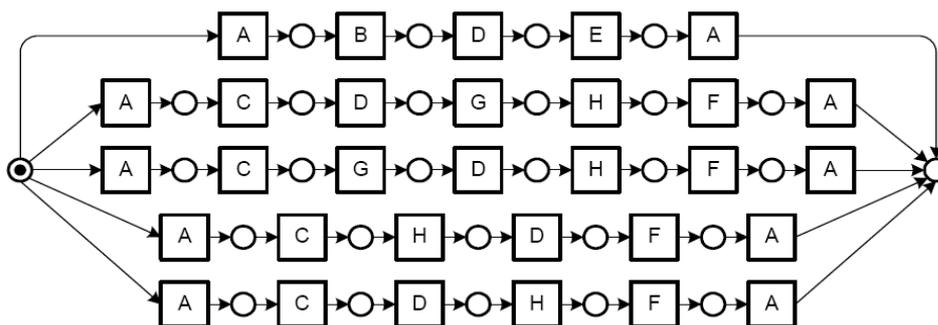


Figura 31 - Modelo de processo M3

Fonte: Adaptado de Rosinat e Aalst (2006).

O segundo modelo M3 não permite mais sequências além das observadas no *log*, apenas lista as sequências possíveis ao invés de expressar o comportamento especificado de forma significativa. Portanto, isto não oferece uma melhor compreensão da que poderia ser obtida apenas observando o *log* agregado.

Um “bom” modelo de processo deve de alguma forma ter uma estrutura mínima que reflita claramente o comportamento descrito, mais adiante referido como *Appropriateness* estrutural, e um comportamento mínimo para representar tão próximo quanto possível, o que realmente está acontecendo, a qual será chamada na sequência de *Appropriateness* comportamental.

As técnicas de análise de conformidade demonstradas nesta pesquisa permitem aos analistas de processos de negócios:

- a) Medir as duas dimensões de conformidade. Métricas são importantes para estimar o risco de desvios potenciais, e para comparar diferentes combinações *log*-modelo. Devem ser estáveis (não sejam afetadas por propriedades irrelevantes) e analisáveis (seja relacionada a alguma escala métrica);
- b) Localizar pontos potenciais de melhoria nos processos de negócio da organização. A localização de erros é crucial, e sem ela não é possível aprofundar dentro do problema e adotar potenciais ações de correção.

É importante observar que um problema de conformidade encontrado pode ser sempre visto por dois ângulos.

Primeiro, é que o modelo deve ser assumido como correto, porque ele representa a maneira como o processo de negócio deve ser executado. Portanto, pode, por exemplo, disparar ações no sentido de obter o comportamento desejado.

Segundo, o *log* de eventos pode ser admitido como correto, porque representa o que realmente ocorreu, e o modelo de processo pode estar desatualizado ou apenas não ter sido adequado as necessidades dos empregados que estão realmente executando as tarefas.

Deixar claro este ponto facilita o redesenho do modelo e desta forma aumenta a transparência. Em qualquer situação, uma interpretação final só deve ser dada por um especialista no assunto.

Mas mesmo no caso em que o modelo e o *log* estejam em conformidade, isto pode ser importante para alavancar a confiança no modelo de processo existente. A validação de modelo através da técnica de conformidade é um ponto de partida para outros tipos de análise.

2.8.4. Métrica *Fitness*

Rosinat e Aalst (2006) mencionam que uma forma de medir o ajuste entre os log de eventos e os modelos de processos é repetir o *log* no modelo e de algum modo medir o desvio.

A repetição de cada trilha lógica do *log* começa com a marcação de lugar inicial no modelo. Depois, as transições que pertencem aos eventos logados são disparadas uma após a outra.

Enquanto a repetição ocorre, conta-se o número de fichas que foram criadas artificialmente (isto é, a transação pertencente ao evento logado não foi habilitada e, portanto não pode ser executada com sucesso) e o número de fichas que foram deixadas no modelo, indicando que o processo não foi adequadamente completado.

O *Fitness* é dado pela expressão:

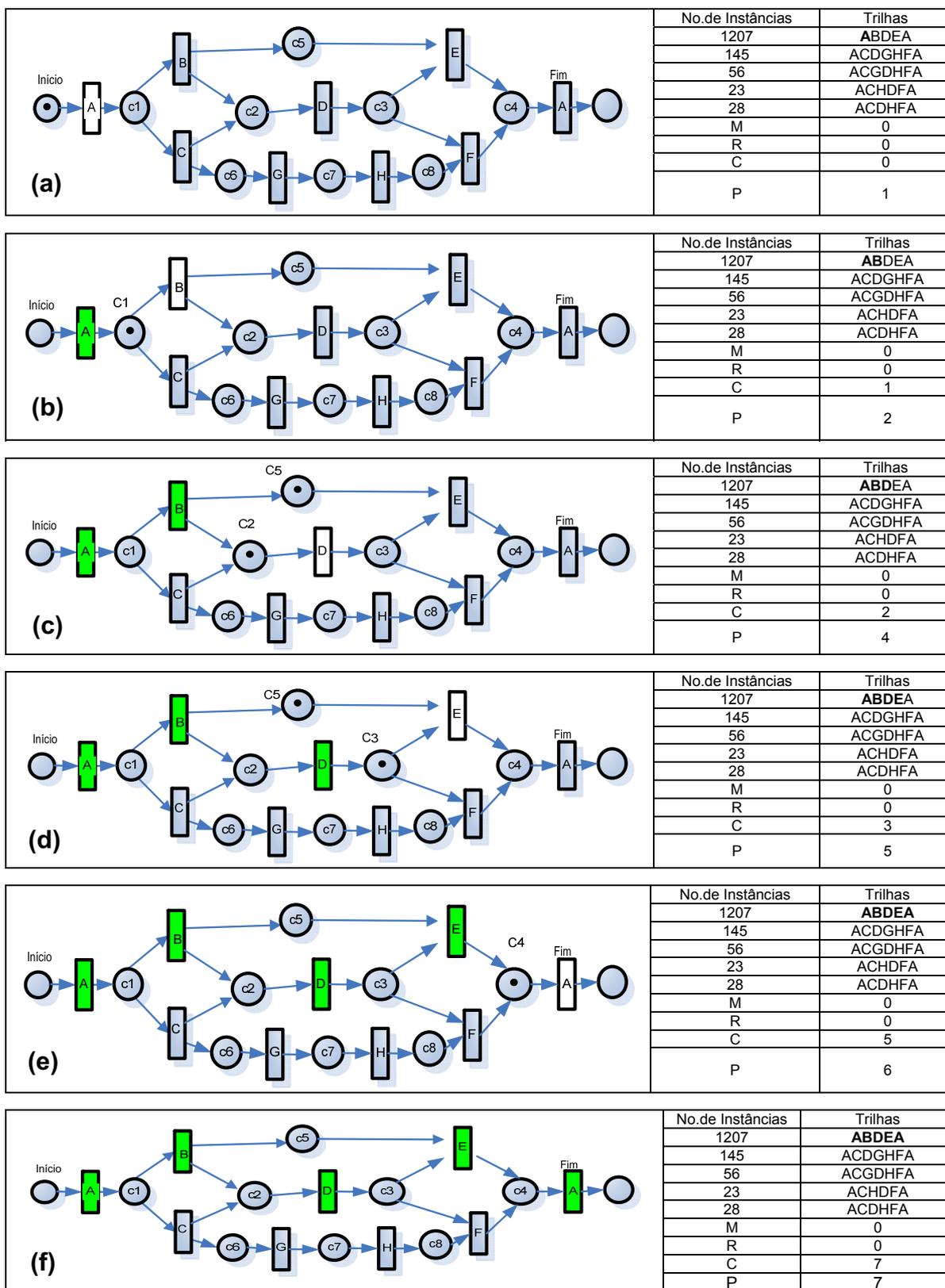
$$f = \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k \mathbf{nim}_i}{\sum_{i=1}^k \mathbf{nici}} \right) + \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k \mathbf{niri}}{\sum_{i=1}^k \mathbf{nipi}} \right)$$

onde:

- k é o número de diferentes trilhas no log
- i é a trilha do log
- n_i é a instância do processo combinada na trilha atual
- m_i é o número de fichas perdidas
- r_i é o número de fichas remanescentes
- c_i é o número de fichas consumidas
- p_i é o número de fichas produzidas

Rosinat e Aalst (2006) demonstram na Figura 32, um exemplo de repetição da primeira trilha do *log* L2 no modelo de processo M1. Em (a), uma ficha inicial é produzida para o lugar Início do modelo. Inicialmente, $m=0$ (sem fichas perdidas), $r=0$ (sem fichas remanescentes), $c=0$ (nenhuma ficha utilizada), e $p=1$ (antes da execução da transição A, uma ficha é colocada no lugar Início).

O primeiro evento do *log* na trilha, A é associado com duas transições no modelo, cada uma com a etiqueta A. Mas somente uma delas é habilitada e assim irá disparar (b), consumindo uma ficha do Início e produzindo uma ficha para o lugar C1 ($c=1$, $p=2$).

Figura 32 - Repetição de L2 e trilha $i=1$ no modelo M1

Fonte: Adaptado de Rosinat e Aalst (2006).

No próximo evento do *log*, a correspondente transição B é habilitada e pode disparar (c), consumindo uma ficha de c1 e produzindo uma ficha para os lugares C2 e C5 (c=2, p=4). Assim, o seguinte evento corresponde a transição D, que é habilitada e pode disparar (d), consumindo uma ficha de C2 e produzindo uma ficha para C3 (c=3, p=5).

Similarmente, a transição associada ao próximo evento E, é também habilitada e dispara (e), consumindo uma ficha de C3 e C5, e produzindo uma ficha para C4 (c=5, p=6).

Finalmente, o último evento do *log* é do tipo A novamente, isto é, associada com as duas transições A do modelo, mas como somente uma é habilitada e, portanto escolhida para disparar (f), consumindo a ficha de C4 e produzindo uma ficha no lugar Fim (c=6, p=7). Como último passo, esta ficha no lugar Fim é consumida (c=7) e a repetição para aquela trilha está completa.

Como não existem fichas perdidas ou remanescentes (m=0 e r=0), esta trilha representa (*Fit*) perfeitamente o modelo M1. Similarmente, a segunda e terceira trilha também podem ser repetidas sem quaisquer problemas, isto é, nenhuma ficha fica para trás ou é perdida (m2 = m3 = r2 = r3 = 0).

Agora considere a Figura 33, que representa a repetição da quarta trilha do *log* de evento L2 no modelo de processo M1.

No início (a) e (b) o procedimento é bastante similar, só que ao invés da transição B, a transição C é disparada (c) consumindo uma ficha de C1 e produzindo uma ficha para C2 e C6 c=2, p=4).

Mas quando a próxima sequência de evento é repetida, a correspondente transição H não é habilitada. Conseqüentemente, a ficha em C7 é artificialmente criada e registrada como perdida (m=1). Assim, a transição H é disparada (d), consumindo a ficha e produzindo uma ficha para o lugar C8 (c=3, p=5).

A sequência seguinte pode ser repetida com sucesso, isto é, suas transições associadas estão habilitadas e podem ser disparadas: (e) transição D consumindo uma ficha de C2 e produzindo uma ficha para C3 (c=4, p=6), (f) transição F consumindo uma ficha de C8 e C3 e produzindo uma ficha para C4 (c=6, p=7), (g) uma das duas transições associadas A está habilitada e pode ser disparada, consumindo uma ficha de C4 e produzindo uma ficha no lugar Fim (c=7, p=8).

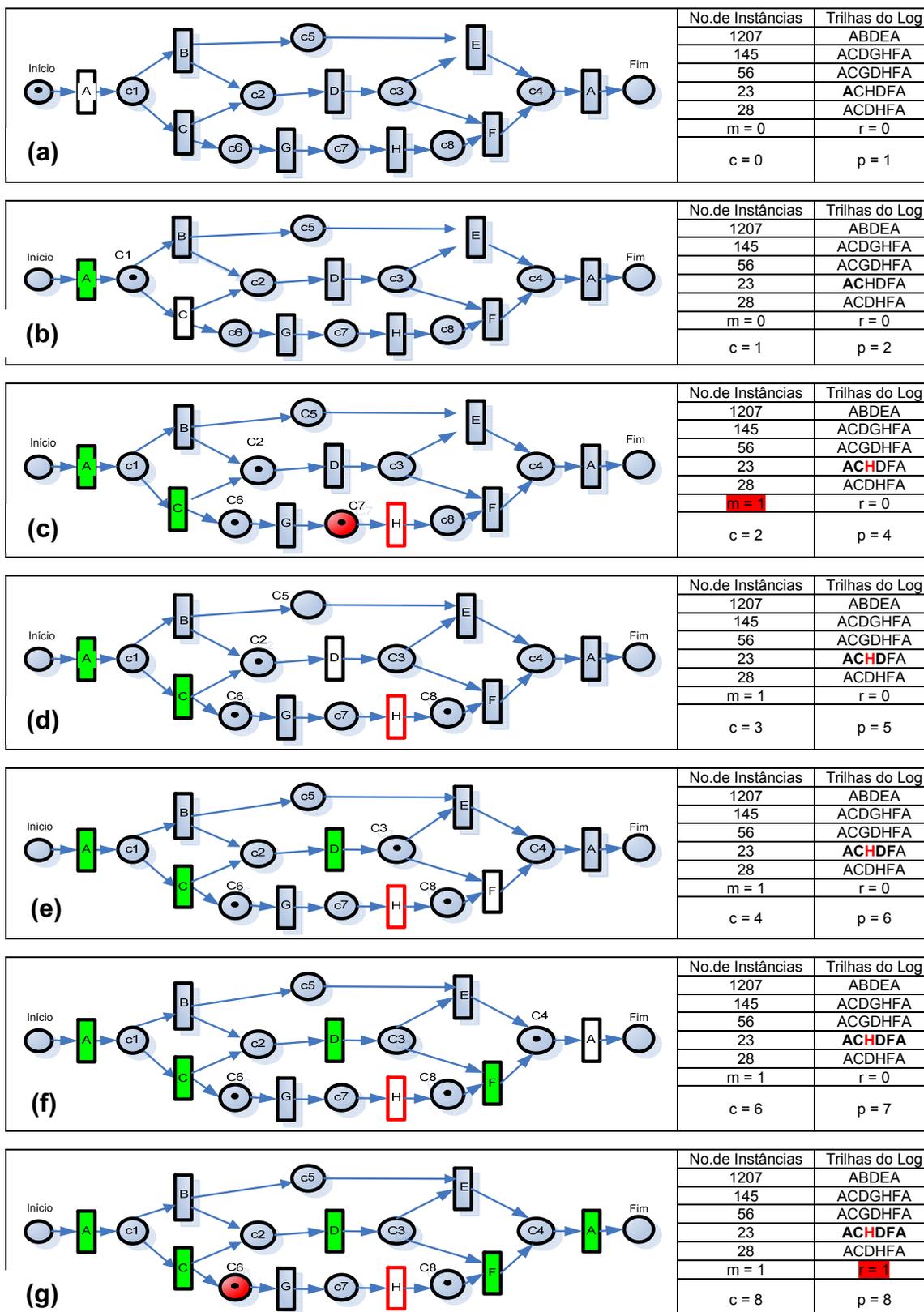


Figura 33 - Repetição de L2 para trilha $i=4$ no modelo de processo M1

Fonte: Adaptado de Rosinat e Aalst (2006).

Por último, a ficha do lugar Fim é consumida novamente ($c=8$). Mas ainda existe uma ficha remanescente no lugar C6, que será penalizado indicando que o

processo não foi executado apropriadamente ($r=1$). Um problema similar será encontrado durante a repetição da última trilha do *log* de evento L2 ($r_5=1$, $m_5=1$).

Usando a métrica f , pode-se calcular o *Fitness* entre todo o *log* de evento L2 e a descrição do processo M1. Como citado anteriormente, além da trilha $i=4$, existiram fichas perdidas e remanescentes na última trilha $i=5$. Contando também o número de fichas que foram produzidas e consumidas enquanto as outras três trilhas eram repetidas ($c_2 = c_3 = p_2 = p_3 = 9$, e $c_5 = p_5 = 8$), e com o número dado de instâncias de processo por trilha, o *Fitness* pode ser medido como:

$$f(M1, L2) = \frac{1}{2} \left(1 - \frac{51}{10666} \right) + \frac{1}{2} \left(\frac{51}{10666} \right) \approx 0,995$$

De maneira análoga, pode-se calcular o *Fitness* entre os *logs* de eventos L1, L3, e o processo descritivo M1. O primeiro *log* de eventos L1 contém somente três trilhas que foram ajustadas com L2.

Assim, não existe nem ficha perdida ou remanescente no modelo durante a repetição do *log* e a métrica $f(M1, L1) = 1$. Em contraste, para o último *log* de evento L3, nenhuma trilha pode ser associada com uma sequência de disparo válida de rede de Petri, e cerca de metade das fichas produzidas e consumidas ficaram perdidas ou remanescentes, o que implica numa métrica de *Fitness* $f(M1, L3) \approx 0,540$.

A localização aproximada do desvio fornece ao analista uma ajuda importante, uma vez que o local das fichas remanescentes e perdidas durante a repetição do *log* proporcionam informações valiosas a respeito do *Fitness*.

A figura 34 demonstra o modelo de processo M1 após a repetição do *log* de evento L2. As fichas remanescentes (indicadas pelo sinal +) no lugar C6 deixam a transição G habilitada, e assim como havia fichas perdidas (indicadas pelo sinal -) no lugar C7, a transição H disparou, retratando um comportamento em desacordo com o modelo.

Conforme mencionado anteriormente, uma interpretação final sobre essa divergência só pode ser dada por um especialista do processo de seguros. Entretanto, a primeira vista, parece que o modelo não contempla a possibilidade de desvio da atividade G.

Isso é demonstrado pelo diagnóstico nos contadores de fichas mostrado na figura 34, que indica a localização dos erros e fornecem informações sobre a localização dos erros. A marca “+51” no lugar C6 indica 51 fichas remanescentes ($r = 51$), e a marca “-51” indica que a execução falhou 51 vezes ($m = 51$).

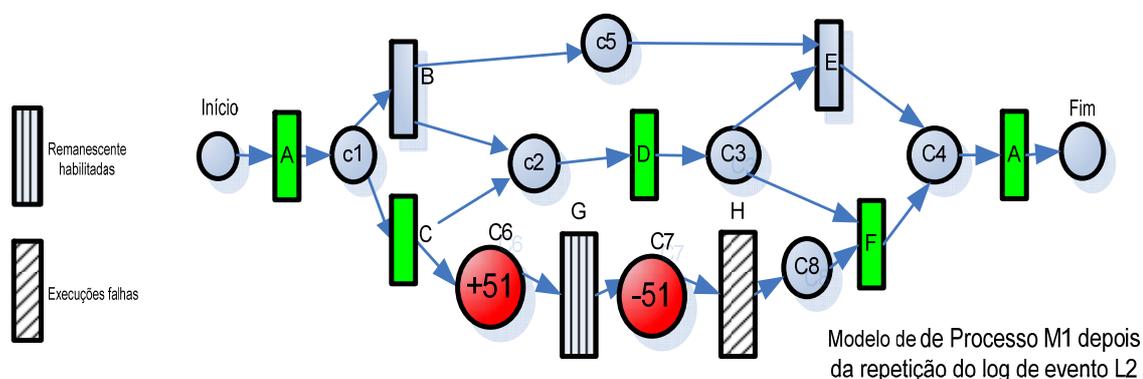


Figura 34 - Diagnóstico nos contadores de fichas

Fonte: Adaptado de Rosinat e Aalst (2006).

Observa-se que a repetição do *log* de evento L2 é realizada de forma não restritiva e baseada na perspectiva do mesmo, isto é, as transações correspondentes a cada trilha do *log* de eventos é disparada, indiferentemente se o caminho atual do modelo é seguido ou não.

Isso conduz ao fato que em contraste com comparações diretas de fluxos de eventos dos modelos e dos *logs*, a série de *log* de eventos perdidos é penalizada com uma métrica de “*Fitness*” f equivalente a 1, uma vez que isso pode ser sempre interpretado como uma ligação perdida no modelo.

Tarefas duplicadas não causam problemas durante a repetição dos *logs*, na medida em que apenas uma delas é executada a cada vez (conforme mostrado na figura 32 e 33) para as duas tarefas identificadas como “A”, mas, de outro modo uma pode habilitar e/ou disparar a tarefa adequada para a correta progressão. As tarefas invisíveis somente disparam se elas podem habilitar a transição em questão.

2.8.5. Métrica *Appropriateness*

Nesta análise, segundo Rosinat e Aalst (2006), uma maneira de resolver o problema visualizado na Figura 34 seria adaptar o modelo de processo ao processo

que realmente acontece (com base no comportamento observado no *log* de eventos), e inserir atividades invisíveis que permitem o desvio da atividade G.

A figura 35 mostra o modelo de processo modificado M4, que agora é 100% compatível com o *log* de eventos L2.

Apesar dos modelos M2 e M3 mostrados na Figuras 30 e 31 também serem 100% compatíveis com o *log* de eventos L2, em relação ao comportamento previsto no *log* de eventos, o modelo M4 parece ser mais apropriado do ponto de vista comportamental (o modelo M2 permite outros comportamentos) e do ponto de vista estrutural (é mais compacto e claramente reflete o comportamento observado no *log* de eventos L2, ao invés de apenas listar possíveis sequências como em M3).

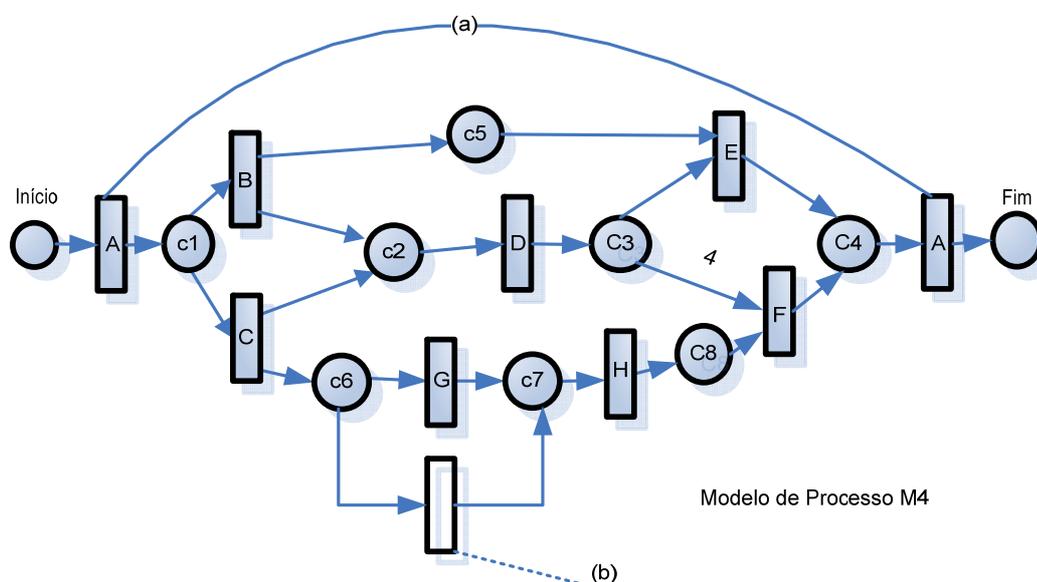


Figura 35 - Modelo M4 - 100% compatível com o log de evento L2

Fonte: Adaptado de Rosinat e Aalst (2006).

2.8.5.1. *Appropriateness* comportamental

Enquanto o *Fitness* analisa a correspondência de cada registro do *log* de eventos com relação ao modelo do processo, a métrica *Appropriateness* Comportamental compara o quanto do comportamento permitido pelo modelo nunca foi observado nas execuções de processo do *log*.

A idéia defendida por Rosinat e Aalst (2006), é que a modelagem do processo seja precisa. Quando o modelo se torna muito genérico e permite realizar um comportamento maior que o necessário (como mostrado no modelo M2), então ele se torna menos informativo, uma vez que não descreve o processo atual, e pode permitir a execução de sequências indesejadas.

O modelo de processo M5 mostrado na Figura 36 é também compatível com o *log* L2 e a atividade H pode ser executada sem a prévia execução da atividade G. Porém, permite repetições arbitrárias da atividade G - Consulta a um Especialista, o que pode não ser desejado.

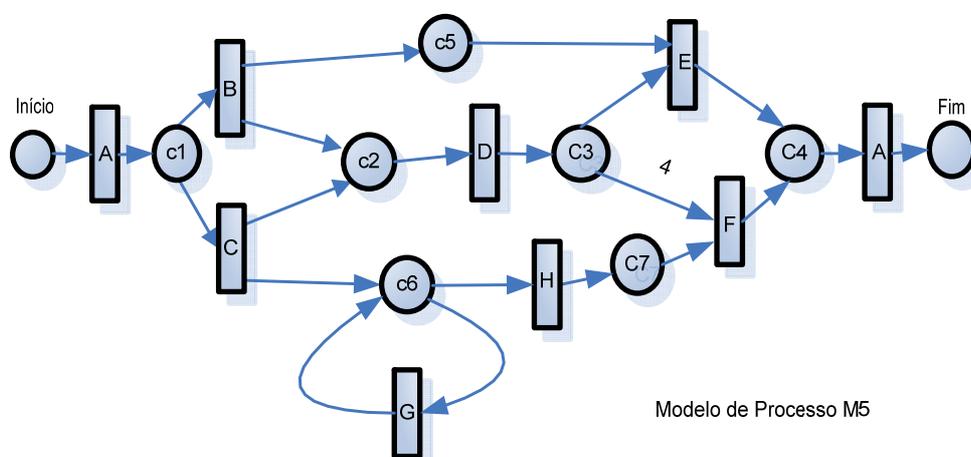


Figura 36 - Modelo M5 que permite o desvio da atividade G

Fonte: Adaptado de Rosinat e Aalst (2006).

Neste exemplo simples, este problema é quase óbvio, mas em modelos de processo mais complexos, isso pode ser difícil de observar. Denota-se em termos práticos que o problema de conformidade percebido pode ser visto de dois ângulos.

Primeiro, o comportamento extra permitido pelo modelo pode corresponder, por exemplo, a uma opção alternativa que trata com situações excepcionais que não ocorrem dentro da janela de tempo em que o *log* de eventos foi gravado. Neste caso, o *log* não é completo.

Segundo, o modelo pode ser realmente muito genérico e permitir situações que não ocorrem na realidade. Neste caso só um especialista da área será capaz de diferenciar estas duas situações, portanto, métricas adequadas e satisfatórias são necessárias.

2.8.5.2. *Appropriateness* comportamental simples

Uma primeira abordagem para medir a quantidade possível de comportamento é determinar o número médio de transições habilitadas durante a repetição do *log* de eventos. Isso corresponde à idéia de que um aumento no número de alternativas ou paralelismo e, portanto, um aumento no potencial de comportamento, irá resultar em um grande número de transições habilitadas durante a repetição do *log* de eventos.

O comportamento estrutural simples é dado pela expressão:

$$aB = \frac{\sum_{i=1}^k ni (|Tv| - xi)}{(|Tv| - 1) \cdot \sum_{i=1}^k ni}$$

onde:

- k é o número de diferentes trilhas no log
- i é a trilha do log
- ni é a instância do processo combinada na trilha atual
- Tv é o conjunto de tarefas visíveis no modelo

Considerando que ($|Tv| > 1$), esta métrica varia de 0 (se todas as atividades visíveis no modelo são sempre habilitadas durante a repetição do *log*, tal como mostrado no modelo M2) até 1 (quando for um processo sequencial, onde há 100% de *fitness* e, portanto, existe no mínimo uma transição habilitada durante a repetição do *log*).

Se calcularmos a *Appropriateness* comportamental para M4, a métrica aB (M4, L2) $\approx 0,967$, que é ligeiramente maior que o valor para o modelo, que permite ciclos arbitrários da atividade G (M5), cuja métrica aB(M5, L2) $\approx 0,964$.

Entretanto, existe o problema que a métrica só pode ser usada de forma comparativa, porque mede a *Appropriateness* relativamente ao grau de flexibilidade do modelo. Assim, o modelo M4 é melhor que o modelo M5, porque quanto menos comportamento permitido ao modelo, melhor.

Essa métrica só atinge o valor 1 num modelo puramente sequencial, onde exatamente uma tarefa é habilitada a cada passo de repetição do *log*. Adicionalmente, a métrica não é estável para situações onde o modelo é ordenado em sequência através de tarefas duplicadas, como mostrado no modelo de processo M3 da figura 31.

Para abordar a *Appropriateness* comportamental independentemente de tal propriedade estrutural e independentemente da flexibilidade do modelo, o comportamento potencial especificado pelo modelo deve ser analisado e comparado com o comportamento realmente necessário para descrever o que foi observado no *log*.

A noção de conjunto de identificadores que servem de conexão entre as tarefas do modelo, e os elementos contidos no *log* torna-se possível derivar relações comparáveis de “*Follow* - Seguir” e “*Precedes* - Preceder”, entre as atividades sob a perspectiva do modelo e do *log*.

Para reduzir as exigências de perfeição do *log* de eventos, e também capturar as dependências mais distantes entre as atividades, a relação “*Follow*” e “*Precedes*” é determinada globalmente, isto é, as tarefas, ou *log* de eventos, não precisam diretamente seguir ou preceder umas as outras.

Assim, ao observar um conjunto de sequências, pode-se determinar se duas atividades (x, y) sempre, nunca ou de vez em quando, seguem ou precedem umas as outras:

- a) Definição de “*Follow*” - duas atividades (x,y) estão em uma relação de “Sempre Seguem”, “Nunca Seguem”, ou “As Vezes Seguem”, no caso que, se x for executado pelo menos uma vez, depois sempre, nunca, ou algumas vezes y é também eventualmente executado respectivamente;
- b) Definição de “*Precedes*” - duas atividades (x,y) estão em uma relação de “Sempre Precedem”, “Nunca Precedem”, ou “As Vezes Precedem”, no caso que, se y for executado pelo menos uma vez, depois sempre, nunca, ou algumas vezes x foi executado algum tempo antes, respectivamente.

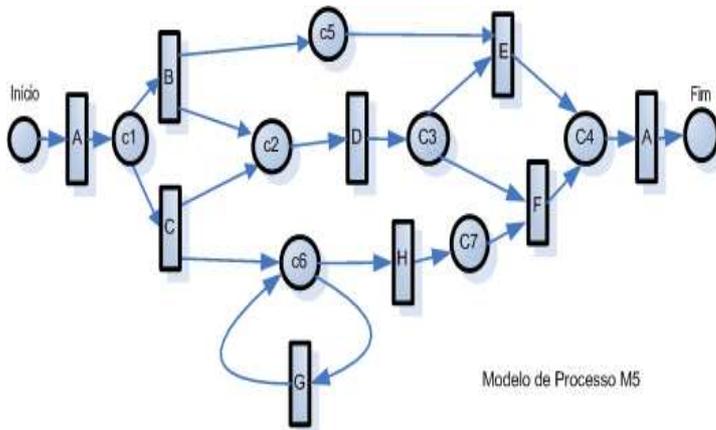
Considerando a figura 37, que demonstra a relação global “*Follow*” que são derivadas do modelo M5 e *log* de eventos L2. Para construir essas relações a partir de uma perspectiva do modelo, analisam-se as possíveis sequências de execuções.

Da perspectiva do *log*, analisam-se as sequências de execuções observadas. A partir daí, pode-se determinar se duas atividades (x,y) seguem Sempre, Nunca, ou Às Vezes umas às outras.

O mesmo pode ser feito para as relações globais “*Precedes*”. Na figura 37 pode-se ver que enquanto a atividade G do modelo M5 pode ser seguida pela atividade G, isto é, (G,G) é também um elemento da relação “Algumas Vezes

Segue”. O fato é que isso nunca acontece no *log* de evento L2, ou seja, nessa condição (G,G) é parte da relação “Nunca Segue”.

Observa-se que o número de caminhos no modelo é maior que o número de trilhas que de fato aparecem no *log*. Portanto, derivar as relações a partir do modelo pode ser problemático, ao contrário do *log*, que não traz nenhum problema.



Log de Eventos L2	
No. de instâncias	Trilhas de Log
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

Análise se as atividades no modelo Sempre (A), Nunca (N), ou Às Vezes (S) seguem uma a outra

Análise se os eventos no log, de fato Sempre (A), Nunca (N), ou Às Vezes (S) seguem um ao outro

F	A	B	C	D	E	F	G	H
A	A	S	S	A	S	S	S	S
B	A	N	N	A	A	N	N	N
C	A	N	N	A	N	A	S	A
D	A	N	N	N	S	S	S	S
E	A	N	N	N	N	N	N	N
F	A	N	N	N	N	N	N	N
G	A	N	N	S	N	A	S	A
H	A	N	N	S	N	A	N	N

(a) Relação "Follow" baseada na perspectiva do modelo.

F	A	B	C	D	E	F	G	H
A	A	S	S	A	S	S	S	S
B	A	N	N	A	A	N	N	N
C	A	N	N	A	N	A	S	A
D	A	N	N	N	S	S	S	S
E	A	N	N	N	N	N	N	N
F	A	N	N	N	N	N	N	N
G	A	N	N	S	N	A	N	A
H	A	N	N	S	N	A	N	N

(b) Relação "Follow" baseada na perspectiva do log.

Figura 37 - Relações globais "Follow" derivadas para o modelo M5 e Log L2

Fonte: Adaptado de Rosinat e Aalst (2006).

Enquanto as relações “Sempre” e “Nunca” descrevem fortes limitações, as relações “As vezes” capturam as variabilidades no comportamento. Por exemplo, atividades concorrentes podem seguir ou preceder uma à outra em qualquer ordem, ver (D,H) e (H,D) mostradas na Figura 37.

Similarmente, atividades precedendo um número de ramos alternativos são às vezes seguidos de um destes ramos alternativos e algumas vezes por outros, ver (A,B) e (A,C) na mesma figura. O mesmo vale para atividades que seguem depois

um número de ramos alternativos que se uniram (refletidos nas relações “Algumas Vezes Precedem”).

Portanto, a idéia da seguinte métrica é comparar a variabilidade do comportamento fornecido pelo modelo e o comportamento observado no *log*, baseado em número cardinais das relações Sf (algumas vezes seguem) e Sp (algumas vezes precedem).

2.8.5.3. *Appropriateness* comportamental avançado

De acordo com Rosinat e Aalst (2006), nesta métrica, devido a uma razão bastante técnica, o conjunto de identificadores considerados para formar essas relações, inclui a tarefa ou *log* de eventos artificialmente inseridas, de Início e Fim, respectivamente.

$$\alpha'B = \frac{|S_F^l \cap S_F^m|}{2 \cdot |S_F^m|} + \frac{|S_P^l \cap S_P^m|}{2 \cdot |S_P^m|}$$

onde:

- S_F^l é a relação Sf (algumas vezes segue) do *log*
- S_P^l é a relação Sp (algumas vezes precede) do *log*
- S_F^m é a relação Sf (algumas vezes segue) do modelo
- S_P^m é a relação Sp (algumas vezes precede) do modelo

Observa-se mais adiante que foi construído uma interseção de S_F^l e S_F^m (e S_P^l e S_P^m) para visualizar somente onde o *log* se torna mais específico, isto é, foi capturado as situações onde de acordo com o modelo, duas atividades podem as vezes seguir uma a outra (e as vezes não), mas no *log* elas sempre ou nunca seguem umas as outras.

O inverso pode também acontecer, isto é, o modelo é mais específico que o *log*, o que indica um problema de *Fitness*. Entretanto, uma vez descartadas estas sequências, os valores assinalados por $\alpha'B$ variam de 0 até 1.

Verifica-se finalmente que apesar das relações S_f e S_p serem simétricas, considera-se ambos e pondera-se igualmente no sentido de estabilizar a métrica com relação a posição do “comportamento extra”.

Ao calcular esta nova métrica de *Appropriateness* Comportamental para o modelo M4, tem-se que $\alpha'B$ (M4, L2) = 1, o que indica que o modelo M4 reproduz precisamente o comportamento que foi observado no *log* de eventos L2. A mesma métrica aplicada no modelo M5 resulta em $\alpha'B$ (M5,L2) = (19/2.20 + 20/2.21) \approx 0,951). As relações “Às Vezes” que são derivadas do modelo contem um elemento a mais que as relações “Às Vezes” que são derivadas do *log*, que é o elemento (G,G) (de acordo com a atividade G do modelo que pode ser seguida ou precedida por ela mesma).

Finalmente, calculando o valor para o modelo M2, resulta em $\alpha'B$ (M2,L2) \approx 0,271. Verifica-se que, por causa da nova métrica, a distância real entre as relações do modelo e o *log* é considerada, assim, o modelo genérico - *Flower* pode ser um bom modelo, nomeadamente se o *log* de evento exibe um comportamento aleatório.

Ao construir a noção de relacionamentos globais sucessores e predecessores, é possível destacar alternativas não utilizadas e partes concorrentes no modelo, que podem ser visualizadas conforme indicado na figura 38.

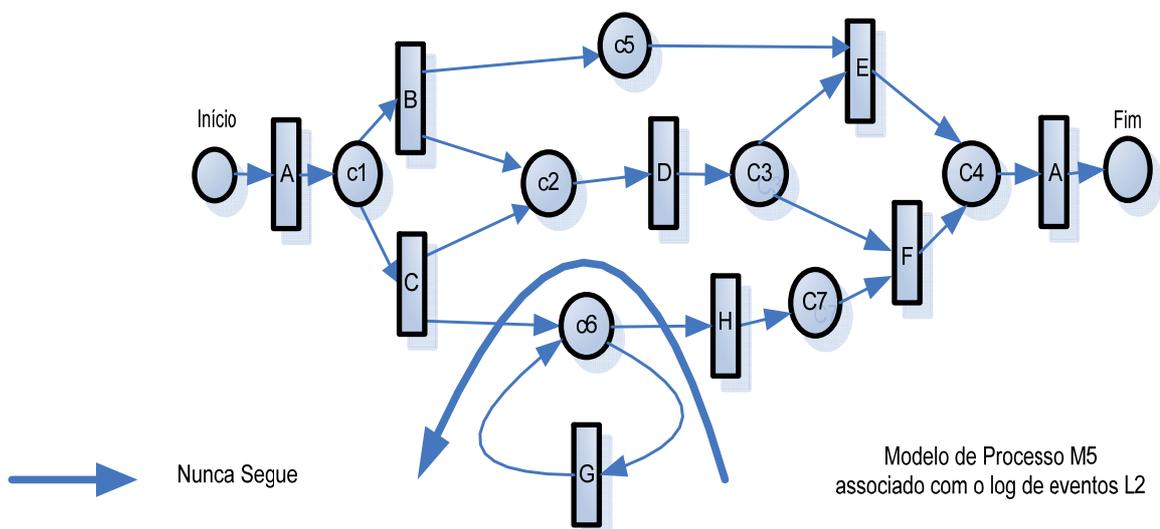


Figura 38 - Diferenças entre atividades sucessoras e predecessoras

Fonte: Adaptado de Rosinat e Aalst (2006).

2.8.5.4. *Appropriateness* estrutural

Para Rosinat e Aalst (2006), a tarefa de modelar um processo de negócio de forma compacta e significativa é difícil de obter por medidas. A aceitação de um modelo como satisfatório depende de preferências subjetivas, e tipicamente relacionadas com propósitos específicos do modelo.

Existem aspectos como, por exemplo, a granularidade das ações do *workflow* descrito, a qual só pode ser determinada por um experiente projetista. Mas a noção da *Appropriateness* Estrutural aqui apresentada é relativa à perspectiva de controle de fluxo, e muitas vezes, num modelo de processo existem diferentes maneiras sintáticas de expressar o mesmo comportamento. Por exemplo, o modelo M6 na figura 39, que permite o mesmo comportamento do modelo M4 e M3. Entretanto, ele contém a seguinte construção que pode “inflar” a estrutura do modelo de processo.

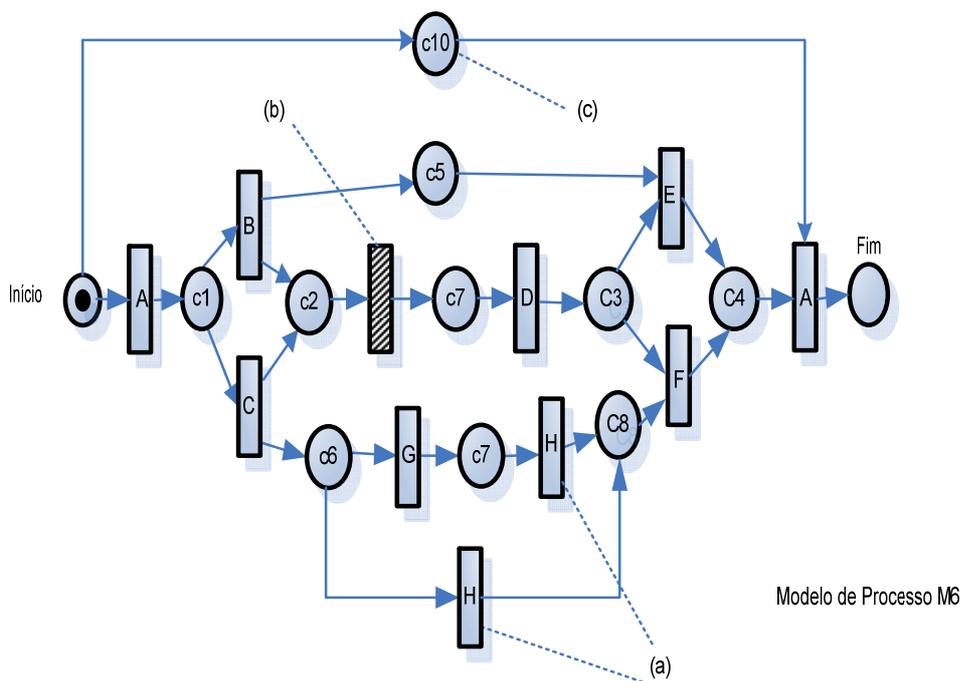


Figura 39 - Modelo contendo construções que podem “inflar” sua estrutura

Fonte: Adaptado de Rosinat e Aalst (2006).

Essas construções que podem tornar o modelo menos compacto e compreensível são compostas de:

- a. Tarefas duplicadas: além das tarefas duplicadas que são necessárias para especificar que certas atividades aconteçam num contexto completamente diferente, tal como a atividade A no início e fim do processo M4 (ver (a) na figura 35), existem também atividades duplicadas que podem ser agrupadas na medida em que seus diferentes contextos são capturados no modelo. Por exemplo, no modelo M6 a duplicação da tarefa H é empregada para demonstrar que depois de executar a atividade C, tanto a sequência GH ou H sozinha pode ser executada (ver (a) na figura 39. Na figura 35 (modelo de processo M4) descreve o mesmo processo com a ajuda de uma atividade invisível (ver (b) na figura 35). Tarefas duplicadas podem reduzir a *Appropriateness* Estrutural do modelo, pois elas previnem abstrações (que no modelo não podem ser vistas mais de duas tarefas que realmente são as mesmas). O modelo M3 mostra o caso extremo de um processo, numa visão completamente baseada em instâncias, com várias tarefas duplicadas e supérfluas;
- b. Tarefas invisíveis: além das tarefas invisíveis utilizadas para propósitos de roteamento, conforme indicado em (b) da figura 35, existem também tarefas invisíveis que retardam tarefas visíveis, tal como mostrado em (b) da figura 39. Se elas não servem para qualquer outro propósito, elas podem ser simplesmente removidas, assim tornando o modelo mais conciso;
- c. Lugares implícitos: são lugares que podem ser removidos sem alterar o comportamento do modelo, conforme exemplificado pelo lugar c10 (ver (c) na figura 39. Observa-se que o lugar c5 da figura 39 não é um lugar implícito, na medida em que influencia a escolha feita depois entre E e F. Ambos os lugares c5 e c10 são lugares silenciosos, isto é, para o modelo M4 não é possível produzir uma sequência de eventos contendo BE ou AA. As técnicas de mineração de processos por descoberta, por definição não são capazes de detectar lugares implícitos e tem problemas em detectar lugares silenciosos.

Observa-se que estas construções são somente um indicador de potenciais problemas de conformidade. Por exemplo, pode muito bem existir situações na qual o projetista de modelagem acha mais conveniente modelar utilizando tarefas duplicadas, mas isso deve ser evitado.

Como um primeiro indicador para a *Appropriateness* Estrutural, define-se uma métrica simples baseada no número de diferentes identificadores de tarefas em relação ao tamanho do grafo do modelo.

2.8.5.5. *Appropriateness* estrutural simples

Rosinat e Aalst (2006) consideram L o número de identificadores que estabelecem o mapeamento entre as tarefas do modelo e os eventos no *log*, e N o número de nós, representado em um modelo de rede Petri por lugares e transições. A métrica *Appropriateness* Estrutural Simples é dada pela equação:

$$aS = \frac{|L| + 2}{N}$$

Em toda rede modelada é esperado a existência de dois lugares dedicados: Início e Fim, ou seja, qualquer grafo deve conter pelo menos uma transição para cada tarefa, mais dois lugares (Início e Fim).

Neste caso, $|N| = |L| + 2$ e a métrica aS resulta no valor 1. À medida que o tamanho do grafo aumenta, devido a lugares adicionais, o valor encontrado aproxima-se de 0. Ao calcular a *Appropriateness* Estrutural para o modelo M3, tem-se que $as(M3) \approx 0,170$ que é um valor muito ruim e é causado pelas tarefas duplicadas, na medida em que aumenta o número de transições com o mesmo identificador.

Para o modelo M4, esta métrica resulta em $aS(M4) \approx 0,5$. Um valor ligeiramente menor é encontrado para $aS(M6) \approx 0,435$. Contudo, esta métrica só pode ser usada como uma forma de comparação de modelos de processos que exibem comportamentos semelhantes, desta forma, tem aplicabilidade limitada.

Para abordar a *Appropriateness* Estrutural independentemente do comportamento real do modelo, uma boa idéia é verificar certas diretrizes de projeto, que define a maneira preferencial para expressar um padrão de comportamento específico, e de alguma maneira punir violações dessas diretrizes. É óbvio que as

diretrizes de projeto irão variar para as diferentes notações de modelagem de projetos e podem depender de preferências pessoais ou corporativas.

Não obstante, a seguir é apresentada uma nova abordagem *Appropriateness* Estrutural baseada nos resultados encontrados e mencionados anteriormente.

Como uma diretriz de projetos, construções tais como tarefas duplicadas alternativas (tarefas duplicadas que nunca ocorrem juntas em um sequência de execução) e tarefas invisíveis redundantes (tarefas invisíveis que podem ser removidas do modelo sem alteração no comportamento) devem ser evitadas, uma vez que elas congestionam a estrutura do modelo e prejudicam a clareza com que o comportamento é refletido.

Observa-se que devido ao número de caminhos no modelo poder se tornar elevado, o custo de detecção de tarefas duplicadas alternativas pode ser problemático. Em contraste, tarefas invisíveis redundantes podem ser detectadas através de análise estrutural do modelo, o que é normalmente mais rápido.

2.8.5.6. *Appropriateness* estrutural avançada

Da mesma maneira, Rosinat e Aalst (2006) consideram T o conjunto de transições em um modelo de rede de Petri, T_{DA} , o conjunto de tarefas duplicadas alternativas, T_{IR} o conjunto de tarefas invisíveis redundantes.

A *Appropriateness* Estrutural Avançada é definida por:

$$a'S = \frac{|T| - (|TDA| + |TIR|)}{|T|}$$

Dado que as tarefas duplicadas estão sempre visíveis, $|TDA| + |TIR| \leq T$, portanto esta métrica varia de $0 \leq a'S \leq 1$. Revendo os modelos utilizados como exemplos, torna-se claro que de acordo com as diretrizes de projeto definidas, somente o modelo M6 e M3 possuem *Appropriateness* Estrutural. Para o modelo M6 o número de tarefas duplicadas alternativas $TDA = 2$ e o número de tarefas invisíveis redundantes $TIR = 1$, o que resulta em $a'S (M5) \approx 0,727$. No modelo M3 as tarefas A,

B e E pertencem ao conjunto de tarefas duplicadas alternativas, resultando em $a'S(M3) \approx 0,387$.

2.8.6. Combinando *Fitness* e *Appropriateness*

As noções de conformidade apresentadas por Rozinat e Aalst (2006) tais como *Fitness*, *Appropriateness* comportamental e *Appropriateness* estrutural são ortogonais entre si. Elas medem algo completamente distinto, portanto, melhorias apontadas por uma métrica não são comparáveis com as melhorias apontadas pelas outras.

A tabela 1 contém os valores medidos de *Fitness*, *Appropriateness* comportamental e *Appropriateness* estrutural para todas as combinações de modelos e *logs* de exemplo utilizados anteriormente.

	M1	M2	M3	M4	M5	M6
L 1	$f=1,0$ $aB = 0,9740$ $a'B = 0,9167$ $aS = 0,5263$ $a's = 1,0$	$f=1,0$ $aB = 0,0$ $a'B = 0,2292$ $aS = 0,7692$ $a's = 1,0$	$f=1,0$ $aB = 0,9739$ $a'B = 0,8474$ $aS = 0,1695$ $a's = 0,3871$	$f=1,0$ $aB = 0,9718$ $a'B = 0,8474$ $aS = 0,5$ $a's = 1,0$	$f=1,0$ $aB = 0,9703$ $a'B = 0,8060$ $aS = 0,5556$ $a's = 1,0$	$f=1,0$ $aB = 0,9749$ $a'B = 0,8474$ $aS = 0,4348$ $a's = 0,7273$
L 2	$f=0,9952$ $aB = 0,9705$ $a'B = 1,0$ $aS = 0,5263$ $a's = 1,0$	$f=1,0$ $aB = 0,0$ $a'B = 0,2708$ $aS = 0,7692$ $a's = 1,0$	$f=1,0$ $aB = 0,9745$ $a'B = 1,0$ $aS = 0,1695$ $a's = 0,3871$	$f=1,0$ $aB = 0,9669$ $a'B = 1,0$ $aS = 0,5$ $a's = 1,0$	$f=1,0$ $aB = 0,9703$ $a'B = 0,9512$ $aS = 0,5556$ $a's = 1,0$	$f=1,0$ $aB = 0,9706$ $a'B = 1,0$ $aS = 0,4348$ $a's = 0,7273$
L 2	$f=0,5397$ $aB = 0,8909$ $a'B = 0,75$ $aS = 0,5263$ $a's = 1,0$	$f=1,0$ $aB = 0,0$ $a'B = 0,4583$ $aS = 0,7692$ $a's = 1,0$	$f=0,4947$ $aB = 0,8798$ $a'B = 0,7434$ $aS = 0,1695$ $a's = 0,3871$	$f=0,6003$ $aB = 0,8904$ $a'B = 0,7434$ $aS = 0,5$ $a's = 1,0$	$f=0,5830$ $aB = 0,8894$ $a'B = 0,7071$ $aS = 0,5556$ $a's = 1,0$	$f=0,6119$ $aB = 0,9026$ $a'B = 0,7434$ $aS = 0,4348$ $a's = 0,7273$

Tabela 1 – Resumo de valores das métricas e conformidade

Fonte: Adaptado de Rosinat e Aalst (2006).

Se ponderarmos igualmente as três métricas f , $a'B$ e $a'S$, então o processo M1 é o que tem melhor conformidade para o *log* de evento L1, o modelo M4 é o que tem melhor desempenho para o *log* L2 e o modelo M2 com relação ao *log* de eventos L3. Nesta avaliação as métricas aB e aS não foram consideradas porque elas não são estáveis o suficiente (conforme demonstrado anteriormente) para comparar todos os modelos de processos entre si.

Recorda-se que a métrica aB é afetada pelas propriedades estruturais. Ainda assim, com algumas restrições, de forma comparativa, as métricas ab e a pode ser bem utilizadas. Por exemplo, a métrica ab determina que o modelo $M1$ é o que tem o comportamento mais adequado ao *log* de eventos $L1$.

Em condições ideais, um modelo de processo e o *log* devem ter ambos 100% de *Fitness*, *Appropriateness* comportamental e *Appropriateness* estrutural, porém, na prática, a dimensão *Fitness* é comumente mais dominante.

Assim, ao desenvolver análise de conformidade, é recomendável proceder em duas fases, onde primeiro analisa-se a dimensão *Fitness* e posteriormente avalia-se a *Appropriateness*.

2.8.7. Considerações finais

Para concluir, vale destacar o pensamento de Gunther e Aalst (2005), que ao analisarem a aplicação prática da técnica de mineração de processos, a caracterizam por três fases:

- a) Importação do *log* de eventos a partir do sistema transacional, que requer um mínimo de domínio e conhecimento de mineração de processos e a aplicação de ferramentas adequadas;
- b) Análise dos dados do *log* através de técnicas apropriadas de mineração de processos, que requer a participação de um especialista em mineração de processos de negócios;
- c) Interpretação e conclusão contextualizada dos resultados obtidos, com a participação de um especialista em mineração de processos e do usuário final ou dono do processo de negócio.

Neste capítulo de revisão teórica, foram apresentadas as definições e características de diversos autores relacionadas a processos, processos de negócios, *workflow*, *log* de eventos e mineração de processos, com sua abordagem formal representada pelas redes de Petri.

Com base nestes conceitos, foi mostrada em detalhes a técnica da conformidade utilizando a mineração de processos de negócios, suas características, métricas de análise e limitações.

Verificou-se que existem dois tipos de métricas para a análise de conformidade:

- a) *Fitness*: determina se os caminhos seguidos pelo log de eventos correspondem aos caminhos válidos no modelo do processo;
- b) *Appropriateness*: avalia o grau de precisão em que o modelo do processo descreve o comportamento observado, combinado com o grau de clareza em que ele é representado.

Constata-se ainda que a métrica dominante para a análise de conformidade é o *Fitness*. De maneira geral, um *log* de eventos está em conformidade com um modelo em redes de Petri, se tal rede pode gerar a seqüência de eventos no log (existe um caminho na rede para o log).

3. PROCEDIMENTOS METODOLÓGICOS

Este capítulo compreende a explicitação dos procedimentos metodológicos, ferramentas utilizadas e experimento realizados para atender os objetivos deste trabalho. Sua estrutura abrange os seguintes tópicos, como mostrado na figura 40.



Figura 40- Procedimentos metodológicos

3.1. Delimitação

Esta pesquisa foi realizada com o objetivo de desenvolver uma abordagem metodológica estruturada de análise de processos de negócios, fundamentadas na técnica de verificação de conformidade desenvolvida por Rozinat e Aalst (2006) e na técnica de geração de *logs* para teste de algoritmos de mineração apresentada por Medeiros e Gunther (2005).

O método utilizado consiste na aplicação da técnica da conformidade, com o uso do *framework* de mineração de processos ProM, que é um conjunto de ferramentas abrangente que suporta uma extensa gama de técnicas de mineração de processos na forma de plug-ins.

Um deles, o *Conformance Checker* executa um arquivo de *log* de eventos dentro de um modelo de rede Petri, enquanto captura informações de diagnóstico que podem ser acessadas posteriormente.

Este *plug-in* calcula as seguintes métricas:

- a) *Fitness*, baseada em fichas (obtendo o número de instâncias de processos para cada trilha de *log* analisado);
- b) *Appropriateness* comportamental simples (aB) e avançada (a'B);
- c) *Appropriateness* estrutural simples (aS) e avançado (a'S).

Durante a repetição do *log* o *plug-in* analisa as tarefas invisíveis que podem habilitar a próxima transição a ser repetida, e esta também apta a tratar as tarefas duplicadas.

Para aplicação da técnica de criação de *logs* de eventos sintéticos foi empregado o aplicativo CPN Tools, que permite a criação, edição, simulação e análise de redes de Petri coloridas.

A utilização desta ferramenta foi motivada pela possibilidade de construção do modelo de processo de negócio em análise, ou importá-lo da ferramenta ProM, num ambiente de simulação e com a aplicação da técnica citada, obter logs de eventos sintéticos gerados pelo modelo simulado, que serão posteriormente analisados pelos especialistas (do processo e em mineração de processos) e comparados no ProM.

Esta pesquisa relaciona os temas associados à modelagem de processos de negócios e a técnica de conformidade, buscando na fundamentação teórica de cada um, oportunidade de validar as idéias abordadas.

Em termos metodológicos, a delimitação consiste nas seguintes etapas:

- a) Efetuar pesquisa e classificação de referencial teórico apropriado.
- b) Identificar no referencial teórico os conceitos e métricas aplicados nas técnicas de conformidade.
- c) Propor após análise do referencial teórico, com relação à mineração de processos pela técnica da conformidade, e com relação à criação de *log* de eventos sintéticos, um roteiro metodológico estruturado para análise e melhoria de processos de negócios.
- d) Criar um processo de negócio para análise e aplicação da metodologia apresentada.

e) Analisar e propor modificações no modelo de processo e na estrutura do *log* de eventos, utilizando o ambiente de mineração de processos – *framework* ProM, e de simulação – *framework* CPN Tools.

f) Elaborar conclusões a respeito da aplicação realizada.

A figura 41 mostra num grau de abstração elevado, o ciclo de vida da abordagem metodológica apresentada nesta pesquisa, demonstra como um processo real de negócio, minerado a partir de seu log de eventos, pode ser conformado com o processo formal, resultando em análises que buscarão auxiliar na resposta a questão principal deste trabalho:

“É possível melhorar a conformidade entre os processos de negócios formalizados e os processos efetivamente realizados?”

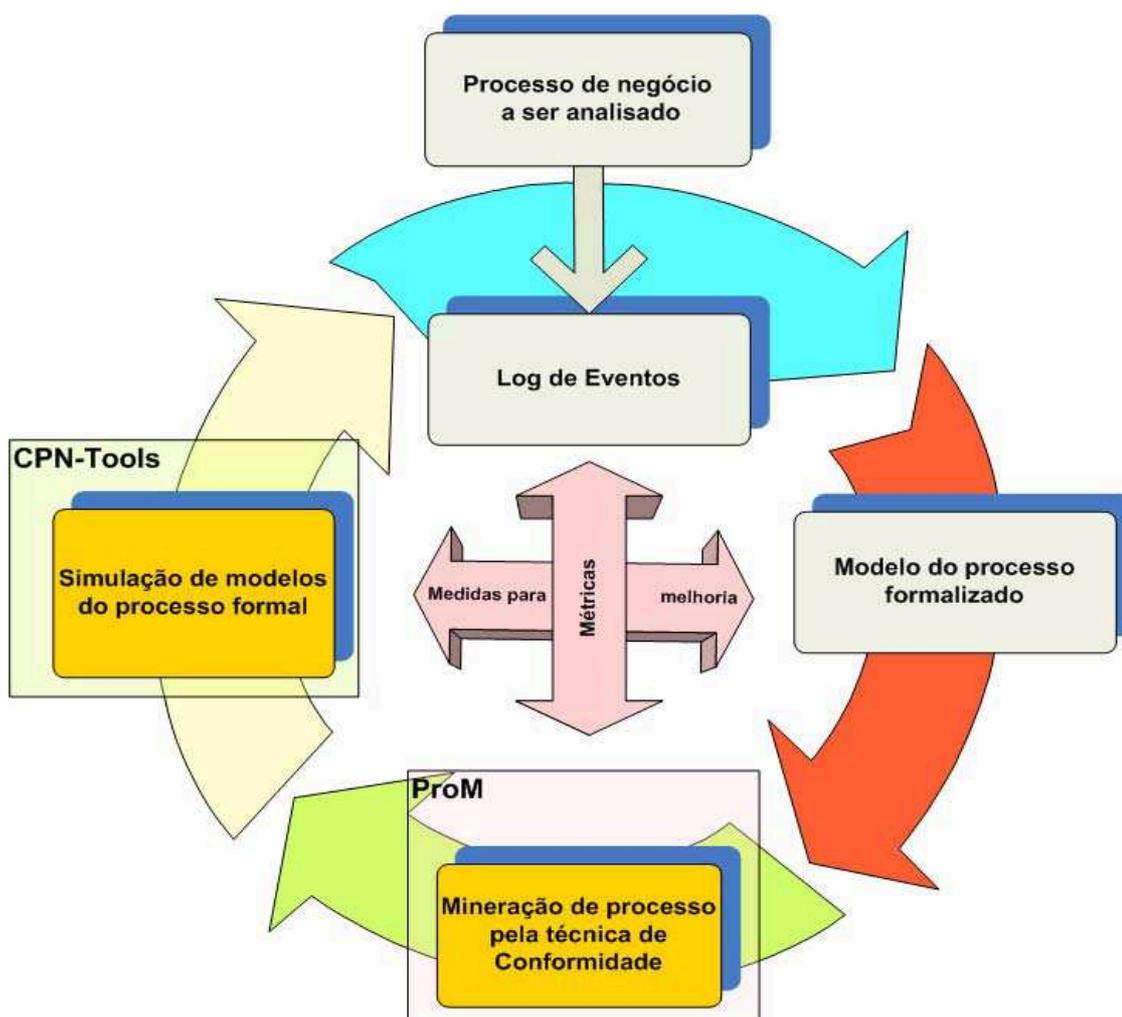


Figura 41 - Ciclo de vida da abordagem metodológica estruturada proposta

Fonte: Desenvolvido pelo autor.

Várias abordagens foram feitas ao longo deste trabalho, concentrando-se na revisão bibliográfica sobre a técnica da conformidade, uso das ferramentas ProM e CPN Tools, especificidades técnicas e descrição quanto operacionalização dos aplicativos e principalmente no descritivo do roteiro objeto da pesquisa.

Várias fontes foram consultadas para o levantamento de dados, em especial, as referências sugeridas numa das principais fontes de informação sobre mineração de processos, o site www.processmining.org.

A seguir serão apresentadas as ferramentas utilizadas, detalhando suas características, funcionalidades e forma de utilização no contexto desta pesquisa.

3.2. Ferramenta ProM

Segundo Medeiros e Weijters (2008), quando se trata de processos de negócios, as funcionalidades das ferramentas comerciais limitam-se em analisar e medir indicadores de desempenho tais como fluxo de tempo, taxa de falhas e frequências.

Visando suprir essas deficiências, para a aplicação das técnicas de mineração de processos, diversas ferramentas foram desenvolvidas na *Eindhoven University of Technology* durante os últimos cinco anos, tais como *Emit - Enhanced Mining Tool*, *Little Thumb*, *MisoN - Mining Social Network* e *MiMo – Mining Model*, cada uma delas se referindo a um determinada técnica e perspectiva de análise por mineração de processos.

Esses aplicativos e suas funcionalidades foram agrupados em uma única ferramenta denominada ProM, plataforma *open-source* desenvolvida em linguagem Java e conta com mais de 250 *plug-ins* que possibilitam a importação de diversas linguagem de modelagem de processos, tais como PNML, TPN, Aris, EPML e outras.

Para a mineração de processos existem *plug-ins* que suportam as técnicas de mineração de *control-flow* (com algoritmos *Alpha*, *Genéticos*, *Multi-phase*, etc); perspectiva organizacional (com algoritmos *Social Network*, *Staff Assignment*, etc);

perspectiva dos dados (com o algoritmo *Decision*) e para processos flexíveis e menos estruturados (com o algoritmo Fuzzy).

O *framework* ProM é uma ferramenta gratuita disponível em <<http://www.prom.sourceforge.net>> e para sua utilização dentro da técnica de mineração de processos por conformidade demonstrada, é necessário que se tenha o processo real modelado em formato de rede de Petri – PNML e um arquivo de *log* de eventos gerado pelo sistema transacional em formato XML.

A figura 42 mostra uma visão geral da ferramenta, e demonstra as relações entre o arquivo de *log*, os *plug-ins* e o usuário. O ProM possibilita a leitura dos arquivos de *log* através do componente Filtro de *log*.

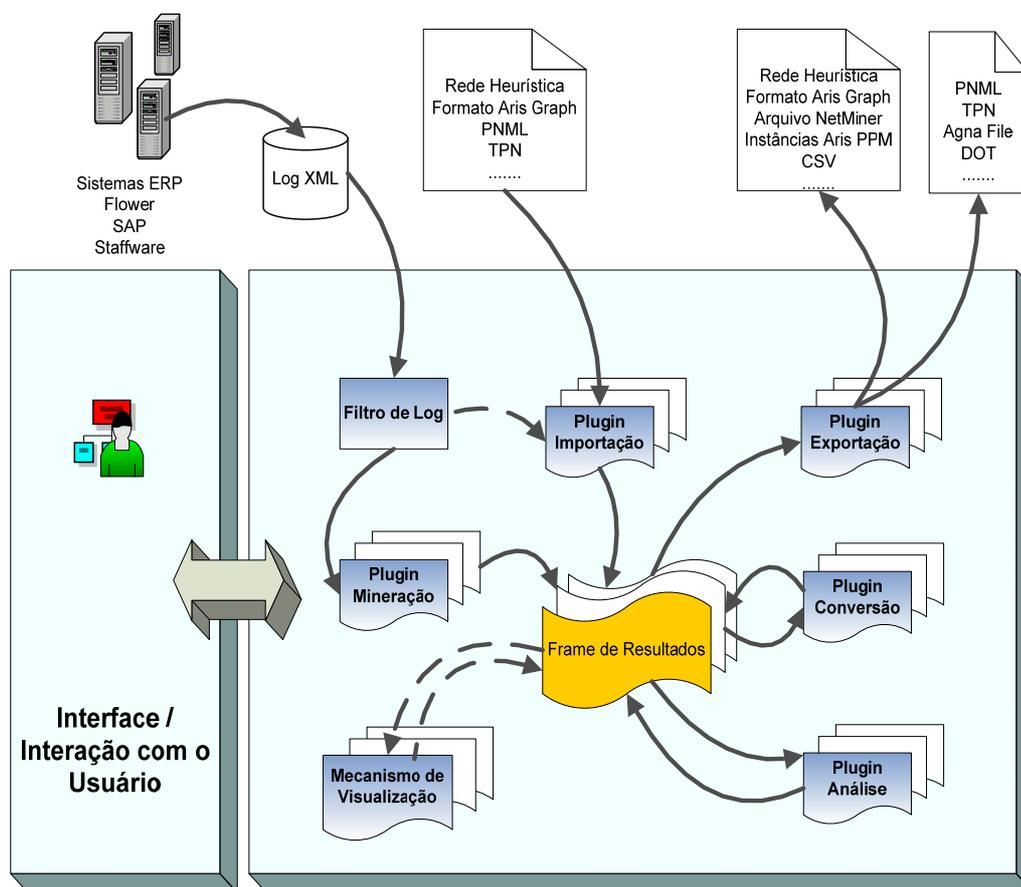


Figura 42 - Visão Geral da ferramenta ProM

Fonte: Adaptado de Dongen et al (2005b).

Através do componente de Importação e Exportação, é possível importar e exportar diversos formatos de modelos de processos. Um dos formatos que será utilizado no contexto desta pesquisa é o CPNTools 2.0 Export *plug-in*, que permite a

exportação de um modelo de processo em Petri para o *framework* CPN Tools, em procedimento já detalhado por Rozinat *et al* (2007).

O *plug-in* Mineração realiza a mineração de processos e armazena os resultados que são visualizados na tela na forma de rede de Petri, EPC ou rede Social. O *plug-in* Análise, a partir dos resultados da mineração, efetua análises relativas à rede de Petri. Por fim, o *plug-in* Conversão permite o intercâmbio dos resultados da mineração entre os diversos sistemas de modelagem de processos.

A etapa que precede o carregamento dos arquivos no ProM é a de inspeção e limpeza no arquivo de *log*. Este procedimento se justifica na medida em que certas questões só poderão ser respondidas se os dados estiverem no *log*. Por exemplo, não será possível determinar o tempo de atravessamento dos cases se o *log* não conter informações de tempo. Além disso, é recomendável remover toda e qualquer informação desnecessária do *log* antes de iniciar a mineração.

A conformidade pode ser verificada, conforme demonstrado por Rosinat e Aalst (2006b), através do *plug-in Conformance Checker* e assim responder a questão: Qual é a compatibilidade dos cases (instâncias de processos) com os modelos de processos?

A análise fornecida pelo *framework* ProM fornece informações detalhadas sobre os problemas encontrados durante a repetição do *log*. Na perspectiva do modelo, é possível saber o número de *tokens* perdidos, tarefas que não foram habilitadas, tarefas que ficaram habilitadas, quais e com que freqüência as tarefas e arcos foram utilizados. Na perspectiva do *log*, são indicados os pontos de comportamento de não conformidade para cada case do *log*.

3.3. Ferramenta CPN TOOLS

O CPN Tools é uma ferramenta de edição, simulação e análise de redes de Petri coloridas, desenvolvida como um projeto de pesquisa, o projeto CPN2000, na Universidade de Aarhus, Dinamarca. É um aplicativo gratuito, disponível através de licença em <http://wiki.daimi.au.dk/cpntools/_home.wiki>.

Segundo Ratzner *et al* (2003), o CPN Tools aproveita os avanços mais recentes obtidos no desenvolvimento da interação homem-máquina, para incorporar

funcionalidades poderosas e uma interface bastante flexível com o usuário, com diferentes tipos de *feedback* visuais que mantêm o usuário informado do estado, erros do sistema modelado, verifica a sintaxe de comandos e variáveis utilizadas, simulação, etc.

A figura 43 mostra uma visão geral da área de trabalho do CPN Tools.

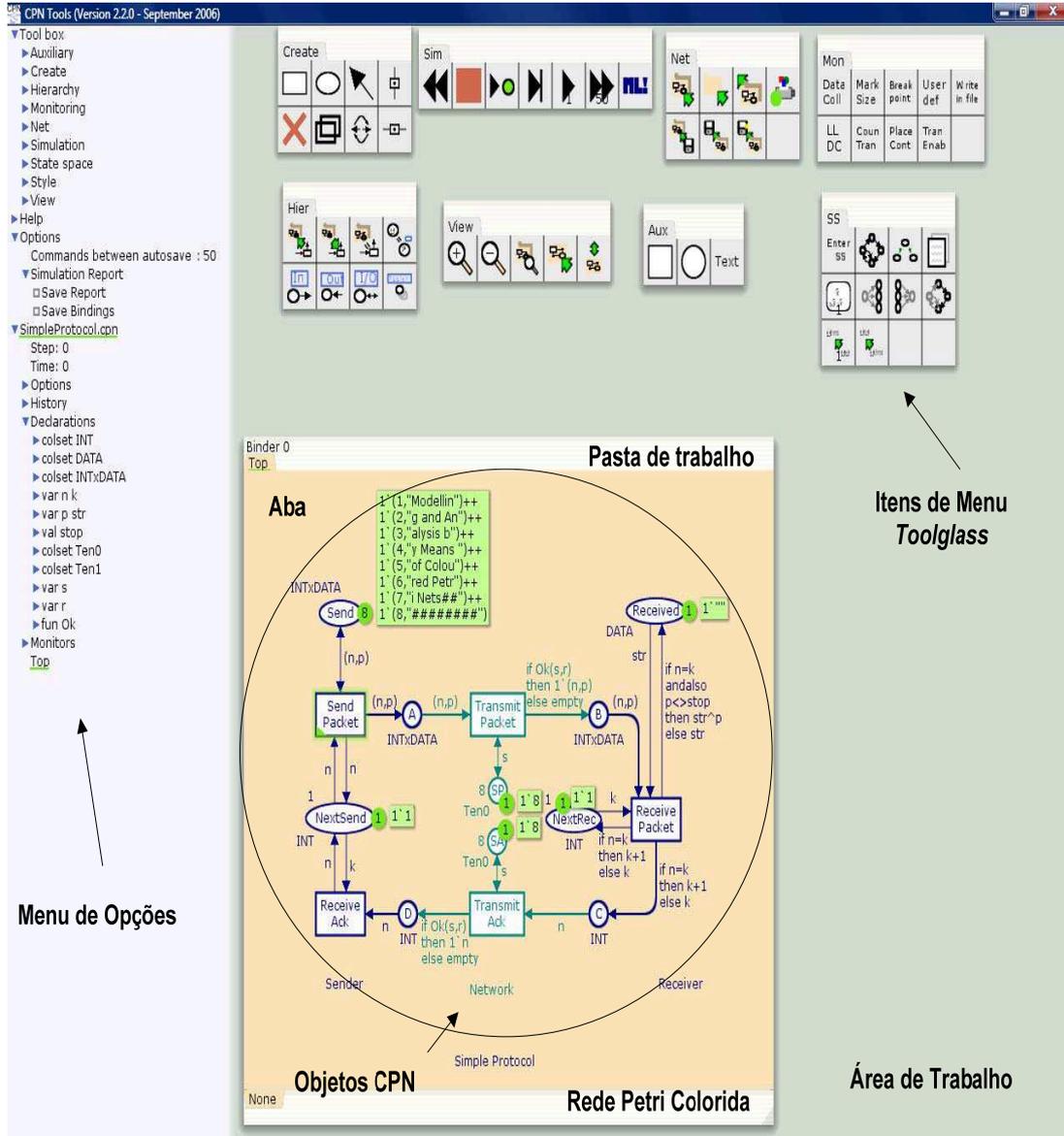


Figura 43 - Visão geral da área de trabalho

Fonte: Desenvolvido pelo autor.

A interface do CPN Tools possui uma coluna na esquerda chamada de índice ou menu de opções, similar a árvore de arquivos do Windows Explorer. As opções desse menu podem ser arrastadas para a área de trabalho e ficam flutuantes (*toolglass*), como indicado na figura 43. A interface contém ainda uma pasta de

trabalho chamada *Binder*, que pode agregar ainda diversas abas, cada uma representando uma rede de Petri colorida ou parte dela.

A criação de redes de Petri coloridas é fácil, rápida e bastante flexível, uma vez que oferece mais de uma maneira de se realizar uma tarefa, graças às funcionalidades da interface. Enquanto uma rede é criada, além de gerar o código de simulação, o CPN Tools auxilia o usuário de diferentes maneiras, fornecendo *feedback* visual a respeito da sintaxe da rede e do estado da ferramenta, ou alinhando automaticamente os objetos em algumas situações.

A inserção dessa ferramenta no contexto desta pesquisa é decorrente da capacidade de interface com o ProM e de suas potencialidades como aplicativo de simulação, permitindo a criação de um círculo virtuoso em termos de melhorias e aprimoramentos dos processos de negócio, como mostrado na figura 41.

Além disso, com o modelo de processo minerado inserido no ambiente de simulação, é possível realizar modificações tanto no processo de negócio como na estrutura do *log* de eventos, que pode ser novamente minerado a partir de *logs* gerados artificialmente no CPN Tools.

Essa sistemática quando adotada no âmbito das três perspectivas da mineração: controle de fluxo, organizacional e instâncias do processo; permite ao especialista detectar discrepâncias, encontrar possíveis gargalos, sugerir e testar suas modificações, sem quem sejam gastos recursos (humanos, financeiros e materiais) para promover alterações no processo real ou em alterações no sistema de gestão empresarial.

3.4. Geração de *logs* artificiais

A técnica de criação de *log* artificiais é obtida a partir de implementações de funções e declarações em ML (linguagem de modelagem do CPN Tools), na rede de Petri colorida, que representa o modelo a ser simulado. A idéia principal por trás desta técnica consiste em gerar *cases*, que irão circular na rede de acordo com as condições estabelecidas nos arcos que unem os lugares às transições.

A figura 44 ilustra o processo de criação de logs de eventos.

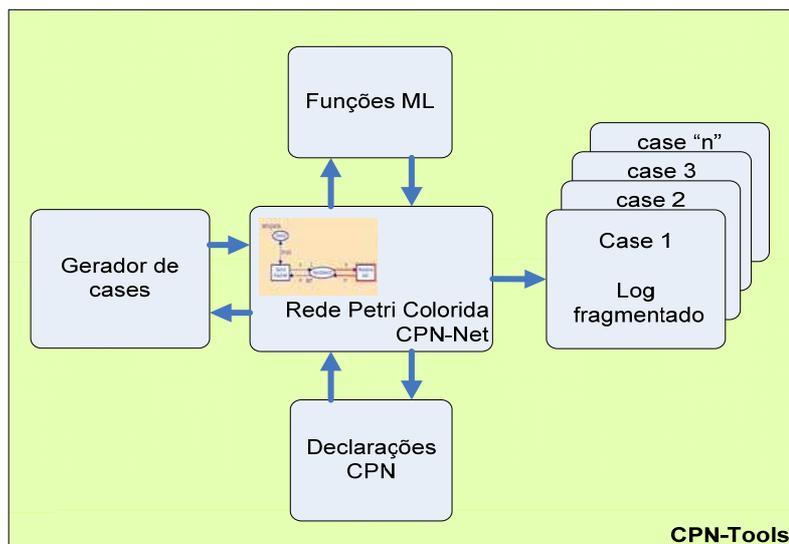


Figura 44 - Processo de criação de logs de eventos artificiais

Fonte: Desenvolvido pelo autor.

Para viabilizar a aplicação da técnica acima ilustrada, é necessário modificar o modelo de processo em rede de Petri colorida em análise, agregando os seguintes itens:

- a) Chamada para o arquivo *logging-FunctionsMultiplesFiles.sml*, com as funções ML para geração de *log*, desenvolvidas por Medeiros e Gunther (2005) e que pode ser obtido em www.processmining.org;
- b) Declarações que permitem declarar as cores das fichas ou *cases*, fazer inscrições, e importar as funções ML para o ambiente de simulação da rede de Petri;
- c) Modificar as inscrições de *input*, *output*, *action* das transições da rede, que irão chamar as funções de geração do *log*.

Para o propósito desta pesquisa, as declarações de uma rede de Petri colorida são divididas em:

- a) *Standard*: define uma ou mais cores, representadas por símbolos ou valores;
- b) *Net*: define os parâmetros que serão repassados para as funções de geração de *log*;
- c) *Log*: define as constantes que são utilizadas para informar o diretório onde o *log* será gravado, o nome do arquivo e sua extensão. Contém ainda a chamada para a função *logging-FunctionsMultiplesFiles.sml*.

Durante a execução da rede, os *cases* irão habilitar as transições, que contém instruções para registro em arquivos de *log*, de cada passagem do *case*. Esses arquivos precisam ser agregados no formato MXML (Mining XML) para que possam ser lidos pelas ferramentas de mineração de processos como o ProM.

Essa agregação é feita pelo *framework* ProMImport, que é um aplicativo *opensource* desenvolvido em Java, gratuito e disponível em <http://sourceforge.net/projects/promimport/> e que possibilita a conversão dos *log* de eventos, dos diferentes sistemas transacionais, para o padrão MXML, como mostrado na figura 45.

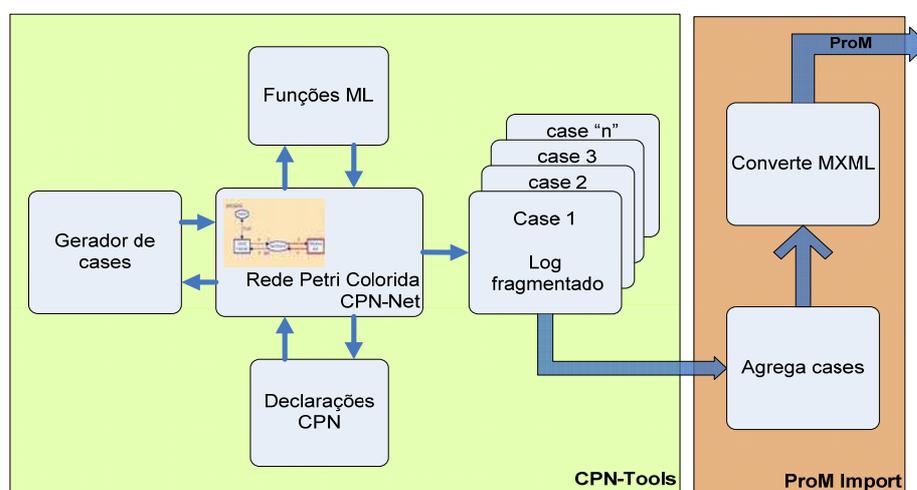


Figura 45 - Agregação de logs de eventos

Fonte: Desenvolvido pelo autor.

3.5. Detalhamento da abordagem metodológica estruturada

A figura 46 mostra em detalhes o esquema proposto na figura 41, onde a partir do modelo de processo formal e do log de eventos gerados pelo sistema de informação, são feitas análises usando a ferramenta de mineração de processos ProM, e propostas modificações no modelo de processo, utilizando a ferramenta de simulação CPN Tools. A estrutura apresentada visa atender o previsto no subcapítulo 1.3.1 – Objetivo geral:

“Esta dissertação tem como objetivo geral propor uma abordagem metodológica estruturada de análise de processos de negócios, através do emprego da técnica de conformidade e do ambiente de simulação de redes de Petri colorida.”

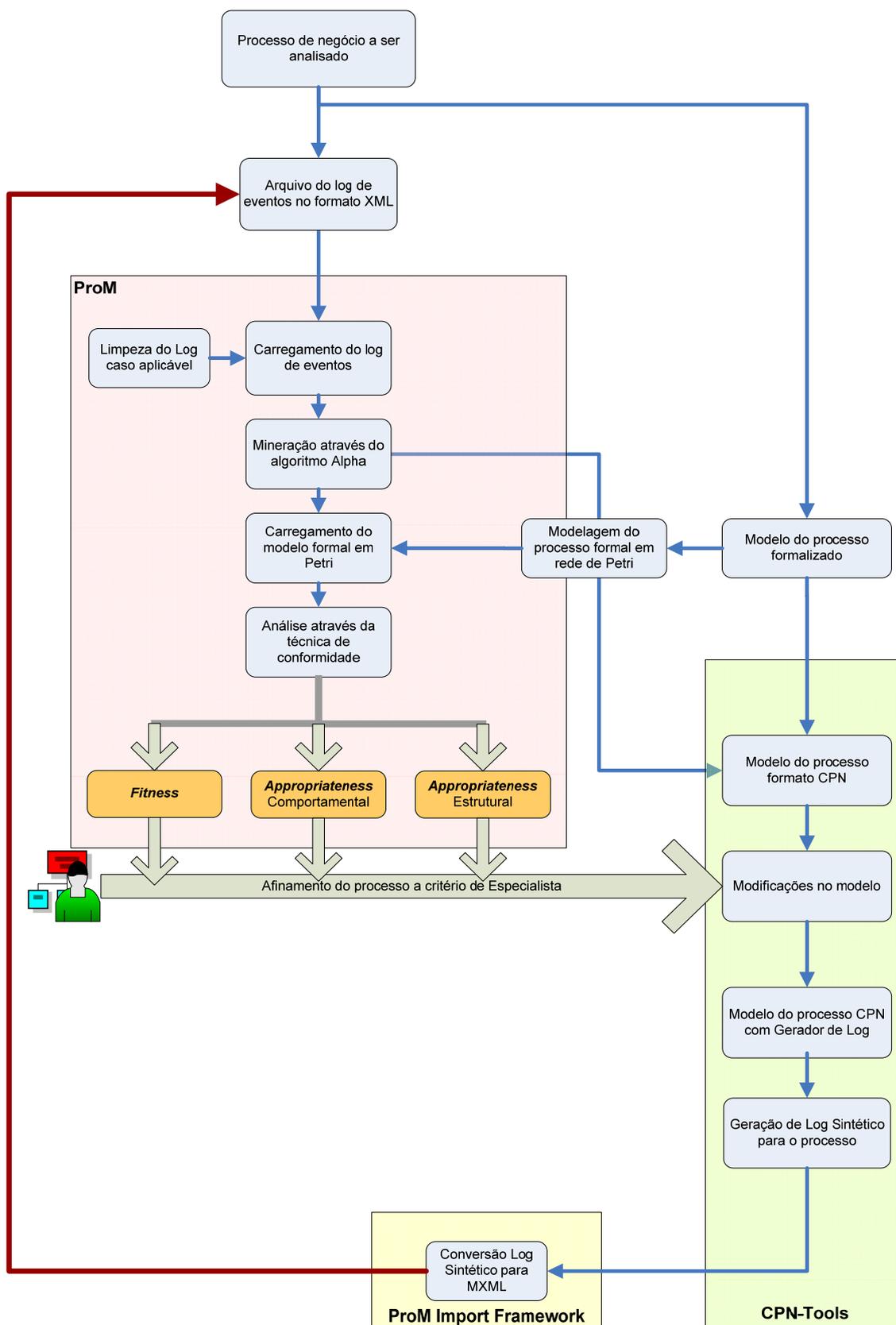


Figura 46 - Framework proposto
Fonte: Desenvolvido pelo autor.

A seguir carrega-se o modelo formal em PNML, sendo requerido pelo ProM que se faça uma correlação entre os eventos encontrados no *log* e no modelo formal, através de uma tela de mapeamento entre ambos. Estabelecida essa correlação, efetua-se a análise de conformidade, com a aplicação do algoritmo *Conformance Checker*. Cada dimensão de conformidade pode ser obtida individualmente ou em conjunto, de acordo com as opções selecionadas.

A técnica de Conformidade é empregada para determinar se o modelo de processo formal se assemelha ao processo que efetivamente está sendo executado pela organização. Essa avaliação deve ser feita por um especialista em processos de negócios e é obtida pelo emprego de três métricas de conformidade:

- a) *Fitness*: determina se os caminhos seguidos pelo log de eventos correspondem aos caminhos válidos no modelo do processo;
- b) *Appropriateness* Comportamental: avalia o grau de precisão em que o modelo do processo descreve o comportamento observado, combinado com o grau de clareza em que ele é representado;
- c) *Appropriateness* Estrutural: avalia o grau de clareza em que o modelo do processo é representado.

Dentro da técnica proposta de verificação entre o modelo de processo formal e real, em termos numéricos, cada um dos três indicadores pode assumir valores situados entre:

- a) 0.0 – não há conformidade observada;
- b) 1.0 – conformidade inteiramente observada.

Uma análise consistente considera as três métricas em conjunto, iniciando pela determinação do *Fitness*, depois pelo *Appropriateness* Comportamental, e por último, o *Appropriateness* Estrutural.

O modelo de processo real, encontrado a partir da mineração de processos com o algoritmo Alpha, é inserido na ferramenta de simulação CPN Tools, para que sejam feitas as modificações e adequações necessárias com o modelo formal. Essa inserção pode ser feita manualmente, desenhando a rede de Petri colorida no CPN Tools ou utilizando uma opção do *framework* ProM chamada Export CPN-Tools 2.0, parametrizável, que cria automaticamente uma rede de Petri colorida.

A forma de verificação se as modificações propostas são adequadas quanto a melhoria da conformidade é testá-las novamente no ProM, utilizando antes disso, as funções de geração de *logs* sintéticos.

3.6. O Experimento

Com a finalidade de avaliar a funcionalidade da abordagem proposta, e se ela atende os demais objetivos propostos no subcapítulo 1.3.2, notadamente o mencionado no item b:

“Propor modificações no processo de negócios e no seu log de eventos e avaliar através da abordagem metodológica estruturada se houve melhoria na conformidade entre o processo formal e o processo real”,

foi desenvolvido um modelo de processo de negócio experimental, que do ponto de vista organizacional, tem suas atividades executadas pelos recursos: Sistema, Luiz, Marta, José e Carla, distribuídos conforme mostra a figura 47.

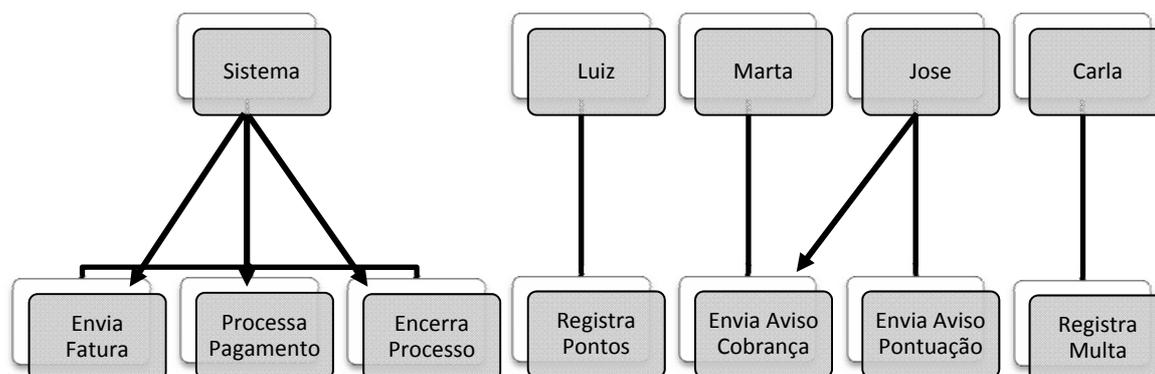


Figura 47 - Estrutura organizacional do processo formal

Fonte: Desenvolvido pelo autor.

O diagrama no formato PNML da figura 48, mostra do ponto de vista do controle de fluxo, o sequenciamento das atividades / tarefas executadas no modelo.

As transições representam as atividades, os lugares correspondem a uma condição atribuída antes ou após a execução das tarefas. Os arcos ligam os lugares

às transições e permitem que as ações sejam executadas dentro da rede de acordo com condições estabelecidas.

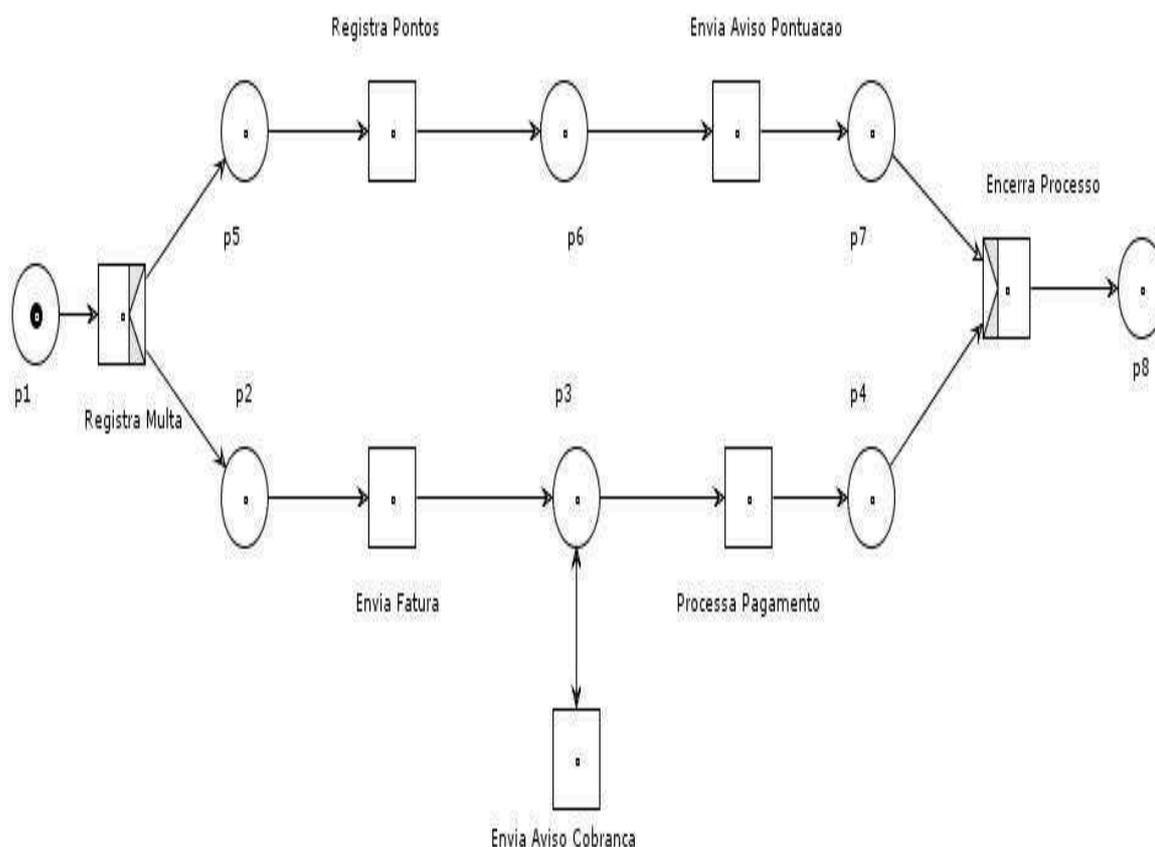


Figura 48 - Modelo de processo formal experimental

Fonte: Desenvolvido pelo autor.

Ainda na definição do experimento, no sentido de testar a conformidade dos modelos representando um modelo real, foi adotado como base um *log* de eventos denominado *FileFines.mxml*, disponível em www.processmining.org e frequentemente utilizado nos trabalhos e artigos científicos, como exemplo para diversas aplicações de mineração de processos.

Esse *log*, conforme ilustra a figura 49, foi adaptado para este trabalho, com tradução para o idioma português, dos tipos de eventos registrados e nome das tarefas executadas.

Com o modelo formal e o *log* de eventos representativo do modelo real, iniciou-se o teste da abordagem metodológica estruturada.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- edited with XMLSpy v2009 sp1 (http://www.altova.com) by Eliseu Ramos (Student) -->
3 <!-- MXML version 1.0 -->
4 <!-- This is a process enactment event log created to be analyzed by ProM. -->
5 <!-- ProM is the process mining framework. It can be freely obtained at http://www.processmining.org/ -->
6 <WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://is.tm.tue.nl/research/pro
7   <Source program="CPN Tools simulation"/>
8   <Process id="DEFAULT" description="Simulated process">
9     <ProcessInstance id="1" description="Simulated process instance">
10      <AuditTrailEntry>
11        <WorkflowModelElement>RegistraMulta</WorkflowModelElement>
12        <EventType>complete</EventType>
13        <Timestamp>1969-12-31T22:01:00.000+01:00</Timestamp>
14        <Originator>Carla</Originator>
15      </AuditTrailEntry>
16      <AuditTrailEntry>
17        <WorkflowModelElement>EnviaFatura</WorkflowModelElement>
18        <EventType>complete</EventType>
19        <Timestamp>1969-12-31T22:01:00.000+01:00</Timestamp>
20        <Originator>sisitema</Originator>
21      </AuditTrailEntry>
22      <AuditTrailEntry>
23        <WorkflowModelElement>EnviaAvisoCobranca</WorkflowModelElement>
24        <EventType>complete</EventType>
25        <Timestamp>1969-12-31T22:31:00.000+01:00</Timestamp>
26        <Originator>Marta</Originator>
27      </AuditTrailEntry>
28      <AuditTrailEntry>
29        <WorkflowModelElement>EnviaAvisoCobranca</WorkflowModelElement>
30        <EventType>complete</EventType>
31        <Timestamp>1969-12-31T23:01:00.000+01:00</Timestamp>
32        <Originator>Marta</Originator>
33      </AuditTrailEntry>
34      <AuditTrailEntry>
35        <WorkflowModelElement>EnviaAvisoCobranca</WorkflowModelElement>
36        <EventType>complete</EventType>
37        <Timestamp>1969-12-31T23:31:00.000+01:00</Timestamp>
38        <Originator>Marta</Originator>
39      </AuditTrailEntry>
40      <AuditTrailEntry>

```

Figura 49 - Recorte do log de eventos LogTrataMultasReal.mxml

Fonte: Adaptado de Medeiros (2006).

3.6.1. Carregamento do *log* e modelo no ProM

Primeiramente deve-se carregar no ProM, o *log* de eventos representativo do processo real, LogTrataMultasReal.mxml. Através do botão *start analyzing this log* e em seguida com a escolha do algoritmo Alpha, inicia-se a mineração do *log*.

Como resultado dessa mineração, é apresentado um fluxo do processo real, em notação Petri, conforme mostra a figura 50.

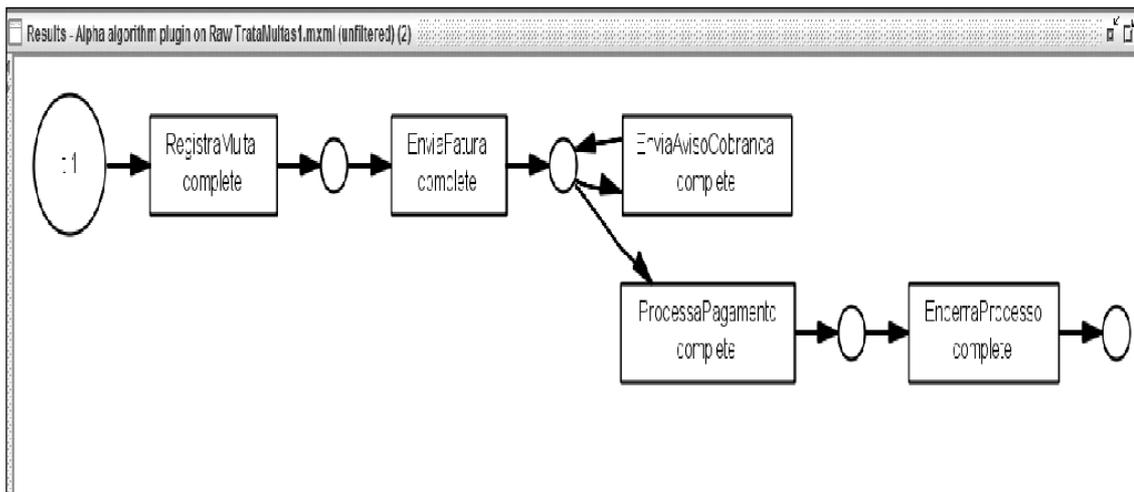


Figura 50 - Fluxo do processo real

Fonte: Desenvolvido pelo autor.

Em seguida é feito o carregamento do arquivo em formato pnml, que representa o modelo formal. Para o estabelecimento da conexão entre os modelos, é solicitado que seja feito um mapeamento entre os eventos encontrados no modelo e no *log*. Caso existam eventos no modelo que não ocorrem no *log*, pode-se escolher entre deixar esses eventos visíveis ou invisíveis.

3.6.2. Verificação da conformidade

Com ambos os modelos carregados, o ProM está pronto para a análise de conformidade, com a escolha do algoritmo *Conformance Checker* e do arquivo do modelo formal pnml. Antes de iniciar a análise, é possível definir quais métricas serão avaliadas.

A verificação de conformidade efetuada traz como resultado um *fitness* de 0.8579, conforme mostra a figura 51. Esse parâmetro e a observação visual no modelo conformado dão conta que existem duas atividades que não estão sendo realizadas no processo real (Registra Pontos e Envia Aviso de Pontuação).

Os dois círculos vermelhos representam lugares e o número no interior do mesmo, as fichas criadas artificialmente para reproduzir no modelo formal, o comportamento existente no *log*, ou seja, determinando se os caminhos seguidos pelo log de eventos correspondem a caminhos válidos no modelo do processo.

As transições mostradas na figura 51, com destaque em verde, indicam que essas atividades foram executadas normalmente no processo minerado.

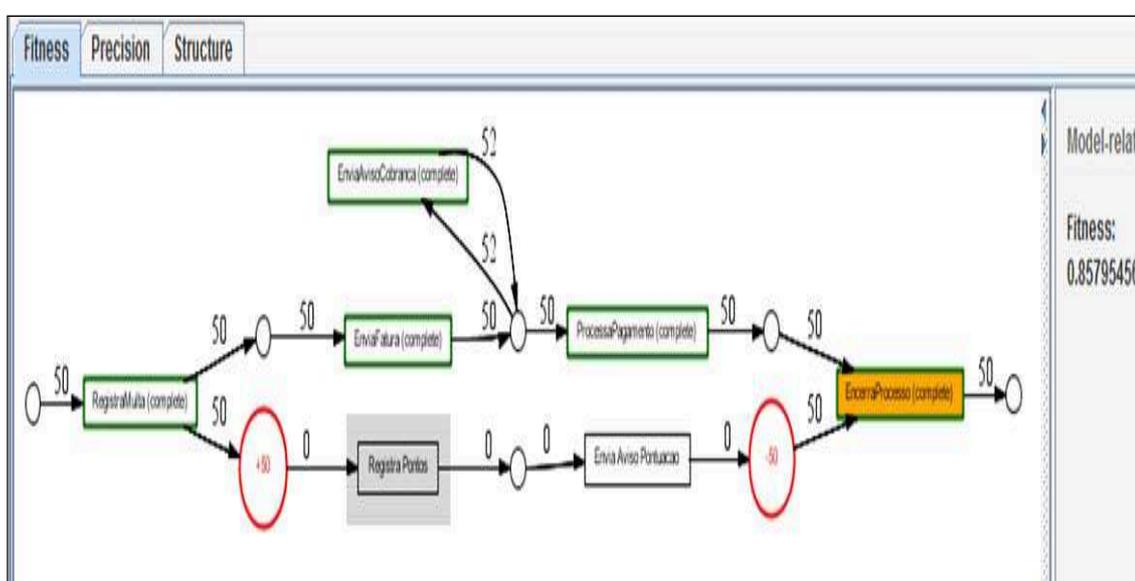


Figura 51 - *Fitness* entre modelo formal e real

Fonte: Desenvolvido pelo autor.

Em seguida foi verificada a métrica *Appropriateness* comportamental avançada, no sentido de avaliar o grau de precisão em que o modelo do processo descreve o comportamento observado, combinado com o grau de clareza em que ele é representado, conforme mostra a figura 52. O valor de *aB* encontrado foi de 0.25, reforçando o conceito da não conformidade entre ambos os modelos.

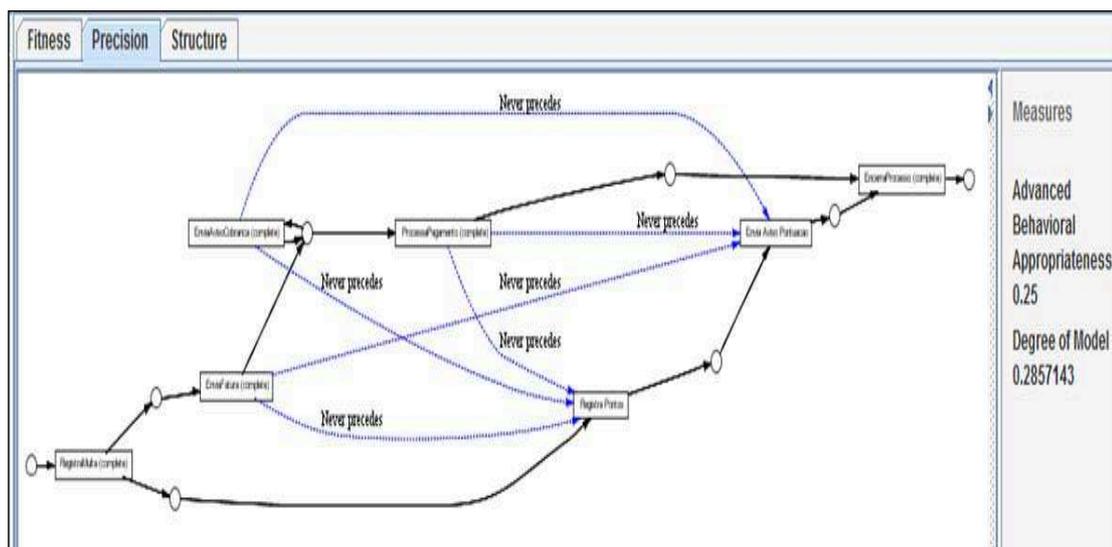


Figura 52 - *Appropriateness* comportamental avançado

Fonte: Desenvolvido pelo autor.

A métrica *Appropriateness* estrutural avançada, que indica o grau de clareza em que o modelo do processo é representado, teve um valor encontrado foi de 1.0, conforme indica a figura 53. Esse fato corrobora com a idéia de que as métricas de conformidade não podem ser analisadas isoladamente.

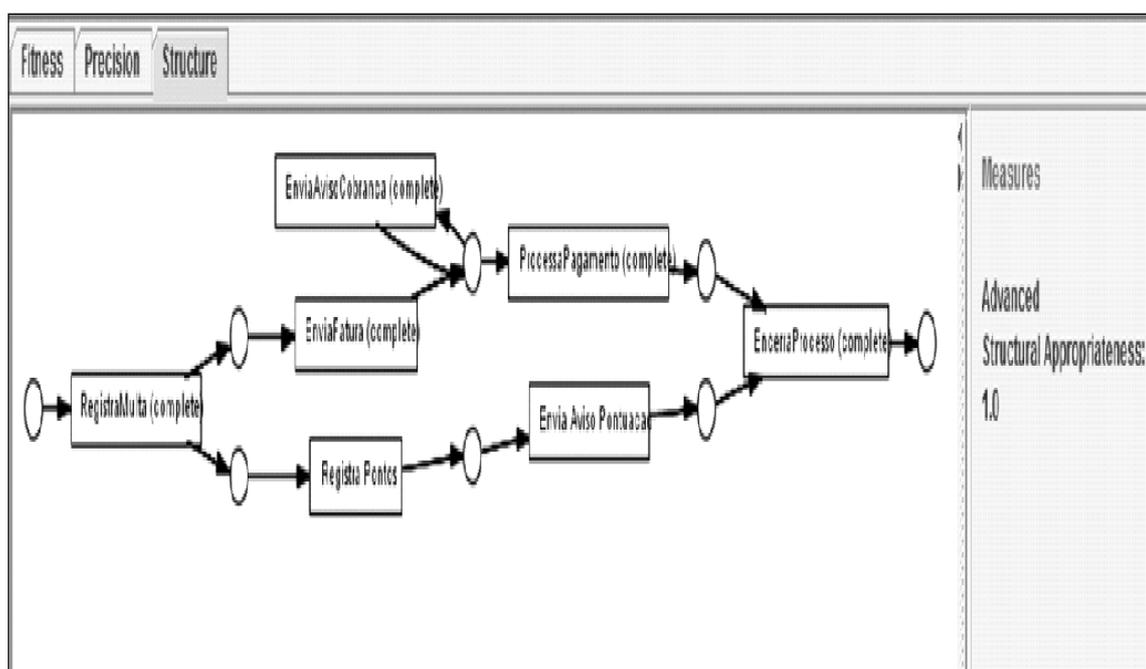


Figura 53- *Appropriateness* comportamental avançado

Fonte: Desenvolvido pelo autor.

3.6.3. Análise do processo encontrado

Uma análise realizada a partir dos dados encontrados denota que no processo de tratamento de multas, duas atividades não são executadas, e isso é visualizado facilmente na tela do *Fitness*.

O ProM oferece outra maneira de identificar as distorções entre os modelos, e saber por exemplo, qual recurso não está atuando de acordo com o esperado, através do algoritmo de mineração *Organization Miner*, que mostra na perspectiva organizacional, os recursos envolvidos nas atividades.

Considerando que o *log* experimentado contém informações relativas aos recursos, após a mineração com esse algoritmo, foi detectado que Luiz não realizou nenhuma atividade, além de confirmar que as atividades de Registra Pontos e Envia Aviso Pontuação não foram executadas.

Essa conclusão foi possível confrontando o diagrama gerado pelo ProM e apresentado na figura 54, com a estrutura organizacional mostrada na figura 47.

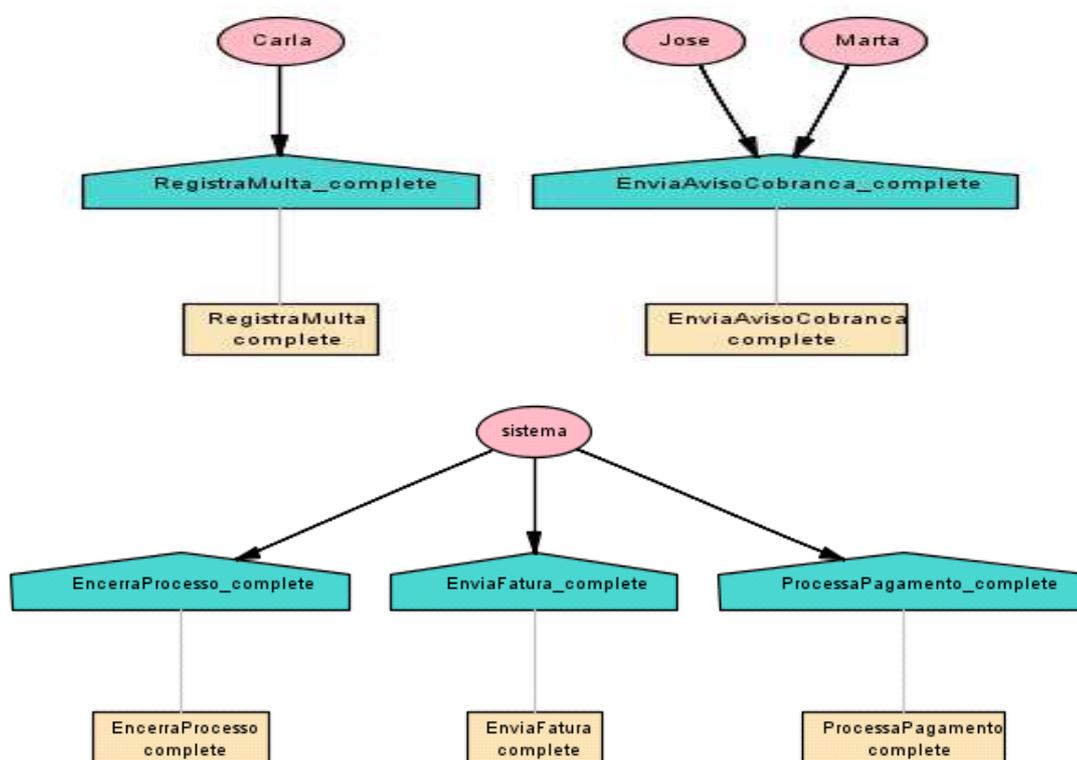


Figura 54 - Estrutura organizacional mostrada no log do processo real

Fonte: Desenvolvido pelo autor.

3.6.4. Modificações no processo

Com a constatação de que o processo formalizado não está sendo executado adequadamente, a abordagem metodológica estruturada sugere que se façam modificações no modelo de processo real ou até mesmo no modelo formal. Essas modificações serão feitas no ambiente de simulação CPN Tools.

O ponto de partida para a utilização da ferramenta de simulação é a criação de uma rede de Petri colorida que reproduza o modelo de processo real, através do seu desenho no ambiente CPN Tools.

Devem ser adicionadas as atividades que não estavam sendo executadas no modelo real: Registra Pontos e Envia Aviso Pontuação, bem como agregados o módulo gerador de *cases* e as inscrições nas transições para geração do *log* artificial.

A RdPC do processo real tem a estrutura como mostrada na figura 55.

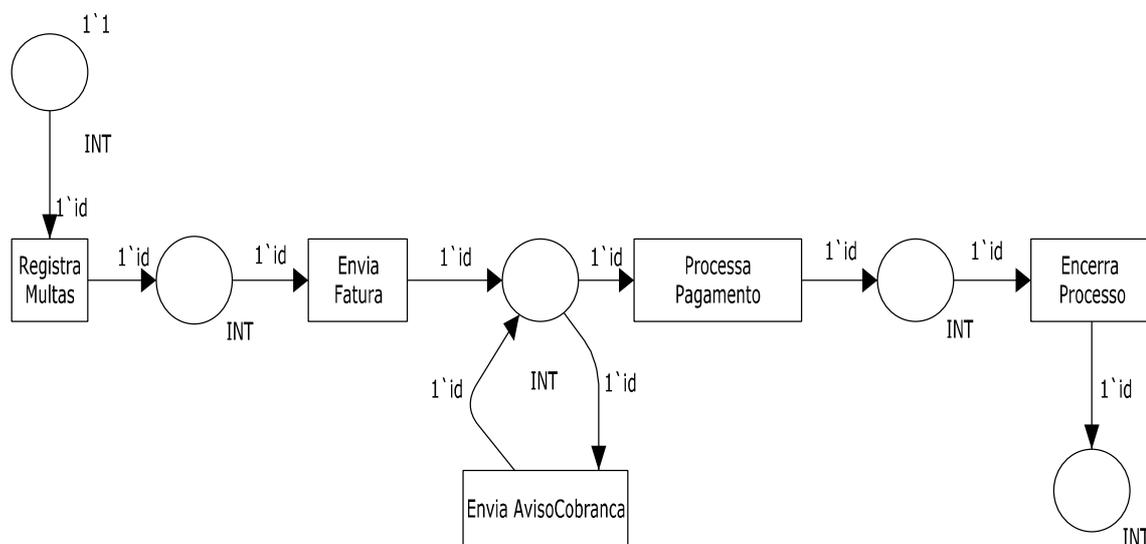


Figura 55 - Representação básica em RdPC do modelo real

Fonte: Desenvolvido pelo autor.

As declarações necessárias para o funcionamento do experimento são as descritas no Quadro 1, incluindo informações a respeito dos recursos utilizados e respectivas funções desempenhadas.

```

(* Standard declarations *)
colset E = with e;
colset INT = int;
colset BOOL = bool;
colset STRING = string;
(* Net declarations *)
colset TIMEDINT = int timed;
colset REGRA0 = subset STRING with ["Sistema"];
colset REGRA1 = subset STRING with ["Carla"];
colset REGRA2 = subset STRING with ["Marta", "Jose"];
colset REGRA3 = subset STRING with ["Jose"];
colset REGRA4 = subset STRING with ["Luiz"];
colset INT_REGRA2 = product INT * REGRA2 timed;
var id: INT;
var regra0: REGRA0;
var regra1: REGRA1;
var regra2: REGRA2;
var regra3:REGRA3;
var regra4:REGRA4;
fun OK(id) =
if id < 500 then true
else false;
(* Log declarations *)
val FILE = "c:/1/cpnlogs/"
val FILE_EXTENSION = ".cpnxml"
use "loggingFunctionsMultipleFiles.sml";

```

Quadro 1 - Declarações para a RdPc do modelo

Fonte: Adaptado de Medeiros e Gunther (2005).

Além das declarações mencionadas é necessário inserir as inscrições de *input/output/action* a cada transição do processo e a chamada para a função *addATE*, responsável pela gravação da trilha do *log*. Para o modelo do experimento, a inscrição genérica consiste em:

- input (id, regraⁿ);
- output();
- action
- (addATE(id,"nomedatransição",["tipodeevento"],calculateTimeStamp(),
regraⁿ, []));

As modificações propostas para o modelo de processo real são vistas na figura 56.

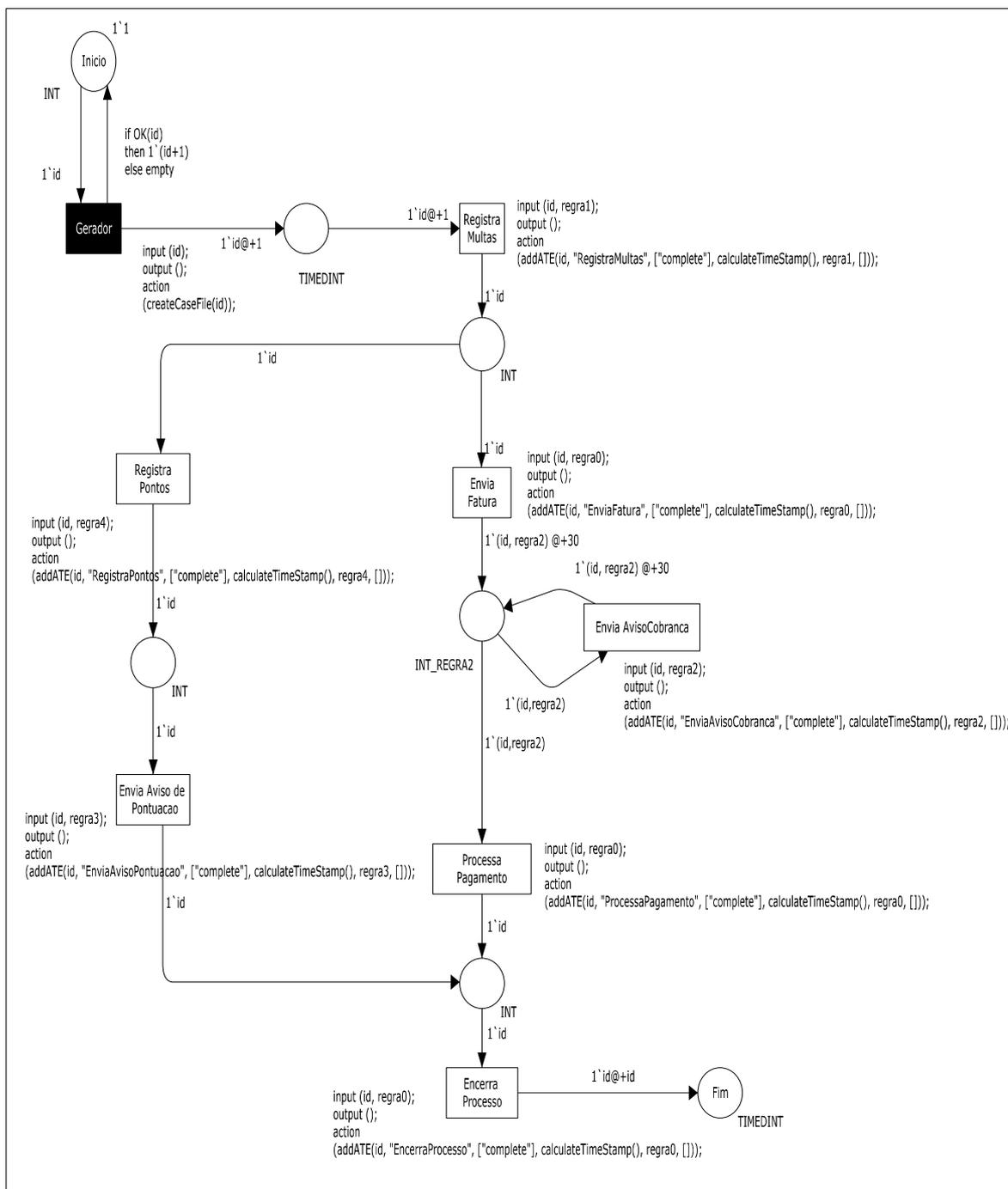


Figura 56- Modelo de processo real melhorado em RdPC
Fonte: Desenvolvido pelo autor.

Com o modelo em RdPC, pode-se iniciar a simulação da rede, atribuindo fichas a transição Gerador. Esse procedimento desencadeia o processo de circulação de cases na rede, gerando uma trilha de log a cada transição habilitada. A quantidade de cases gerada é determinada pela função OK, que pode ser alterada conforme a necessidade.

3.6.5. Geração e Agregação de Logs

No caso da rede utilizada no experimento, após a geração dos cases na quantidade desejada, a rede é finalizada. O próximo passo é agregar os arquivos gerados para formar um arquivo de *log* no padrão MXML. Para isso é preciso utilizar a ferramenta ProMImport, citada anteriormente e mostrado na figura 57.



Figura 57 - Ferramenta ProMImport

Fonte: Desenvolvido pelo autor.

Através da aba *Filter properties*, indica-se os diretórios origem, destino e extensão dos arquivos de processamento. O processo de agregação é disparado pelo botão *Start*. Com a conclusão da operação, um arquivo mxml é disponibilizado e pode ser aberto pela ferramenta ProM.

Para o experimento, foram gerados 500 arquivos, que formaram um de *log* de eventos, chamado *tratamultascpn.mxml*. Esse arquivo foi carregado na ferramenta de mineração de processos ProM e nova verificação de conformidade foi realizada.

3.6.6. Análise do processo modificado

Com a verificação da conformidade entre o modelo formal e o modelo modificado, foi obtido um modelo de processo modificado, mostrado na figura 58, a partir da mineração do *log* sintético TrataMultasCPN.mxml, gerado na rede acima.

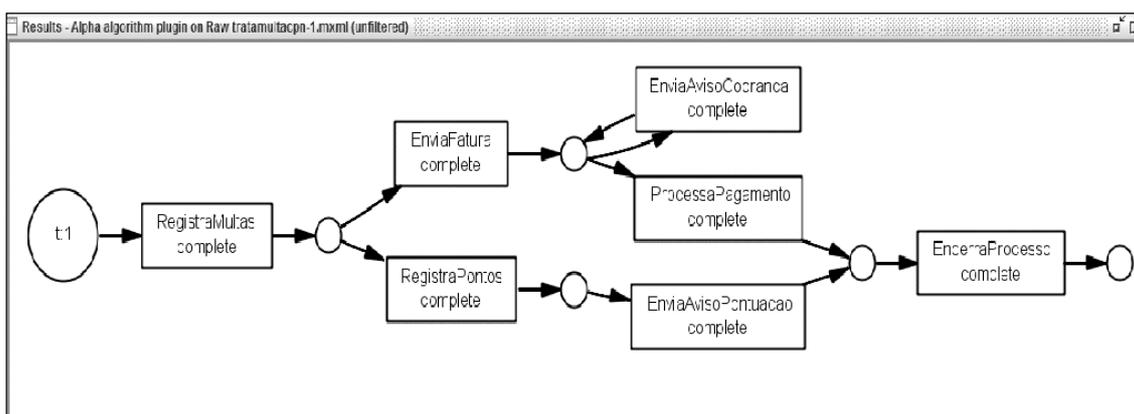


Figura 58 - Modelo de processo modificado obtido a partir da mineração do log
Fonte: Desenvolvido pelo autor.

A verificação de conformidade efetuada trouxe como resultado um *fitness* de 1.0, conforme mostra a figura 59, indicando que o modelo formal e o modelo modificado reproduzem o mesmo comportamento, quando submetido à execução do log. Com a adição das duas atividades (Registra Pontos e Envia Aviso de Pontuação) que não estavam sendo realizadas, ambos os modelos, formal e modificado, se tornaram idênticos e podem ainda ser o ponto de partida para outras análises.

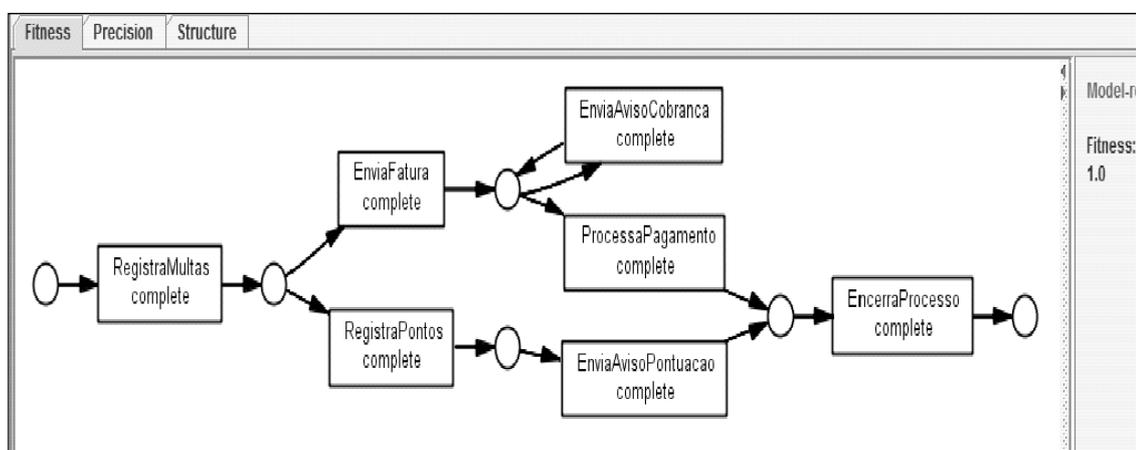


Figura 59 - *Fitness* entre modelo formal e modificado
Fonte: Desenvolvido pelo autor.

Quanto ao grau de precisão e clareza dos modelos analisados, dado a igualdade entre os mesmos, a *Appropriateness* comportamental avançada e a *Appropriateness* estrutural avançada encontrada foi de 1.0, conforme ilustrado na figura 60 e 61.

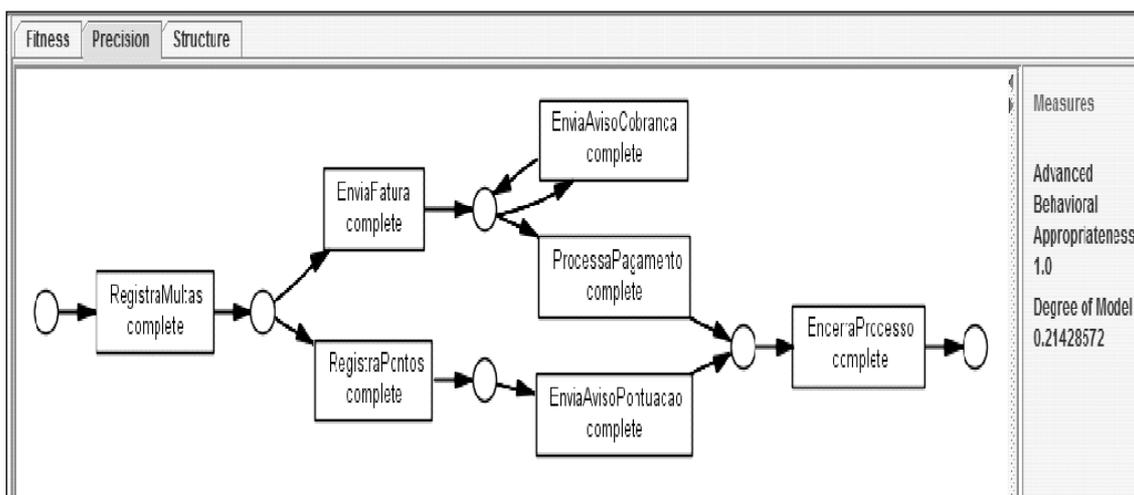


Figura 60 - *Appropriateness* comportamental avançado

Fonte: Desenvolvido pelo autor.

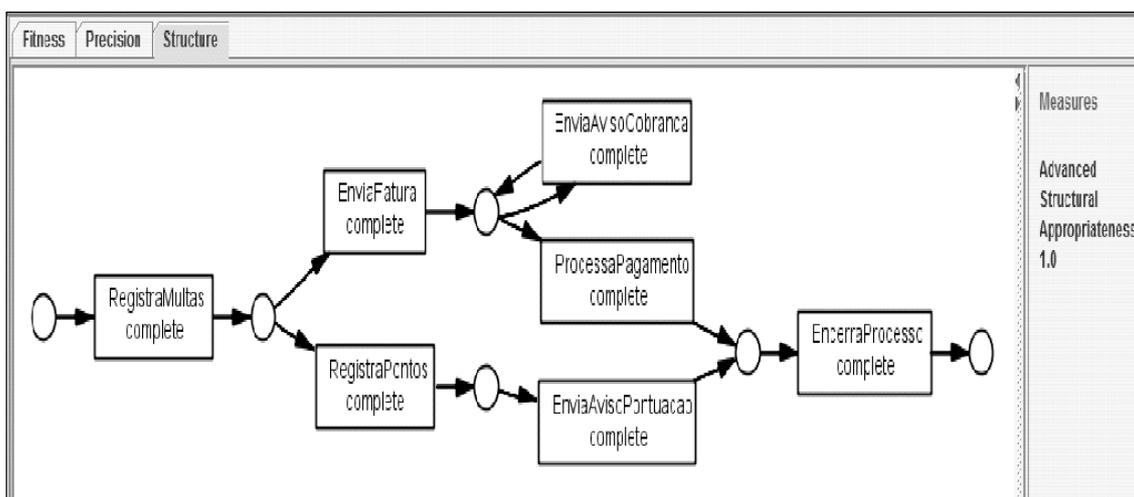


Figura 61 - *Appropriateness* estrutural avançada

Fonte: Desenvolvido pelo autor.

A figura 62, mostra uma nova configuração para a estrutura organizacional, resultado de uma nova mineração no log, demonstrando como as as duas atividades (Registra Pontos e Envia Aviso de Pontuação) e os recursos Luiz e José estão localizados no contexto do organograma da empresa.

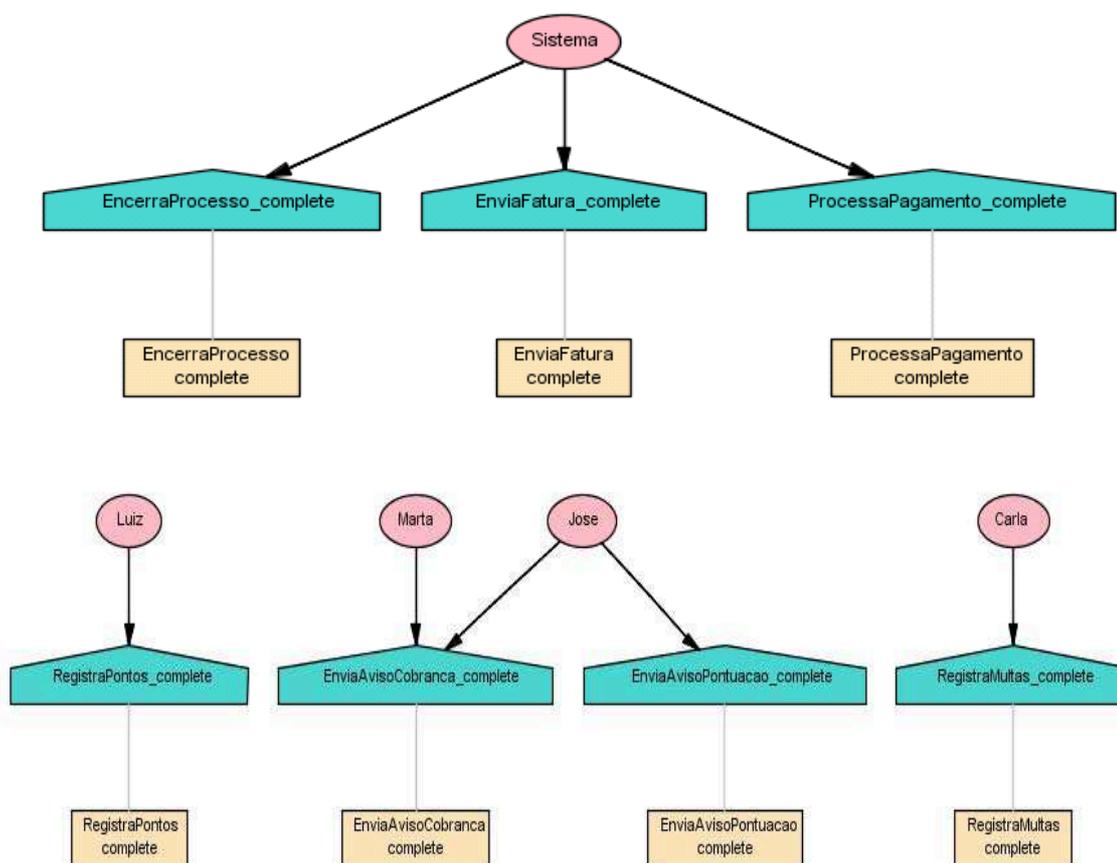


Figura 62 – Nova configuração da estrutura organizacional do processo modificado

Fonte: Desenvolvido pelo autor.

3.7. Considerações sobre o Experimento

Este capítulo buscou apresentar os procedimentos metodológicos adotados nesta pesquisa, detalhando os passos necessários para a aplicação da abordagem metodológica estruturada para a análise de processos de negócios, que de forma resumida, consistem em:

- a) Obter o modelo do processo em formato pnml;
- b) Obter o *log* de evento no formato XML / MXML;
- c) Carregar ambos na ferramenta ProM;
- d) Verificar a conformidade através do *Fitness* e *Appropriateness*;
- e) Confrontar o modelo formal e real e analisar junto com o especialista;

- f) Criar um modelo de simulação na ferramenta CPN Tools, a partir do modelo real, implementando modificações e melhorias necessárias;
- g) Adicionar ao modelo criado, o gerador de log;
- h) Avaliar necessidade de inserir perspectivas adicionais no *log*;
- i) Gerar log para o processo modificado;
- j) Analisar log criado com a ferramenta ProM;
- k) Avaliar a conformidade obtida, nas dimensões *Fitness* e *Appropriateness*, verificando se modificações efetuadas trouxe melhora nas métricas e se é necessário promover novas alterações;

Procurou-se detalhar também, o uso das ferramentas empregadas na realização do experimento. É importante ressaltar que apesar da eventual dificuldade de adaptação no ambiente de simulação CPN Tools e na modelagem em redes de Petri coloridas, a ferramenta ProM oferece recursos de integração com o CPN Tools, que praticamente eliminam a necessidade de projetar uma RdPC a partir do zero.

Esse aspecto é fundamental no caso de processos de negócios complexos, que naturalmente demandariam horas adicionais no desenho da RdPC equivalente.

4. CONCLUSÃO

As organizações são formadas por processos de negócios, e o entendimento a respeito de seu funcionamento, suas características e interações são fundamentais para o sucesso das empresas.

A identificação dos processos efetivamente realizados proporciona obter uma maior compreensão da organização, auxiliando a racionalizar o fluxo de informações existente na empresa.

Sistemas de gestão empresarial estão cada vez mais presentes nas empresas, para suportar seus processos de negócios. Ocorre que muitas vezes esses processos formalizados e implementados, podem não atender todos os cenários possíveis e se tornarem desatualizados.

É nesse contexto que a mineração de processos é aqui abordada. Com o uso de ferramentas e técnicas específicas, possibilita analisar e aperfeiçoar os processos modelados.

Esse trabalho se propôs a examinar a partir de uma extensa revisão teórica, as principais questões envolvidas na aplicação da técnica da conformidade, demonstrar suas características, dimensões e funcionalidades.

Foram analisados conceitos e definições sobre o formalismo das redes de Petri clássicas e coloridas, essencial para sustentar os conceitos relativos à mineração e simulação de processos.

A revisão bibliográfica foi concebida com o propósito de obter subsídios para a elaboração da abordagem metodológica estruturada, que posteriormente foi testada e avaliada através de um experimento.

4.1. Análise dos objetivos propostos

Com relação ao atendimento ao objetivo geral estabelecido para esta pesquisa, constata-se que com os resultados obtidos, a partir da aplicação da abordagem estruturada e das métricas verificadas, que a proposta constitui numa alternativa perfeitamente aplicável, para auxiliar na resposta a pergunta tema deste

trabalho: “é possível melhorar a conformidade entre os processos de negócios formalizados e os processos efetivamente realizados?”.

Além de contribuir na melhora da conformidade entre os processos, a técnica e as ferramentas de mineração e simulação de processos, auxiliam o especialista a localizar a origem das distorções entre o modelo formal e real.

Para viabilizar o objetivo principal desta dissertação, foram estabelecidos três objetivos específicos, e aqui é descrito o que foi atendido para cada um deles.

O primeiro objetivo específico era relacionado à identificação no referencial teórico, de definições e conceitos sobre gerenciamento de processos de negócios, *log* de eventos, mineração de processos, métricas de conformidade em processos de negócios e redes de Petri clássica / colorida. Todo esse aporte bibliográfico constante do capítulo 2.1 contribuiu significativamente para o desenvolvimento e aplicação da abordagem metodológica estruturada.

O segundo objetivo específico propunha introduzir modificações no processo de negócio e no seu *log* de eventos, de forma a avaliar através da abordagem metodológica estruturada, se houve melhoria na conformidade entre o processo formal e o processo real. Os resultados obtidos com o experimento realizado no capítulo 2.2, apontam que a conformidade pode ser melhorada com a aplicação da metodologia proposta.

Por fim, o terceiro objetivo consistiu de analisar os resultados obtidos e elaborar conclusões sobre o experimento realizado no segundo objetivo. Essa análise e considerações finais estão relatadas no decorrer da realização dos testes, no capítulo de Procedimentos Metodológicos.

4.2. Limitações da pesquisa e sugestão de trabalhos futuros

Um trabalho de pesquisa não pretende exaurir um tema e ser conclusivo, ao contrário, espera-se que possa ser estendido a partir de novas idéias, conforme as potencialidades de cada pesquisador.

Este estudo propôs a partir de concepções estabelecidas para a técnica de conformidade e para a geração de *logs* sintéticos, uma abordagem metodologia para a análise de processos de negócios. Entretanto a aplicação desta metodologia

abrangendo um número maior de experimentos pode servir de ponto de partida para outras pesquisas no assunto, com um aprofundamento na perspectiva organizacional, e a utilização dos *plug-ins* Genetic e Organizational Miner de mineração de processos.

Outro vetor de pesquisa que pode ser explorado se refere à utilização das técnicas apresentadas, para desenvolver e testar outros formatos de log, em particular o proposto pela Interface 5 da WfMC.

4.3. Considerações finais

Observou-se com este trabalho, que a abordagem metodológica estruturada proposta pode ser utilizada para analisar outras perspectivas dentro da mineração de processo.

A otimização de processo em ambiente de simulação e com a abordagem proposta, pode trazer vantagens no que diz respeito a redução de tempo, custos de desenvolvimento, alteração de sistemas e recursos envolvidos.

O estudo confirmou as ferramentas de mineração de processos ProM e de simulação de redes de Petri coloridas CPN Tools como adequadas a aplicação da metodologia proposta, cujo foco foi à verificação de conformidade e a análise das três métricas: *Fitness*, *Appropriateness* comportamental e *Appropriateness* estrutural. Entretanto, outras análises são perfeitamente aplicáveis, mesmo que o *log* do processo real não contemple todas as informações necessárias.

Com a geração de log sintético, é possível incrementar a análise do processo quanto ao aspecto organizacional, no que se refere à utilização dos recursos; de desempenho, no que se refere aos tempos de atravessamento e espera; auditoria, no que se refere às regras e acessos estabelecidos para cada usuário.

Quanto ao experimento realizado, verificou-se que a abordagem metodológica estruturada, baseada na verificação da conformidade entre modelos de processos formal e real, e a modelagem de processos em um ambiente de simulação, pode ser utilizada para auxiliar o especialista na melhoria e racionalização dos processos de negócio.

REFERÊNCIAS

AALST, W. M. P. V. D.; HEE, K. M. V.; **Workflow Management: Models, Methods and Systems**. MIT Press. Cambridge, Massachusetts, London, England. 2002.

AALST, W.M.P. van der et al. **Workflow mining: a Survey of issues and approaches**. Data and Knowledge Engineering, 47(2):237-267, 2003.

AALST, W. M. P. V. D.; HEE, K. M. V.; HOUBEN. G. J.; **Modelling and analysing workflow using a Petri-net based approach**. Proceedings of Second Workshop on Computer Supported Cooperative Work, Petri nets related formalisms. p. 31-50, 1994.

AALST, W.M.P. van der; WEIJTERS, A.J.M.M.; MARUSTER, I. **Workflow mining: discovering process models from event logs**. IEEE Transactions on Knowledge and Data Engineering, 16(9):1128–1142, 2004.

AALST, W.M.P. van der et al. **Business process mining: an industrial application**. In Department of Mathematics and Computer Science Technische Universiteit Eindhoven. 2006.

AALST, W. V. D. **Making Work Flow: On the Application of Petri nets to Business Process Management**. Springer-Verlag, Berlin, volume 2360, pages 1-22: 2002;

AALST, W.M.P. van der; ROSINAT, A. **Decision mining in ProM**. In Department of Mathematics and Computer Science Technische Universiteit Eindhoven. 2006.

AALST, W.M.P. van der; ROSINAT, A. **Decision mining in business processes**. In Department of Mathematics and Computer Science Technische Universiteit Eindhoven. 2006

AALST, W.M.P. van der; WEIJTERS, A.J.M.M. **Process mining**. In Department of Mathematics and Computer Science Technische Universiteit Eindhoven. 2006.

AALST, W.M.P. van der; GUNTHER, C.W. **Finding structure in unstructured processes: the case for process mining**. 2007

ABREU, Maurício. **Administração e Sistemas de Informação: soluções na era do conhecimento**. Rio de Janeiro. Apresentação. 72 páginas. 1999.

ANDER-EGG, Ezequiel. **Introducción a las técnicas de investigación social: para trabajadores sociales**. 7 ed. Buenos Aires: Humanitas, 1978. Parte II, capítulo 6.

AGRAWAL, R. GUNOPULOS, D. LEYMANN, F. **Mining process models from workflow logs**. In Sixth International Conference on Extending Database Technology, páginas 469–483, 1998.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 9001**: Sistema de Gestão da Qualidade: requisitos. Rio de Janeiro, 2000.

BPMN. **Business process modeling notation**. Business Process Management Initiative, 2006. 308 p. Disponível em: <http://www.bpmi.org>. Acesso em: 12 jul. 2009

DAVENPORT, T. H. **Process innovation - reengineering work through information technology**. Cambridge, Mass: Harvard Business School Press.1993.

DONGEN, B.F. van; ALST, W.M.P. van der. **A Meta model for process mining data**. In J. Casto and E. Teniente, editors, Proceedings of the CAiSE'05 Workshops (EMOI-INTEROP Workshop), volume 2, pages 309–320. FEUP, Porto, Portugal. 2005a.

DONGEN, B.F. van et al. **The ProM framework: a new era in process mining tool support**. In accepted tool presentation at ATPN 2005, 2005b.

ESTORILIO, Carla C. A. **O trabalho dos engenheiros em situações de projeto de produto: uma análise de processo baseada na ergonomia**. São Paulo, 2003. 317f. Tese (Doutorado em Engenharia de Produção). Universidade de São Paulo, USP, 2003.

EVANS, Bill. **Event log management: a guide to a Stress-free audit**. White Paper. <http://whitepaper.informationweek.com>. Acesso em: 30 jul. 2009.

GEORGAKOPOULOS, Dimitrios and HORNICK, Mark. **An overview of workflow management: from process modeling to workflow automation infrastructure**. Distributed and Parallel Databases, 3, 119-153. 1995.

GIL, A.C. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 1991.

GOLANI, M.; PINTER, Sholomit S. **Generating a process model from a process audit log: business process management**. Springer-Verlag Berlin Heidelberg. 2003.

GÜNTHER, C.W.; AALST, W.M.P. van der. **A generic import framework for process event logs**. In: Eder, J., Dustdar, S. (Eds.), Business Process Management Workshops Workshop on Business Process Intelligence (BPI 2006), Lecture Notes in Computer Science, vol. 4103. Springer, Berlin. pp. 81-92.

HAMMER, M.; CHAMPY, J. **Reengineering the corporation: a Manifesto for Business Revolution**. Harper Business, New York, 1993.

HORNIX, Peter. **Performance analysis of business processes through process mining**. Master's Thesis. Department of Mathematics and Computer Science Technische Universiteit Eindhoven. 2007.

JENSEN, K. **Coloured Petri Nets : Basic Concepts, Analysis Methods and Practical Use**. 1st EARCS Monographs on Theoretical Computer Science, Springer. 1992.

JENSEN, K. **A brief introduction to coloured Petri nets**. In: E. Brinksma (ed.): Tools and Algorithms for the Construction and Analysis of Systems. Proceeding of the TACAS'97 Workshop, Enschede, The Netherlands 1997, Lecture Notes in Computer Science Vol. 1217, Springer-Verlag 1997, 203-208.

JESTON, John; NELIS, Johan. **Business process management: practical guidelines to successful implementations**. Butterworth-Heinemann. 2006.

JOCHEM, R.; MERTINS, K. **Architectures, methods and tools for enterprise engineering**. International Journal of Production Economics. p.179-188. Elsevier. 2005.

LAWRANCE, P. **Workflow Handbook 1997, Workflow Management Coalition**. John Wiley and Sons, New York, ed. 1997.

LEYMANN, Frank; ROLLER, Dieter. **Production workflow: concepts and techniques**. New Jersey : Prentice Hall, 2000.

MARCONI, M.A.; LAKATOS, E. M. **Metodologia do trabalho científico**. 6^a ed. – 5^a reimpressão. São Paulo: Atlas, 2007.

MEDEIROS, A.K. **Genetic process mining**. PhD thesis, Department of Technology Management, Technical University Eindhoven, 2006.

MEDEIROS, A.K. Alves de; GUNTHER, C.W. **Process Mining: using CPN Tools to create test logs for mining algorithms**. Proceedings of the Sixth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, October 2005, Department of Computer Science, University of Aarhus, 177-190.

MEDEIROS, A.K.A. de; WEIJTERS, A.J.M.M. **ProM framework tutorial**. In Department of Mathematics and Computer Science Technische Universiteit Eindhoven. 2008.

MEDINA-MORA, R. et al. **The action workflow approach to workflow management technology**. Proceedings of the CSCW'92, ACM, Toronto, pp.281-8. 1992.

MUEHLEN, zur M. **Workow-based process controlling: Foundation, Design and Application of workow-driven Process Information Systems**. Logos, Berlin, 2004.

MUEHLEN, zur M.; Rosemann, M. **Evaluation of workow management systems - a meta model approach**. Australian Journal of Information Systems, 6(1):103-116, 1998.

MUEHLEN, zur M.; TING-Yi HO, Danny. **Risk management in the BMP lifecycle**. In Stevens Institute of Technology. Howe School of Tecnology Management. 2004.

MURATA, T. **Petri net: properties, analysis and applications**. Proceedings of the IEEE, v. 77, n. 4, p. 541-579, 1989.

OLIVEIRA, D.P.R. de. **Sistemas, organização & métodos: uma abordagem gerencial**. 13º ed. São Paulo: Atlas, 2002.

PROJECT MANAGEMENT INSTITUTE. **Um guia do conjunto de conhecimento em gerenciamento de projetos**. PMBOK, 2004 3ªedição. Newtown Square: PMI, 2005.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ. Sistema Integrado de Bibliotecas. **Normalização de trabalhos técnico-científicos**. 2008. Disponível em: http://www.pucpr.br/biblioteca/sibi/manual_normas2009.pdf. Acesso em: 16 julho 2009.

POPPER, K.S. **A lógica da pesquisa científica**. 2º ed. São Paulo: Cultrix, 1975.

RATZER, A.V. et al. **CPN Tools for editing, simulating and analysing coloured Petri nets**. In van der Aalst,W., Best, E., eds.: Applications and Theory of Petri Nets 2003: 24th International Conference, ICATPN 2003, Eindhoven, The Netherlands, June 23-27, 2003. Proceedings. Volume 2679 of Lecture Notes in Computer Science., Springer-Verlag (2003) 450–462

ROZENFELD, H. **Reflexões sobre a manufatura integrada por computador (CIM)**. In: Manufatura Classe Mundial: Mitos e Realidade. (1996: São Paulo). Anais. São Paulo, 1996.

ROSINAT, A.; AALST, W.M.P. van der; **Conformance Testing: measuring the alignment between event logs and process models**. In Department of Mathematics and Computer Science Technische Universiteit Eindhoven. 2005.

ROSINAT, A.; AALST, W.M.P. van der; **Conformance Checking of Processes Based on Monitoring Real Behavior**. In Department of Mathematics and Computer Science Technische Universiteit Eindhoven. 2006a.

ROSINAT, A.; AALST, W.M.P. van der. **ConformanceTesting: measuring the fit and appropriateness of event logs and process models**. Business Process Management. 2005. Workshops, volume 3812 of Lecture Note. Computer Science, pages163–176.Springer-Verlag, Berlin, 2006b.

ROSINAT, A.; AALST, W.M.P. van der; MANS, R.S.;SONG, M.; **Discovering colored Petri nets from event logs**. Int J Softw Tools Technol Transfer. 2007.

SANTOS, Antonio Raimundo dos. **Metodologia científica: a construção do conhecimento**. 5. ed. Rio de Janeiro: DP&A, 2002.

SILVA, Edna L.; MENEZES, Ester M. **Metodologia da pesquisa e elaboração de dissertação**. 3ª Edição. UFSC/PPGEP/LED. Universidade Federal de Santa Catarina. Florianópolis, 2001.

SILVA, Sérgio L. **Informação e competitividade**: a contextualização da gestão do conhecimento nos processos organizacionais. **Ci. Inf., Brasília**, v. 31, no 2, p. 142-151, mai. – ago. 2002.

SMITH, H. & FINGAR, P. **Business process management**: the third wave. MKPress, 2003.

THE PROM FRAMEWORK (2009). [http:// www.prom.sourceforge.net](http://www.prom.sourceforge.net). Acesso em 02 jul.2009.

THOM, L.H.; CHIAO, C.; YOCHPE, C.; **Padrões de workflow para reuso em modelagem de processos de negócio**. Instituto de Informática – Universidade Federal do Rio Grande do Sul. 2006.

VALERIANO, D. L. **Gerência em projetos**: pesquisa, desenvolvimento e engenharia. São Paulo: Makron Books, 1998.

VERGARA, S. C. **Projetos e relatórios de pesquisa em administração**. São Paulo: Atlas, 2009.

VERNADAT, F.B. **Enterprise modeling and integration**: principles and applications. Chapman & Hall, London. 1996.

WESKE, Mathias. **Business process management**: concepts, languages, architectures. Springer-Verlag Berlin Heidelberg, 2007.

WEN, Lijie; WANG, Jianmin; SUN, Jiaguang. **Detecting implicit dependencies between tasks from event logs**. In School of Software, Tsinghua University. 2006.

WORKFLOW MANAGEMENT COALITION. **The Workflow Reference Model**, 1995. Disponível em: <http://www.wfmc.org/standards/model.htm>. Acesso em: 16 julho 2009.

WORKFLOW MANAGEMENT COALITION. **Workflow Management Coalition Audit Data Specification**, 1998 Document Number WFMC-TC-1015,1998. 22 Sep 1998 Version 1.1 Páginas 14 a 20.