



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UM MÉTODO PARA DESCOBERTA DE REGRAS DE NEGÓCIO ATRAVÉS DE MINERAÇÃO

Raphael Crierie da Silva Castro

Orientadores

Prof^ª. Dr^ª. Fernanda Araujo Baião

Prof^ª. Dr^ª. Flávia Maria Santoro

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2009

**UM MÉTODO PARA DESCOBERTA DE REGRAS DE NEGÓCIO ATRAVÉS
DE MINERAÇÃO**

Raphael Cerie da Silva Castro

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA
OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-
GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO
DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO
EXAMINADORA ABAIXO ASSINADA.

Aprovada por:

Fernanda Araujo Baião, D. Sc. – UNIRIO

Flávia Maria Santoro, D. Sc. – UNIRIO

Astério Kioshi Tanaka, Ph. D. – UNIRIO

Eber Assis Schmitz, Ph. D. – UFRJ

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2009

C355 Castro, Raphael Cerie da Silva.
Um método para descoberta de regras de negócio através de mineração / Raphael Cerie da Silva Castro, 2009.
103f.

Orientador: Fernanda Araujo Baião.

Co-orientador: Flávia Maria Santoro.

Dissertação (Mestrado em Informática) – Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2009.

1. Regras de negócio. 2. Mineração de dados. 3. Mineração de processos. I. Baião, Fernanda Araujo. II. Santoro, Flávia Maria. III. Universidade Federal do Estado do Rio de Janeiro (2003-). Centro de Ciências Exatas e Tecnologia. Curso de Mestrado em Informática. III. Título.

CDD – 005.5

A minha família

Tente mover o mundo. O primeiro passo será mover a si mesmo.

Platão

Agradecimentos

Primeiramente, agradeço às minhas orientadoras, Fernanda Araujo Baião e Flávia Maria Santoro, pela oportunidade e confiança depositada em mim há três anos e, em especial, pelos ensinamentos que obtive ao conviver com elas como mestrando. Considero-as verdadeiros exemplos de competência, inteligência e garra; exemplos estes que levarei para o resto da vida.

À Escola de Informática Aplicada da UNIRIO, instituição onde ingressei sete anos atrás. São dignos de destaque os professores Astério Tanaka, Márcio Barros, Renata Araújo, Angelo Ciarlini, Luis Amâncio e Alexandre Andreatta.

Ao Núcleo de Pesquisa e Prática em Tecnologia (NP2Tec), do qual faço parte. Agradecimentos especiais para Ercilia Moreira, Leonardo Guerreiro Azevedo e Andréa Magalhães.

Aos professores do Liceu Nilo Peçanha, C. M. Ernani Faria e da E. M. Prof^a Zulmira Mathias Netto Ribeiro, especialmente as professoras Célia, Carla e Eliana, que contribuíram de forma decisiva na minha formação.

A todos os “companheiros de luta” do Programa de Pós-Graduação em Informática (PPGI), em especial ao Leonardo Gatti e à Lícia.

A Leonardo Matriciano, pelo apoio e liberação do trabalho nos momentos críticos, e ao amigo Adriano Simões por “segurar a barra” durante as minhas ausências.

Aos meus amigos e sócios de vida! Diego Fernandes, Raphael Martins, Bruno Arruda, Sylvestre Mergulhão e Ubiratan Ávila.

Aos meus pais que sempre apoiaram e incentivaram meus estudos, afinal, eles são a origem de tudo...

A minha querida esposa, pela força nos momentos mais difíceis, quando pensei que não fosse conseguir chegar até o final. Agradeço também pela paciência e compreensão nas longas noites e finais de semana de estudos. Dani, este trabalho é especialmente dedicado a você.

CASTRO, Raphael Cerie da Silva. **Um Método Para Descoberta de Regras de Negócios Através de Mineração**. UNIRIO, 2009. 105 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

Regra de Negócio é uma diretriz para influenciar ou guiar o comportamento do negócio, que torna explícitos os conceitos e regras envolvidos com as atividades realizadas na organização, uniformizando a visão sobre o negócio. Devido a estas características, as regras de negócio tornam-se muito importantes para as organizações. No entanto, a falta de documentação sobre as regras de negócio em vigor, ou a divergência entre a documentação existente e a realidade da organização, são problemas muito frequentes. Este trabalho apresenta um método para identificação e representação das regras de negócio em uso em uma organização. O método extrai as regras de negócio a partir de *logs* de eventos dos sistemas de informação da organização, fazendo uso de técnicas e conceitos de mineração adequados a cada tipo de regra de negócio. O método foi implementado através de extensões sobre o *framework* ProM. A aplicação do método em um estudo de caso em ambiente simulado mostrou a sua viabilidade na obtenção de regras de negócio do tipo sentenças de ação de autorização, sentenças de ação de condição e sentenças estruturais.

Palavras-chave: Regras de negócio; mineração de dados; mineração de processos.

ABSTRACT

Business Rule is a guideline to influence or to guide the conduction of business in an organization; it makes concepts and rules involved with organization activities explicit, thus providing a uniform view of the business activities. As such, business rules become very important for organizations, but nevertheless, however, the lack of business rules documentation, or the gap between the existing documentation and actual business rules being observed during activities execution, are very frequent problems. This work presents a method for identifying and representing business rules in an organization. The proposed method extracts business rules from event logs generated by the information systems, through the use of data mining techniques carefully chosen to handle each type of business rule. The identified business rules are then translated to SBVR and presented to business analysts. The method was implemented on top of the Pro-M framework. The results of a case study in a simulated environment showed the method applicability for identifying and representing condition action assertions rules, authorization action assertions rules and structural assertions rules.

Keywords: Business Rules; data mining; process mining.

Índice

1	Introdução	1
1.1	Motivação.....	1
1.2	Problema	2
1.3	Hipótese.....	3
1.4	Objetivos	3
1.5	Organização da Dissertação	4
2	Regras de Negócio	5
2.1	Importância das Regras de Negócio.....	5
2.2	Aplicações das Regras de Negócio	6
2.3	Classificação das Regras de Negócio.....	7
2.3.1	Sentenças Estruturais	9
2.3.2	Sentenças de Ação	10
2.3.3	Derivação	12
2.4	Descoberta das Regras de Negócio	13
2.4.1	Trabalhos Relacionados	14
2.5	Representação de Regras de Negócio	17
2.5.1	Bases do Metamodelo SBVR.....	21
2.5.2	Classificação das Regras de Negócio em SBVR	23
2.5.3	Palavras-chave e Representação das Regras de Negócio em SBVR	24
3	Descoberta de Conhecimento	28
3.1	Mineração de Dados.....	30
3.1.1	Classificação	30
3.1.2	Estimativa	32
3.1.3	Predição.....	32
3.1.4	Agrupamento (Clusterização)	33
3.1.5	Associação	33

3.2	Mineração de Processos	34
3.2.1	Relevância da Mineração de Processos	34
3.2.2	Uso da Mineração de Processos.....	35
3.2.3	Conceitos Sobre Mineração de Processos.....	35
3.2.4	Tipos de Mineração de Processos	36
3.2.5	Premissas para a Execução da Mineração de Processos.....	37
3.2.6	Perspectivas para a Mineração de Processos	39
3.2.7	Framework ProM e o Mining Xml (Mxml).....	40
4	Método para Descoberta de Regras de Negócio Através de Mineração.....	44
4.1	Regras de Negócio e os Sistemas de Informação.....	44
4.2	Arquitetura Proposta	46
4.3	Método Baseado na Arquitetura Proposta.....	48
4.3.1	Etapa 1: Registro de Execução e Necessidades dos Logs de Sistemas de Informação	49
4.3.2	Etapa 2: Conversão para Log de Eventos em Formato Padronizado.....	50
4.3.3	Etapa 3: Descoberta de Padrões e Regras	50
4.3.3.1	Descoberta de Sentenças de Ação de Autorização	51
4.3.3.2	Descoberta de Sentenças de Ação de Condição	52
4.3.3.3	Descoberta de Sentenças Estruturais	54
4.3.4	Etapa 4: Tradução para Linguagem de Representação de Regras de Negócio 55	
5	Implementação do Método	56
5.1	A Implementação do Método.....	56
5.1.1	Implementação da Etapa 2	57
5.1.2	Implementação dos recursos necessários para a Etapa 3 e 4	58
5.1.2.1	Aplicação do Método para Descoberta de Sentenças de Ação de Autorização.....	60
5.1.2.2	Exemplo de Sentenças de Ação de Autorização Descobertas	62

5.1.2.3	Aplicação do Método para Descoberta de Sentenças de Ação de Condição	63
5.1.2.4	Exemplo de Sentenças de Ação de Condição Descobertas	65
5.1.2.5	Aplicação do Método para Descoberta de Sentenças Estruturais	66
5.1.2.6	Aplicação do Método Sobre Outros Tipos de Regras de Negócio	69
6	Avaliação da proposta	70
6.1	Método de Avaliação	70
6.2	Cenário da Avaliação	71
6.3	O Processo de Negócio Utilizado	71
6.4	Geração do Log de Eventos	75
6.5	Aplicação do Método e Resultados Obtidos	79
6.5.1	Resultados para as Sentenças de Ação de Autorização	79
6.5.2	Resultados para as Sentenças de Ação de Condição.....	80
6.5.3	Resultados para as Sentenças Estruturais.....	83
6.5.4	Avaliação dos Resultados	84
6.5.5	Conclusões	86
7	Conclusão.....	87
7.1	Comparações com Outros Trabalhos	87
7.2	Contribuições	88
7.3	Limitações e Trabalhos Futuros	89
7.4	Considerações Finais.....	90
	Referências	91
	Apêndice A – Geração de dados com a ferramenta CPN Tools.....	102

Índice de Figuras

Figura 2.1 - Classificação das regras de negócio.	9
Figura 2.2 - Evolução das regras de negócio nos códigos legados.....	16
Figura 2.3 - Visão do “Mantra” das Regras de Negócio suportado pelo metamodelo...	21
Figura 2.4 - Os cinco aspectos de SBVR e suas relações.....	23
Figura 3.1 - As cinco etapas de KDD. Fonte (FAYYAD et al., 1996a).....	29
Figura 3.2 - Realidade e o fluxo da mineração de processos.	36
Figura 3.3 - Exemplo de log bem formado, seguindo as pré-condições da Mineração de Processos.	38
Figura 3.4 - Alguns resultados de mineração baseados nas perspectivas de processos (a) e organizacional (b e c) (AALST & WEIJTERS, 2005).	40
Figura 3.5 - Representação visual do padrão Mxml.....	41
Figura 4.1 - Instâncias de uma atividade onde o valor do atributo “Cliente Antigo?” influencia o valor de “Valor de Desconto” devido à existência de uma regra de negócio.	45
Figura 4.2 – Representação do método de descoberta de regras de negócio. Os retângulos representam os módulos da arquitetura e os círculos numerados representam os passos do método definido nesta pesquisa.	47
Figura 4.3 - Exemplo de informações de contexto em um subconjunto de log de eventos.	50
Figura 4.4 - Log de Eventos contendo papel do usuário e nome da atividade em destaque.	52
Figura 4.5 - Exemplo de atributos de contexto de atividade.	52
Figura 4.6 - Conjunto de valores possíveis de atributos de contexto.	53
Figura 5.1 - O método para descoberta de regras de negócio com os conceitos e ferramentas utilizados na implementação.....	57
Figura 5.2 - DER para a conversão para Mxml (EindhovenTU, 2009, MALLENS, 1997).....	58
Figura 5.3 - Detalhe da tela do ProM: Tela de seleção do algoritmo (Plug-in).....	60
Figura 5.4 - Parte do processo base utilizado como exemplo (MEDEIROS & GÜNTHER, 2005).....	62
Figura 5.5 - Árvore de decisão obtida através do algoritmo J48.....	64

Figura 6.1 - Processo de Negócio adaptado utilizado na avaliação.....	72
Figura 6.2 - Detalhe de parte do processo adotado nessa avaliação e modelado na ferramenta CPN Tools (CPNTOOLS, 2007).....	76
Figura 6.3 - Representação de instâncias de uma atividade qualquer, exemplificando a falta de consistência dos atributos entre as instâncias (a) e a manutenção da consistência (b).	77
Figura 6.4 - Representação de listagens na ferramenta CPN Tools, com destaque para a lista de código de campos.....	77
Figura 6.5 - Representação de funções na ferramenta CPN Tools, com destaque para a função que define se um projeto é liberado.....	78
Figura 6.6 - Em destaque exemplo do log gerado e, ao fundo, uma parte dos arquivos gerados pela simulação.....	79
Figura 6.7 - Funções incluídas no modelo criado na ferramenta CPN tools, para controle de atributos de alguns contextos.....	81
Figura 6.8 - Duas regras relacionando os mesmos atributos de contexto.....	85

Índice de Tabelas

Tabela 2.1 - Classificação para regras de negócio.....	7
Tabela 2.2 - Abordagens para descoberta de regras de negócio.....	13
Tabela 2.3 - Tabela comparativa de linguagens de representação de regras de negócio.	18
Tabela 2.4 - Palavras-chave em SBVR representando quantificação.	24
Tabela 2.5 - Palavras-chave em SBVR representando Operações Lógicas.	25
Tabela 2.6 - Palavras-chave em SBVR representando Operações Modais.	25
Tabela 2.7 - Outras palavras-chave em SBVR.	26

Lista de Abreviaturas Utilizadas

RNs	Regras de Negócio
KDD	Knowledge Discovery in Databases – Descoberta de Conhecimento em Banco de Dados
TI	Tecnologia de Informação
SVBR	Semantics of Business Vocabulary and Business Rules
BRG	Business Rule Group
SWLR	Semantic Web Rule Language Combining OWL and RuleML
OCL	Object Constraint Language
OMG	Object Management Group
MOF	Metadata Object Facility
XML	eXtensible Markup Language
XMI	XML Metadata Interchange
PIM	Plataform Independente Model
PSM	Plataform Specific Model
MDA	Model Driven Architecture
BPM	Business Process Management
ERP	Enterprise Resource Planning
Mxml	Mining Xml
ProM	Process mining framework
CPN	Coloured Petri Nets
OLAP	On-Line Analytical Processing
RuleML	Rule Markup Language

1 Introdução

Neste primeiro capítulo será apresentada a visão geral da dissertação, incluindo a motivação para a pesquisa, a relevância do tema proposto e uma visão sobre a importância das regras de negócio e sobre os problemas relativos à sua documentação.

1.1 Motivação

As regras de negócio são importantes porque guiam o funcionamento de uma organização, tornando explícitos as regras e os conceitos envolvidos com as atividades realizadas, uniformizando a visão sobre o negócio realizado. Além disso, sua documentação oferece importante fonte de informação para geração e manutenção de modelos tecnológicos.

Apesar de sua importância e visão já conhecida e consensual de que deve existir uma representação conceitual unificada das regras de negócio de um determinado domínio (ROSS, 2003, ROSS, 2000), muitas organizações não possuem as suas regras de negócio documentadas. Esta situação ocorre em decorrência de vários fatores, como dificuldade na formalização das mesmas, rápida evolução dos ambientes empresariais, alto custo de manutenção da documentação gerada, existência de um grande legado de regras já implementadas em sistemas de informação de difícil extração (BRG, 2003, ROSS, 2003).

A descoberta e documentação das regras de negócio de uma organização fornecem ao gestor da organização, por um lado, uma base formal e rica de conhecimento para servir como referência sobre as regras existentes na organização – especialmente em cenários em que a organização não possui documentação alguma sobre as suas regras de negócio – e, por outro lado, um conjunto de regras que pode ser utilizado para auditar se estas estão de acordo com as regras documentadas na organização – nos casos em que esta documentação exista. Neste último caso, as

divergências porventura encontradas entre os dois conjuntos poderiam indicar regras que não estão documentadas, regras desatualizadas na documentação existente ou implementações de sistemas de informação que estão permitindo a violação de tais regras. Em todos os casos, a descoberta e documentação produzem um importante subsídio para a tomada de importantes decisões por diversos papéis em uma organização.

Diversas pesquisas vêm sendo realizadas sobre a descoberta de regras de negócio, e é possível observar uma tendência por mecanismos (semi-) automáticos a partir do código fonte de aplicações (SIQUEIRA, 2002, PUTRYCZ & KARK, 2007, WANG *et al.*, 2008) ou de dados gerados pelo uso de sistemas de informação, como, por exemplo, a descoberta de regras sobre dados endógenos de SGBDs proposta por SULAIMAN (2002). Esta tendência com foco em sistemas ocorre por que o comportamento da organização tem sido cada vez mais controlado e registrado por sistemas de informação que dão suporte às suas atividades cotidianas, tornando estes um importante recurso de onde as regras podem ser extraídas. No que se refere à documentação, foram construídas e aplicadas diversas propostas para representar as regras de negócio (BRG, 2000, HALLE, 2002, ROSS, 2003, OMG 2006c, OMG, 2008), o que expõe a importância de armazená-las de forma adequada e torná-las explícitas dentro de uma organização. Através do compartilhamento dos conceitos e comportamentos de determinada organização para todas as pessoas que a compõem, a documentação das regras uniformiza o conhecimento, evitando contradições. Dentre estas propostas de documentação das regras de negócio, apresenta-se uma importante preocupação com relação ao entendimento das regras de negócio por pessoas, em especial por especialistas do negócio, que não possuem formação na área de TI (ROSS, 2003, OMG, 2008), pois as regras precisam ser entendidas por tais profissionais para serem aplicadas de forma adequada no cotidiano organizacional.

1.2 Problema

O problema abordado nessa dissertação refere-se a como realizar a descoberta de regras de negócio implementadas nos sistemas de informação presentes em uma organização, e representá-las de forma legível para análise de profissionais especialistas do negócio. Há na literatura pesquisas com o objetivo de descobrir as regras de negócio de forma automatizada (PUTRYCZ & KARK, 2007, POO, 1999, LEITE &

CERQUEIRA, 1995, SULAIMAN, 2002, WANG *et al.*, 2008, SIQUEIRA, 2002), e ao se analisar estes trabalhos publicados, incluindo os mais recentes (PUTRYCZ & KARK, 2007, WANG *et al.*, 2008), é possível perceber que as soluções apresentadas para a descoberta de regras de negócio possuem caráter específico, sendo dependentes, por exemplo, da linguagem de programação com a qual o sistema de informação do qual se extrairão as regras foi escrito, o que restringe a utilização destas.

1.3 Hipótese

Considerando-se que as informações geradas pela execução dos sistemas de informação em uma organização são comumente (ou possivelmente) registradas em logs de eventos, os quais possuem dados importantes sobre o comportamento e utilização destes sistemas e, conseqüentemente, sobre os processos de negócio apoiados por estes sistemas, a hipótese deste trabalho é de que:

“Os logs de execução dos sistemas de informação de uma organização possuem informações suficientes para tornar possível a identificação e representação das regras de negócio da organização.”

1.4 Objetivos

O objetivo fundamental desta dissertação é definir um método que viabilize a elaboração, de forma (semi-) automática, da documentação sobre as regras de negócio em vigor em uma organização, tendo como entrada os dados extraídos da execução de sistemas de informação em funcionamento na organização.

Como objetivos específicos, este trabalho propõe uma arquitetura e um método para descoberta e representação de regras de negócio. A descoberta se dá através da aplicação de técnicas e conceitos de mineração de processos e de mineração de dados sobre informações registradas nos logs de eventos dos sistemas de informação. As regras descobertas são então representadas na linguagem SBVR (*Semantics of Business Vocabulary and Business Rules*), a qual possui formalismo suficiente para evitar ambigüidades, sem, no entanto, perder legibilidade. A definição do método contemplou a escolha das técnicas de mineração adequadas para cada tipo de regra de negócio existente. A arquitetura proposta foi implementada para descoberta das regras de negócio do tipo sentenças de ação de autorização, sentenças de ação de condição e

sentenças estruturais. O método proposto foi avaliado quanto à sua viabilidade através de um estudo de caso, no qual foram identificadas e representadas diversas regras de negócio dos tipos endereçados, a partir de um *log* de eventos gerado artificialmente, com base em um processo de negócio de um domínio real de uma grande organização.

1.5 Organização da Dissertação

Esta dissertação está organizada em cinco capítulos. No Capítulo 2, são apresentados conceitos sobre regras de negócio, onde elas podem ser aplicadas e como elas podem ser classificadas e representadas. No Capítulo 3, são apresentados conceitos sobre o processo de descoberta de conhecimento, havendo um aprofundamento em mineração de dados e de processos. Após isso, no Capítulo 4, é finalmente apresentado o método para a descoberta de regras de negócio proposto e sua arquitetura. O Capítulo 5 apresenta a implementação do método e expõe detalhes sobre como o método foi implementado para 3 tipos de regras de negócio: sentenças de ação de autorização, sentenças de ação de condição e sentenças estruturais. O Capítulo 6 apresenta a avaliação sobre a aplicação do método, expondo resultados obtidos para cada tipo de regra tratado e uma análise sobre os resultados obtidos. Por fim, no Capítulo 7 são apresentadas conclusões obtidas através da aplicação de todo o método e são apontados trabalhos futuros.

2 Regras de Negócio

Este capítulo tem como objetivo apresentar o que são Regras de Negócio, explicando porque são importantes, onde são aplicadas e conceitos relacionados.

2.1 Importância das Regras de Negócio

A convivência em sociedade necessita de um conjunto de normas que regulem as atividades individuais para que estas não comprometam outros indivíduos e a própria sociedade. Segundo MIRABETE (2003) uma das tarefas do Estado é regular a conduta dos cidadãos por meio de normas objetivas sem as quais a vida em sociedade seria praticamente impossível. São assim estabelecidas regras para regulamentar a convivência entre as pessoas e as relações destas com o próprio Estado, formando o denominado Direito Objetivo, que exterioriza a vontade do Estado quanto à regulamentação das relações sociais entre os indivíduos, entre organismos do Estado e entre uns e outros (MIRABETE, 2003). Disso resulta que é lícito um comportamento que está autorizado ou não está vedado pelas normas jurídicas. É um sistema de limites aos poderes e faculdades do cidadão, que está obrigado pelo dever de respeito aos direitos alheios ou do Estado. Este mesmo conceito amplo, que regula toda a vida em sociedade, é aplicado para fins mais específicos como pode ser observado nas práticas esportivas e também no ramo empresarial, onde apresenta papel fundamental na organização das atividades cotidianas (MORGADO *et al.*, 2004). Em uma empresa, as regras que definem e determinam como o negócio deve ser guiado, tendo a característica de manter sua estrutura e influenciar quaisquer aspectos inerentes a ele, são denominadas Regras de negócio (OMG, 2006a).

As regras de negócio são, basicamente, o conhecimento que o negócio de determinada organização gerou sobre si própria (MORGADO *et al.*, 2007) e precisam ser sólidas, protegidas e gerenciadas. Este conhecimento sobre o negócio tem se tornado

muito importante no cenário competitivo instalado na atualidade, onde cada organização necessita se diferenciar de seus concorrentes através de detalhes, o que torna fundamental cada atividade envolvida com o negócio e vem servindo de combustível para a pesquisa e para a evolução das atividades referentes à organização e gerência do que acontece nas organizações.

Existem diversas definições de regras de negócio, segundo ROSS (2003), Regra de Negócio é uma diretriz para influenciar ou guiar o comportamento do negócio, representada por uma sentença que define ou qualifica algum aspecto relativo a ele, representando o conhecimento dos especialistas (BRG, 2000, ROSS, 2003). Já o *Business Rules Group* define as regras de negócio como sentenças que influenciam ou conduzem o comportamento do negócio de uma organização. Sua intenção é definir a estrutura, controlar ou influenciar o comportamento do negócio (BRG, 2000). Com base nestas visões sobre o que são regras de negócio, a definição adotada nessa dissertação é que regras de negócio são regras que guiam o comportamento e uniformizam os conceitos no negócio.

2.2 Aplicações das Regras de Negócio

Em uma organização, as regras de negócio representam o conhecimento gerado por especialistas do negócio e, devido a esta característica, elas representam um importante ativo intelectual para as organizações, gerando a necessidade de organizá-las em uma documentação formal com o objetivo de, efetivamente, tornar essas regras uma fonte de divulgação de conhecimento organizacional. Isto viabilizaria um fortalecimento do tratamento uniforme e eficiente do negócio por parte dos profissionais envolvidos, devido à facilitação do entendimento do negócio (MORGADO *et al.*, 2007).

Regras de negócio têm crescido em importância nos últimos anos (KOVACIC & GROZNIK, 2004) e, segundo o Manifesto de Regras de Negócio (BRG, 2003), são essenciais para os modelos de negócios e para os modelos de tecnologia (modelos de sistemas e de implementação). Sob a perspectiva de TI, estas regras podem ser vistas como requisitos prioritários para o desenvolvimento de aplicações de suporte ao negócio, elas tendem a assegurar que os objetivos da organização estejam alinhados aos sistemas (MORGADO *et al.*, 2007), portanto são consideradas importantes sob o ponto de vista do desenvolvimento de sistemas (DIAS *et al.*, 2006).

Ross (2003) apresenta uma abordagem que afirma que as regras de negócios devem ser geradas sem a intervenção de profissionais da Tecnologia da Informação (TI), e devem ser registradas em linguagem natural de forma que possa ser entendida por todos no escopo do negócio, tendo como objetivo registrar o conhecimento do negócio e expô-lo onde ele é necessário. Existem trabalhos no sentido de formalizar as Regras de Negócio de forma que o conhecimento armazenado possa ser utilizado para execução de inferências e processamentos automatizados (OMG, 2008, ZIMBRÃO *et al.*, 2002).

De uma forma geral, as regras de negócio servem para controlar as atividades do negócio de uma organização e são requisitos fundamentais utilizados por profissionais de TI para o desenvolvimento de sistemas que apóiam o negócio (MORGADO *et al.*, 2007).

2.3 Classificação das Regras de Negócio

HALLE (2002) apresenta diversas classificações de regras de negócio, apresentadas na Tabela 2.1.

Tabela 2.1 - Classificação para regras de negócio (HALLE, 2002).

Fonte	Classificação das regras de negócio
Business Rules Group (BRG, 2000)	<ul style="list-style-type: none"> • Derivação: Uma expressão de conhecimento que é derivada de outro conhecimento do negócio <ul style="list-style-type: none"> ○ Cálculo matemático ○ Inferência • Sentença estrutural: Uma definição de conceito ou expressão de um fato que expressa algum aspecto da estrutura da organização. Inclui os termos e os fatos que são construídos a partir desses termos. <ul style="list-style-type: none"> ○ Termos ○ Fatos • Sentenças de ação: Uma expressão de restrição ou condição que limita ou controla as ações em determinada organização. <ul style="list-style-type: none"> ○ Autorização ○ Condição ○ Restrição de integridade
ROSS (2001, <i>apud</i> HALLE, 2002)	<ul style="list-style-type: none"> • Fatos • Termos • Regras • Restrições • Derivações • Inferências • Controle de tempo • Sequência • Heurísticas
DATE (2000, <i>apud</i> HALLE, 2002)	<ul style="list-style-type: none"> • Restrição • Restrição de estado • Restrição de transição • Estímulo - resposta • Derivação • Computação

		<ul style="list-style-type: none"> • Inferência
DATE	(2000, <i>apud</i> HALLE, 2002)	Chris Date ainda propõe outro esquema para restrições que são baseadas na estrutura dos dados em si. <ul style="list-style-type: none"> • Restrições de domínio • Restrições de coluna • Restrições de tabela (restrições dentro de uma tabela) • Restrições de banco de dados (restrições entre duas ou mais tabelas)
Versata	Inc. (PHILLIPS, 2002, <i>apud</i> HALLE, 2002)	<ul style="list-style-type: none"> • Regras de integridade referencial • Derivações • Validação • Restrição • Ação/evento • Regras de apresentação
USoft	Inc. (MALLENS, 1997, <i>apud</i> HALLE, 2002)	<ul style="list-style-type: none"> • Regras de restrição: Restrições do negócio sobre a informação que será armazenada. Representa o que não é permitido. • Regras de comportamento: Como o sistema deve se comportar em determinadas situações. O que ele deve fazer automaticamente? • Regras dedutivas: Como informações devem ser derivadas ou calculadas. • Regras de apresentação: Como o sistema se apresenta para o usuário, como trabalha e como tarefas estão organizadas. • Regras de instrução: Como o usuário deve operar o sistema em certas situações.

As classificações apresentadas por DATE (2000, *apud* HALLE, 2002), Versata Inc. (PHILLIPS, 2002, *apud* HALLE, 2002) e USoft Inc. (MALLENS, 1997, *apud* HALLE, 2002) apresentam características que as aproximam da área tecnológica, que podem ser observadas pela presença de grande preocupação com restrições de informação e por características como validação e apresentação. Por outro lado, as classificações apresentadas pelo *Business Rules Group* (BRG, 2000) e por ROSS (2001, *apud* HALLE, 2002) possuem características mais voltadas para o negócio em si, sem se preocuparem com conceitos e recursos tecnológicos. Além disso, ambas baseiam-se na geração das regras de negócios por especialistas do negócio, sem a intervenção de profissionais de TI (Ross, 2003, BRG, 2008), o que demonstra o foco claro no negócio. Dentre estas duas classificações, foi adotada como base para esta pesquisa a classificação do *Business Rules Group* (BRG, 2009), uma organização não comercial formada por pesquisadores importantes, que atua desde 1989 com foco em regras de negócio em organizações públicas e privadas, devido ao foco no negócio.

O *Business Rules Group* classifica as regras de negócio de 3 formas, como pode ser visto na Figura 2.1 e detalhados a seguir:

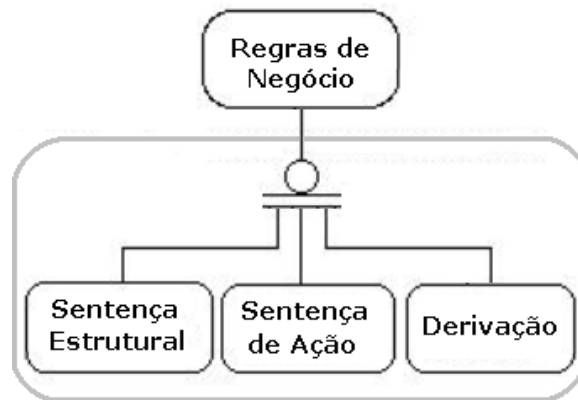


Figura 2.1 - Classificação das regras de negócio (BRG, 2008).

2.3.1 Sentenças Estruturais

Uma sentença estrutural é uma sentença que expressa algum aspecto estrutural da organização: um conceito ou um relacionamento entre conceitos. Devido a isso são freqüentemente representadas graficamente por modelos entidade-relacionamento e são as regras base para a definição de todas as outras regras de negócio.

- Exemplos (no contexto de curso de pós-graduação):
 - Aluno (conceito);
 - Secretaria (conceito);
 - Aula (conceito);
 - Disciplina (conceito);
 - Aluno assiste à aula (relacionamento);
 - Aula pertence à disciplina (relacionamento);

As sentenças estruturais subdividem-se em termos e fatos, conforme definição abaixo:

- a) **Termos:** São palavras ou frases que representam um significado específico do negócio. Eles se dividem em termos de negócio (ligados a contextos específicos e precisam ser definidos através de fatos) e termos comuns (significado aceito independente de conceito).

- Exemplos (no contexto de curso de pós-graduação):
 - (termo comum): Aluno;
 - (termo comum): Professor;
 - (termo de negócio): Graduação;
 - (termo de negócio): Disciplina;

- b) **Fatos:** Representam uma associação entre dois ou mais termos. Os fatos são

chamados de fatos base, quando representam um caso do negócio, ou de fatos derivados, quando construídos baseados em uma ou mais regras de negócio através de derivação.

- Exemplos (no contexto locação de automóveis):
 - (Base): “Um modelo automóvel está em uma categoria de automóveis”;
 - (Base): “Uma categoria de automóveis possui uma taxa”;
 - (Derivado): “O valor da locação de um automóvel é a taxa da categoria de automóveis da qual o modelo do automóvel pertence”;

Além disso, as sentenças do tipo Fato possuem uma classificação, na qual estas podem ser divididas em: Atributo, Generalização e Participação:

- Atributo: É um fato que expressa um termo que descreve algo que existe em outro termo.
 - Ex: “Nome é atributo de pessoa”;
- Generalização: Fato que descreve que um termo representa um subconjunto de outro termo.
 - Ex: “Operário é um empregado”;
- Participação: Fato que associa um conjunto de termos de forma significativa para o negócio;
 - Ex: “Um grupo de locação é composto por modelos de carros”;

2.3.2 Sentenças de Ação

Sentença de ação é uma restrição ou condição que limita ou controla as ações na organização, em outras palavras, controla a dinâmica do negócio.

- Exemplos (no contexto de uma locadora de automóveis):
 - “Um locatário precisa ter CPF e CNH”;
 - “Se o locatário possui CPF, possui CNH e forneceu seus dados, então ele pode alugar o automóvel”;

As sentenças de ação são classificadas de acordo com de acordo com as relações que elas efetuam entre os conceitos do negócio em: Restrição de Integridade, Condição e Autorização; e são caracterizadas de acordo com a forma como elas controlam a interação entre regras como sendo dos tipos: Habilitadora, Temporizadora e Executora. Abaixo estão definições e exemplos destas 3 classes e dos 3 tipos.

- Restrição de integridade: Classificação que se refere a uma sentença que precisa SEMPRE ser verdadeira. Normalmente, elas representam elementos (atributos ou relacionamentos, por exemplo) obrigatórios. Abaixo, um exemplo no contexto de locadora de automóveis:
 - “Um carro precisa ser registrado”;
 - “Um locador precisa ter CPF”;
- Condição: Esta classificação refere-se a uma sentença que se for verdadeira, outra regra de negócio existe ou é aplicada. Abaixo segue um exemplo no contexto de instituição de ensino:
 - “Se o aluno concluiu o Ensino Médio e apresentou documento do serviço militar, então o aluno pode efetuar matrícula”;
- Autorização: Classificação que se refere a uma sentença que restringe quem pode executar determinada ação. A seguir são apresentados exemplos no contexto de instituição de ensino:
 - “Somente professor pode aprovar aluno”;
 - “Somente aluno inscrito pode assistir aula”;
- Habilitadora: Tipo que se refere a uma sentença que, se é verdadeira, permite a existência de determinado objeto ou execução de determinada ação. Abaixo segue um exemplo no contexto de vendas:
 - “As regras de avaliação de crédito só podem ser aplicadas se o cliente possui CPF”.
- Temporizadora: É um tipo que representa uma sentença que executa um controle de tempo sobre determinada sentença, agindo como um simples contador assinalando o momento de parada ou como um alarme (efeito da regra ocorre quando o alarme “toca”). Abaixo um exemplo no contexto de uma firma de segurança:
 - “A partir das 22hs a entrada no prédio é permitida somente com autorização”.
- Executora: Tipo que se refere a uma sentença que causa a execução de uma ou mais ações. Abaixo um exemplo no contexto de locadora de automóveis:
 - “Se um cliente atrasa a entrega por 3 meses, então se deve recuperar o automóvel”.

2.3.3 Derivação

Derivação são regras de negócio formadas pela relação de uma ou mais regras de negócios existentes, que geram como resultados os fatos derivados, sendo que um fato derivado necessariamente precisa ser gerado através de uma derivação. Em outras palavras, derivação é um tipo de regras de negócio que tem como principal função a geração de fatos derivados. Abaixo está descrito um exemplo de derivação no contexto de uma operação de venda:

- “Preço final do produto = valor do item + quantidade” (derivação);
- “Preço final é baseado em valor do item e quantidade” (fato derivado);

As regras de negócio classificadas como derivação se dividem em dois tipos: Cálculo matemático e inferência. Abaixo estão definições e exemplos deles:

- Cálculo matemático: Derivação que utiliza algum cálculo matemático do negócio para produzir um fato derivado. Abaixo pode ser visto um exemplo no contexto de locadora de automóveis:
 - “Valor do seguro = taxa de seguro X número de dias da locação – desconto” (derivação);
 - “Valor do seguro é baseado na taxa de seguro, no número de dias da locação e no desconto” (fato derivado).
- Inferência: Derivação que utiliza indução lógica ou dedução. Novamente, pode ser visto um exemplo no contexto de locadora de automóveis:
 - “A taxa de aluguel na locação é inferida pela taxa de aluguel para o automóvel que está sendo alugado, que é inferida pela taxa de aluguel da categoria do automóvel que está sendo alugado” (derivação);
 - “A taxa de aluguel de uma locação é a taxa da categoria da qual o automóvel pertence” (fato derivado).

Todos os exemplos apresentados acima foram escritos em linguagem natural, o que deixa o texto de definição claro para quaisquer pessoas, mais traz dificuldade para a realização de algum processamento automático e torna fácil a ocorrência de ambiguidades e inconsistências. Devido a isso, torna-se importante representar as regras de negócio com certo formalismo. Neste sentido, foram desenvolvidas diversas propostas para a representação das regras de negócio (RULEML, 2009, ROSS, 2003, OMG, 2006b, OMG, 2008).

2.4 Descoberta das Regras de Negócio

Segundo MORGAN (2002), existem 3 principais abordagens para a descoberta de regras de negócios nas organizações, as quais podem ser aplicadas de forma independente ou associada, dependendo do contexto organizacional. Uma destas abordagens é a chamada **Análise Estática**, que é caracterizada por uma cuidadosa análise das informações da documentação existente para obter as regras de negócio. A análise estática é considerada a melhor abordagem quando há documentação relevante disponível. Outra abordagem são as **Sessões Interativas** onde analistas e especialistas do negócio se reúnem para explorar as áreas onde o conhecimento da organização não está documentado de alguma forma. E por fim, a última abordagem é a **Descoberta Automática de Regras**, que pode ser utilizada para buscar regras através de análise computacional de fontes de dados adequadas.

Tabela 2.2- Abordagens para descoberta de regras de negócio (MORGAN, 2002).

Fonte	Análise Estática	Sessões Interativas	Descoberta Automática
Documentação	Alta	Moderada	Não confiável
Conhecimento Tácito	Não aplicável	Alta	Não aplicável
Sistemas	Baixa	Moderada	Alta
Registros do negócio	Depende da fonte	Baixa	Depende da fonte

A Tabela 2.2 apresenta algumas diferenças entre as abordagens onde é possível observar que cada uma das abordagens tem como ponto forte de aplicabilidade uma fonte diferente, o que indica que essas abordagens tendem a ser complementares. De forma diferente do que está exposto nessa tabela, POO (1999) afirma que é possível extrair, através de uso de descoberta automática, regras de documentos (ele se refere a casos de uso) de forma confiável, desde que estes documentos contenham informações sobre a política da empresa, além dos passos de determinada tarefa de sistema. Outro importante ponto a se observar é que os registros de negócio não necessariamente possuem informações ou estrutura necessárias para a descoberta de regras, seja através da abordagem de análise estática ou pela descoberta automática.

MORGAN (2002) afirma que apesar de soar como uma grande idéia, a descoberta automática possui problemas devido ao conhecimento sobre regras de negócio ser expresso comumente em linguagem natural e sem uma estrutura consistente. Mas, de forma contraditória, há muitos anos pesquisas vêm sendo realizadas

(PUTRYCZ & KARK, 2007, CRERIE *et al.*, 2008, POO, 1999, LEITE & CERQUEIRA, 1995, SULAIMAN, 2002, WANG *et al.*, 2008, CRERIE *et al.*, 2009, SIQUEIRA, 2002) com o objetivo de descobrir as regras de negócio de forma automatizada com base em artefatos distintos, como documentação de sistemas, código fonte e estrutura ou registros das bases de dados. Baseando-se nisto, é possível afirmar que a descoberta automática pode sim ser realizada, pois todos estes artefatos foram construídos para suprir necessidades do negócio e, por consequência, têm que estar em conformidade com regras de negócio existentes na organização, o que gera um lastro de comportamento através do qual as regras de negócio podem ser descobertas.

Independente da forma com a qual se pretende automatizar a descoberta das regras de negócio, o processo de KDD (*Knowledge Discovery in Databases*, ou Descoberta de Conhecimento em Bancos de Dados) (FAYYAD *et al.*, 1996b) se enquadra como um importante meio para a realização desta descoberta, visto que as regras, neste contexto, estão em meio à grande quantidade de informação e a busca por padrões nessas informações podem indicar a presença de um comportamento causado pelo controle exercido por regras de negócio. Devido a isso o Capítulo 3 será voltado para a descoberta de conhecimento e serão expostos conceitos de mineração de dados e mineração de processos.

2.4.1 Trabalhos Relacionados

Existem diversos trabalhos voltados para a descoberta automática de regras de negócio. SULAIMAN (2002) apresenta uma abordagem para a descoberta de fluxos de trabalho e regras de negócio baseando-se em dados endógenos, que são dados gerados pelos Sistemas de Gerenciamento de Banco de Dados (SGBDs). Segundo ele, a aplicação de operações *On-Line Analytical Processing* (OLAP) e de minerações de dados sobre os logs de bancos de dados contendo dados endógenos auxilia no entendimento do processo de negócio do sistema que os gerou devido à obtenção parcial do workflow e das regras de negócio presentes no sistema. São apresentados como premissas para sua proposta os seguintes itens:

- Logs do Banco de dados são orientados por assunto. O assunto é o armazenamento das transações de banco de dados;
- Logs de banco de dados são integrados. Uma vez que o log registra o histórico da transação a informação obtida é semanticamente relacionada com suas aplicações de origem;

- Logs de banco de dados evoluem com o tempo. Eles registram as modificações dos objetos a cada alteração de estado;
- Logs de banco de dados não são voláteis. Eles têm que permanecer preservados para a recuperação de falhas, auditoria de sistemas e para controle de concorrência.

A abordagem proposta por Sulaiman propõe que a análise dos dados endógenos deve ser feita com base no catálogo do SGBD (as estruturas, relações e restrições formam uma ontologia parcial do banco de dados) e no log das transações de banco de dados que registram ocorrências de criação, recuperação, alteração ou retirada de itens de dados e relaciona-as com a transação de origem, sendo o termo transação, no contexto do trabalho do Sulaiman, um conjunto de operações de banco de dados interdependentes, onde caso ocorra alguma falha em uma das operações, todas as outras devem ser desfeitas.

SIQUEIRA (2002) propôs a extração de regras de negócio de sistemas legados em ambiente Unisys utilizando o Source Inspector. O trabalho apresenta uma proposta de modelo de classes a partir de sistemas legados em UNISYS que possuam bancos de dados DMSII e código fonte COBOL. Essa abordagem visa resolver o problema de divergência entre as regras documentadas no início do sistema para as que efetivamente passam a existir no sistema após anos de sua utilização devido a diversas mudanças ocorridas no negócio e que não são documentadas, ficando perdidas no código legado (Figura 2.2). Para alcançar seu objetivo de extrair as regras de negócio a partir do código e Cobol, foram desenvolvidas heurísticas específicas para o ambiente Unisys.

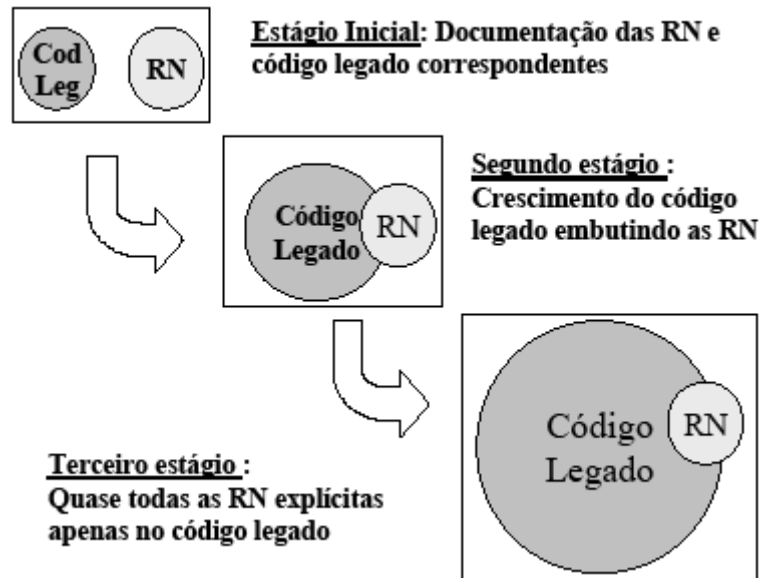


Figura 2.2 - Evolução das regras de negócio nos códigos legados. (SIQUEIRA, 2002).

WANG *et al.* (2008) em sua pesquisa também propõem uma abordagem para solucionar o problema de extração de regras. Eles afirmam que, com o tempo, desenvolvedores tendem a não documentar mais as alterações que são requisitadas e estas passam a estar somente no código fonte da aplicação e, para recuperá-las, é necessário um extrator de regras.

PUTRYCZ & KARK (2007) possuem um trabalho sobre a extração de regras de negócios a partir da engenharia reversa do código fonte de sistemas legados, de forma que estas regras possam ser facilmente interpretadas por especialistas do negócio.

ROZINAT & AALST, (2006) identificam, através de mineração de processos, “pontos de decisão” (*decision points*) que podem ser considerados regras de negócio do tipo sentenças de ação, pois eles exibem atividades que são restringidas por regras existentes.

De uma forma geral, estes trabalhos são voltados para ambientes muito específicos, gerando como consequência uma abrangência de aplicação igualmente restrita. Enquanto SIQUEIRA (2002), WANG *et al.* (2008) e PUTRYCZ & KARK (2007) buscam regras sobre códigos fontes em linguagens específicas, SULAIMAN (2002) e ROZINAT & AALST, (2006) buscam regras de forma muito restrita, sendo que este último nem cita diretamente o termo regra de negócio.

Devido a isso, muitos dos sistemas de informação, sejam sistemas legados ou

não, ficam fora do escopo destas propostas e torna-se importante o desenvolvimento de formas mais abrangentes de se obter as regras de negócio que independam de implementações específicas e que possam ser aplicadas a sistemas desenvolvidos em plataformas distintas, de forma a se adequar ao ambiente heterogêneo e complexo presente nas grandes organizações. Nesse contexto, esta pesquisa visa aplicar técnicas e conceitos de KDD (*Knowledge Discovery in Databases*), que serão discutidas nos próximos capítulos, com o objetivo de propor uma solução mais ampla para a descoberta de regras de negócio.

2.5 Representação de Regras de Negócio

Propostas sobre a representação das regras de negócio foram desenvolvidas em diversas pesquisas tais como OCL (*Object Constraint Language*) (OMG, 2006c), *RuleSpeak* (ROSS, 2003) proposta por Ronald Ross, *RuleML* (Rule Markup Language) (RULEML, 2009), SWLR (*Semantic Web Rule Language Combining OWL and RuleML*) (HORROCKS, 2004) e SBVR (*Semantics of Business Vocabulary and Business Rules*). Na Tabela 2.3 pode ser visualizada uma comparação de algumas formas de representação de regras de negócio, incluindo estas citadas, onde são avaliados os seguintes critérios:

- Qual o objetivo que deu origem ao modelo de representação? Onde ele pode ser utilizado?
- O modelo de representação foi construído especificamente para representar regras de negócios? Ou foi construído com foco em outro contexto e adaptado para regras de negócio?
- O modelo de representação pode ser facilmente entendido por especialistas do negócio? O objetivo é saber se pessoas que não são de TI e que possuem amplo conhecimento do negócio tem facilidade para compreender o que é representado.
- O modelo possui tratamento eficiente para ambiguidades? Impedindo-a de ocorrer ou mesmo tornando fácil a identificação de ocorrências ambíguas.

Tabela 2.3- Tabela comparativa de linguagens de representação de regras de negócio.

	RuleSpeak	OCL	RuleML	SWLR	SBVR
Qual seu objetivo?	Representação de regras de negócio.	Representação de expressões, que podem ser regras, sobre modelos UML.	Representação de regras, incluindo regras de negócio.	Representação de regras, incluindo regras de negócio.	Representação de regras de negócio
Foi construído para regras de negócio?	Sim	Algumas fontes indicam que sim, mas em seu documento de especificação (OMG, 2006c) menciona genericamente que foi desenvolvido como uma linguagem de modelagem de negócio.	Não especificamente , mas contempla as regras de negócio.	Não especificamente , mas contempla as regras de negócio.	Sim
Pode ser facilmente entendida por especialistas do negócio?	Sim	Não	Sim, pois possui representação em linguagem natural.	Não	Sim
Trata bem Ambiguidade?	Sim	Sim	Não	Sim	Sim
Observações	Possui um conjunto de melhores práticas para representar as regras de negócio. Foi uma das linguagens usadas como referencia para a definição do padrão SBVR.	A própria OMG, organização que mantém a especificação OCL, possui outra linguagem para representação de regras de negócio: SBVR.	Linguagem baseada em XML para representação e intercâmbio de regras com foco na web.	Propõe a combinação de RuleML com OWL, para acrescentar semântica a regras e ontologias.	Foi definida pela OMG como padrão para representação de regras de negócio, e tem como base as idéias defendidas pelo Business Rules Group.

Apesar da existência de diversas frentes de pesquisa sobre a representação de regras de negócio, até o ano de 2006, nenhuma delas havia sido adotada como padrão de representação das regras de negócio. Diante da inexistência de um consenso para definição de uma terminologia para representação de regras de negócio, a OMG (*Object Management Group*) (2009) gerou um pedido de propostas (OMG, 2003), solicitando a criação de um metamodelo com o objetivo de permitir que especialistas do negócio definam políticas e as regras que conduzem as organizações usando a mesma linguagem utilizada no cotidiano do negócio, e efetuar a documentação de uma forma clara, sem

ambiguidades e em um formato que seja facilmente convertido para outros formatos. Os itens requisitados neste pedido de propostas da OMG são:

- Um metamodelo com representação MOF (*Metadata Object Facility*) (OMG, 2006b) para especificação de regras de negócios por especialistas do negócio;
- Um metamodelo para a captura de vocabulários e definição de termos utilizados nas regras de negócio;
- Uma representação XML para regras de negócio e vocabulários, baseada em XMI (OMG, 2005) para permitir a interoperabilidade destes conceitos entre ferramentas diferentes que manipulem regras de negócio;

Com base no que foi pedido pela OMG, foi apresentada uma proposta para a especificação SBVR (*Semantics of Business Vocabulary and Business Rules*, também presente na Tabela 2.3), que foi aceita e teve sua primeira versão disponibilizada em janeiro de 2008, definindo o vocabulário e regras para a documentação da semântica dos vocabulários, dos fatos e das regras de negócio.

Esta especificação é interpretável na lógica de predicados (com uma pequena extensão na lógica modal), e é aplicável no domínio de vocabulários e regras de negócios de forma independente do ramo da atividade do negócio, podendo ser utilizada para qualquer organização.

Dentre os grupos de profissionais que podem fazer uso do SBVR, existe uma visão diferenciada de como o metamodelo pode ser utilizado e de como ele pode ser útil (KAMADA, 2006). Abaixo segue a caracterização de como cada grupo pode utilizar e se beneficiar do metamodelo SBVR:

- Analistas de Negócio: Possuem visão do negócio da organização ou de parte desta visão, se for considerado que uma comunidade dentro da organização tem uma visão parcial do negócio. Eles precisam negociar com gerentes a inclusão de novos conceitos, termos e regras de negócio no vocabulário e os especificar precisamente. Porém, para isto, não precisam de qualquer conhecimento detalhado da formulação lógica de SBVR, pois se assume que o tratamento desta formulação lógica de SBVR será realizado por facilidades incorporadas nas ferramentas de suporte a vocabulários e regras de negócio, tais como, a existência de

modelos (templates), configuração de restrições, e verificação de consistência.

- Desenvolvedores de sistemas: A OMG considera que o intercâmbio de padrões é de grande importância e, por isso, o principal requisito em relação ao desenvolvimento de ferramentas aderentes ao SBVR é garantia de compatibilidade e aderência a MOF (OMG, 2006b) e XMI (OMG, 2005). Em relação às ferramentas para desenvolvimento e manutenção de vocabulários e regras de negócio, o principal requisito refere-se à necessidade de suportar o compartilhamento de conjuntos de termos e expressões de negócio com significados precisamente definidos entre múltiplas comunidades, bem como a formulação lógica a partir destes. Além disso, outro aspecto importante que as ferramentas devem incorporar é um processo de desenvolvimento de vocabulários e regras de negócio de modo a assegurar a integração e o compartilhamento com outras comunidades. O SBVR necessita de dois tipos de ferramentas:
 - Ferramentas para intercâmbio de vocabulários e regras de negócio entre diferentes plataformas;
 - Ferramentas para o desenvolvimento e manutenção dos vocabulários e regras de negócio de determinada comunidade.
- Especialistas em semântica, lingüística e lógica: O metamodelo SBVR fornece capacidades lógicas, matemáticas e lingüísticas que permitem a transformação de vocabulários e regras de negócios contidos em modelos de negócios para modelos PIM (*Platform Independent Model*) e PSM (*Platform Specific Model*), duas subdivisões da arquitetura MDA (*Model Driven Architecture*) (OMG, 2009). Além disso, é possível estruturar uma variedade de sentenças em linguagem natural em construções SBVR e verbalizar estruturas SBVR em termos de sentenças em linguagem natural. Assim, é possível projetar algoritmos que assegurem a integridade de vocabulários e regras de negócio em documentos de intercâmbio e na tradução entre documentos de intercâmbio e modelos internos das ferramentas. Com isto, assegura-se também a formalização matemática e lógica dos modelos internos usados nas ferramentas de desenvolvimento de vocabulário e regras de negócio.

2.5.1 Bases do Metamodelo SBVR

O SBVR é altamente adequado para a utilização por especialistas do negócio, tendo sido projetado para ser usado para finalidades de negócio, independente de projetos de sistemas da informação, seguindo os preceitos do Manifesto de Regras de Negócio (BRG, 2003) do *Business Rules Group* que serviu de base conceitual para o trabalho de especificação do metamodelo SBVR (OMG, 2008). Segundo esse manifesto do *Business Rules Group*, as regras são baseadas em fatos, que se baseiam nos conceitos expressos por termos, que, por sua vez, expressam conceitos do negócio. Além disso, seguindo o chamado “mantra” de regra de negócios, os fatos fazem afirmações sobre os conceitos e as regras restringem e dão suporte aos fatos (Figura 2.3).

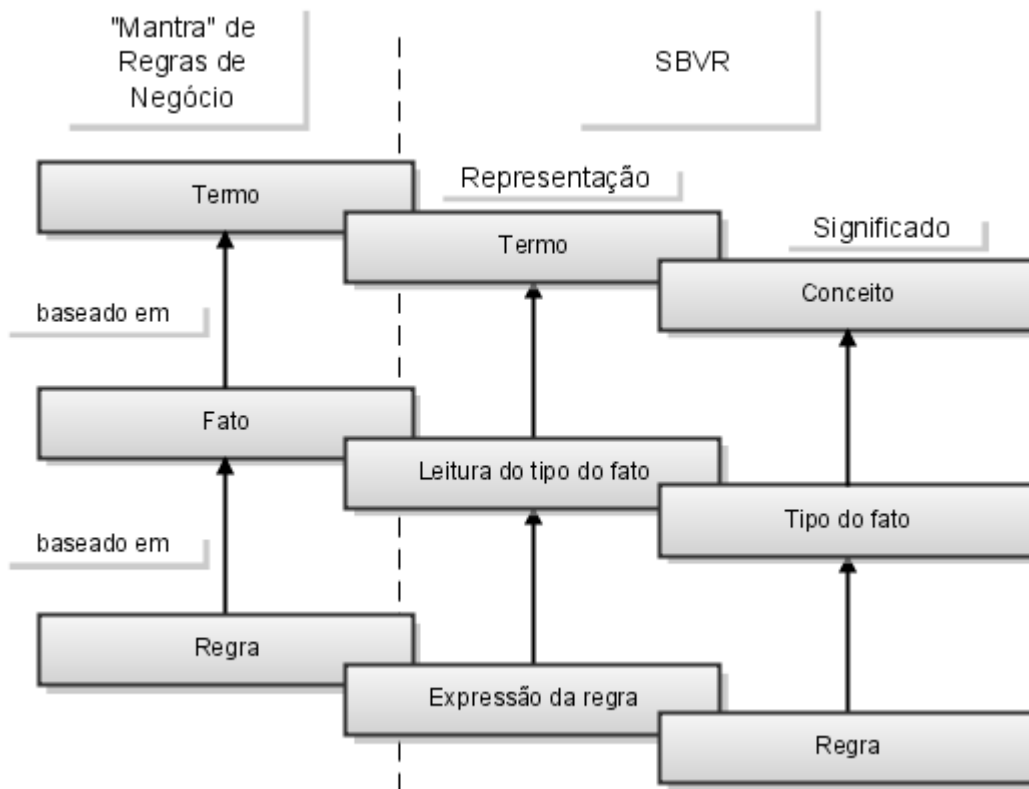


Figura 2.3 - Visão do “Mantra” das Regras de Negócio suportado pelo metamodelo SBVR (OMG, 2008).

O metamodelo SBVR pode ser analisado a partir de cinco aspectos (Figura 2.4), descritos a seguir:

- Comunidade (*Community*): A comunidade é a base para o vocabulário de negócio. Comunidades de importância primária são compostas pelas empresas para as quais as regras de negócio estão sendo estabelecidas,

mas outras comunidades como, por exemplo, o setor de mercado destas empresas, empresas parceiras, organizações de padronização e autoridades reguladoras, também podem e devem ser consideradas durante o levantamento e documentação das regras de negócio.

- Conjunto de Significados Compartilhados (*Body of Shared Meanings*): Uma determinada comunidade tem um conjunto de elementos cujos significados são compartilhados, compreendendo os conceitos, os fatos e as regras de negócio. Para que estes significados compartilhados possam ser discutidos e validados, devem ser expressos, mas SBVR trabalha com o conceito de separação do significado do negócio de qualquer forma particular de expressão.
- Formulação Lógica (*Logical Formulation*): A formulação lógica do metamodelo SBVR fornece sintaxe formal, abstrata e independente de linguagens, e possui como objetivo a captura da semântica de determinado conjunto de elementos com significados compartilhados. Ela possui suporte a diversas formas de representação linguística, tais como, substantivos e expressões verbais fazendo uso de voz ativa e passiva. A formulação lógica suporta duas das características essenciais do metamodelo SBVR:
 - Mapeamento do conjunto de elementos com significados compartilhados para vocabulários usados pelas comunidades existentes no contexto do negócio;
 - Mapeamento para XMI, padronizando o intercâmbio de conceitos, fatos e regras negócio entre ferramentas que suportam o SBVR.
- Representação de Negócio (*Business Representation*): Comunidades que compartilham determinados significados, que representam conceitos e regras de negócios, necessitam que estes sejam representados em vocabulários aceitos e efetivamente utilizados por cada uma delas. Estes vocabulários podem estar em linguagens naturais distintas, em linguagens artificiais como o XML ou UML, ou ainda em subconjuntos especializados de alguma linguagem natural, como as usadas por médicos ou advogados. O SBVR suporta a representação de elementos de negócio associando símbolos a conceitos ou fatos, como está

exemplificado a seguir:

- Símbolos para conceitos: para termos como "loja", "cliente", "produto";
- Símbolos para fatos (freqüentemente expressões verbais) como "vender", "deve entregar" ou "deve registrar".
- **Lógica Formal (*Formal Logic*):** SBVR possui como base da sua fundamentação teórica a lógica formal, suportando formulações lógicas e estruturas dos elementos que possuem significados compartilhados. A lógica de predicados de primeira-ordem é utilizada, com algumas extensões limitadas em lógica modal com o objetivo de expressar obrigação, proibição, e necessidades.

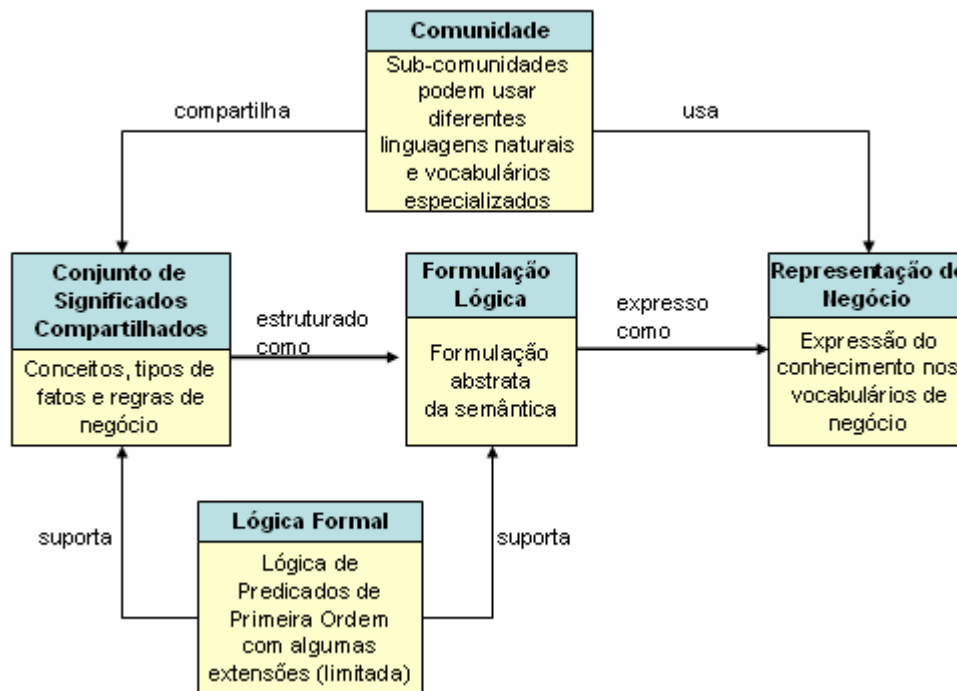


Figura 2.4 - Os cinco aspectos de SBVR e suas relações (KAMADA, 2006).

2.5.2 Classificação das Regras de Negócio em SBVR

O padrão SBVR divide as regras de negócio em dois grandes grupos: as regras de negócio operativas e as regras de negócio estruturais. As regras estruturais são regras rígidas que especificam como o negócio é organizado e os artefatos com os quais ele opera, ou seja, documentam a estrutura do negócio. Por exemplo, no contexto de comércio eletrônico, uma regra de negócio estrutural pode ser definida da seguinte

forma:

- Um cliente possui:
 - Uma operação de venda em andamento.
 - Uma operação de venda concluída nos últimos 2 anos.

As regras operativas são regras que guiam o comportamento das pessoas no negócio, são diretrizes para as atividades executadas na organização, introduzindo obrigações ou proibições. De forma diferente das regras estruturais, as regras de negócio operativas podem, potencialmente, ser violadas diretamente por pessoas do negócio durante a execução de suas atividades, pois elas podem representar uma sugestão de como as pessoas devem agir. Por exemplo, no contexto de instituição de ensino, uma regra de negócio operativa pode ser formulada da seguinte forma:

- Um aluno que não realizou inscrição em determinada disciplina não pode assistir à aula desta disciplina.

2.5.3 Palavras-chave e Representação das Regras de Negócio em SBVR

O padrão SBVR possui um conjunto de palavras-chave para que formulações lógicas sejam expressas em inglês estruturado. Nas tabelas expostas a seguir, contendo estas palavras-chave, as letras “n” e “m” referem-se números inteiros e as letras “p” e “q” estão representando expressões de preposições. As palavras-chave podem ser divididas em palavras de quantificação (Tabela 2.4) que visam delimitar a idéia de quantidade de elementos ou ocorrências de algum evento, palavras de operações lógicas (Tabela 2.5) que expressam relações lógicas entre conceitos, palavras de operações modais (Tabela 2.6) que visam expressar modos como possibilidade e necessidade da lógica modal, e outras palavras-chave (Tabela 2.7) não classificadas.

Tabela 2.4 - Palavras-chave em SBVR representando quantificação.

Quantificação	
Each	Quantificação universal
Some	Quantificação existencial
at least one	Quantificação do tipo “ao menos um”
at least n	Quantificação do tipo “ao menos n”
at most one	Quantificação do tipo “no máximo um”
at most n	Quantificação do tipo “no máximo n”
exactly one	Quantificação do tipo “exatamente um”

exactly n	Quantificação do tipo “exatamente n”
at least n and at most m	Quantificação de faixa de valores
more than one	Quantificação do tipo “ao menos n” com $n = 2$

Tabela 2.5 - Palavras-chave em SBVR representando Operações Lógicas.

Operações Lógicas	
it is not the case that p	Negação lógica
p and q	Conjunção
p or q	Disjunção
p or q but not both	Disjunção exclusiva
if p then q	Implicação
q if p	Implicação
p if and only if q	Equivalência
not both p and q	Formulação lógica “nand”
neither p nor q	Formulação lógica “nor”
p whether or not q	Formulação lógica “whether-or-not”

Tabela 2.6 - Palavras-chave em SBVR representando Operações Modais.

Operações Modais	
it is obligatory that p	Obrigaç�o
it is prohibited that p	Obrigaç�o atrav�s de nega�o l�gica
it is necessary that p	Necessidade
it is impossible that p	Possibilidade atrav�s de nega�o l�gica
it is possible that p	Possibilidade
it is permitted that p	Permiss�o
... must ...	Obrigaç�o
... must not ...	Obrigaç�o atrav�s de nega�o l�gica
... always ...	Necessidade
... never ...	Possibilidade atrav�s de nega�o l�gica
... may ...	Possibilidade
... may ... only if p	Obrigaç�o sobre uma implica�o
it is permitted that q only if p	Obrigaç�o sobre uma implica�o
it is possible that q only if p	Possibilidade sobre uma implica�o

Tabela 2.7 - Outras palavras-chave em SBVR.

Outras Palavras-Chave	
The	1. Usado com uma designação para fazer uma referência pronominal a um uso anterior da mesma designação. É formalmente uma ligação para uma variável de uma quantificação.
	2. Introdução do nome de um elemento ou de uma descrição.
a, na	Pode ser uma quantificação universal ou existencial.
Another	É usada em conjunto com um termo usado na mesma declaração, indicando a existência de outro elemento e, portanto, representa uma quantificação existencial.
a given	Quantificação universal que expõe uma formulação lógica onde um dado é usado de forma que represente um elemento em um determinado momento. É utilizado para evitar ambigüidades com o uso da quantificação existencial "a".
That	1. Quando precede uma designação de um conceito substantivo, serve para referenciá-lo, como a palavra-chave "the".
	2. Quando utilizada após a designação para um conceito substantivo e antes de uma designação para um tipo de fato, ela é usada para introduzir uma restrição sobre coisas denotadas pela prévia designação baseada em fatos sobre ela.
	3. Quando seguida por uma declaração proposicional, é usada para introduzir informações sobre o elemento (nominalization ou objectification).
Who	O mesmo que o segundo uso da palavra-chave "that" mas usado para uma pessoa.
is of	"of" é usada como abreviação de "that is of". Para uma expressão "n has m" existe a expressão inversa "m is of n".
What	Usada para introduzir uma variável em uma projeção, bem como indica que uma projeção está sendo formulada.

Com base na aplicação destas palavras-chave é possível descrever as regras de negócio de um determinado contexto como, por exemplo, as seguintes regras no contexto de comércio eletrônico:

- It is necessary that each **VENDA** has at least one **ITEM**;
- It is obligatory that the **TEMPO_DE_ENTREGA** of each **VENDA** is at most **10 dias**.

3 Descoberta de Conhecimento

Com a constante evolução nos meios de armazenamento da informação, grande quantidade de informação tem sido sistematicamente coletada e armazenada por grandes organizações. A quantidade de informação armazenada ultrapassa a habilidade técnica e a capacidade humana na sua interpretação, situação que abriu espaço para uma nova geração de técnicas computacionais e ferramentas para suportar a extração de conhecimento útil de grandes e crescentes volumes de dados. Essas técnicas e ferramentas deram origem à área de Descoberta de Conhecimento em Banco de Dados (*Knowledge Discovery in Databases - KDD*) (FAYYAD *et al.*, 1996b).

Tradicionalmente, técnicas de descoberta de conhecimento em bases de dados (KDD) têm sido utilizadas para identificação de padrões novos, válidos, potencialmente úteis e compreensíveis em dados (FAYYAD *et al.*, 1996a). KDD é um processo composto por cinco etapas que tornam explícitos conhecimentos extraídos de grandes bases de dados. Essas etapas são (Figura 3.1):

- Seleção dos dados;
- Pré-processamento e limpeza dos dados;
- Transformação dos dados;
- Mineração de Dados (Data Mining); e
- Interpretação e avaliação dos resultados.

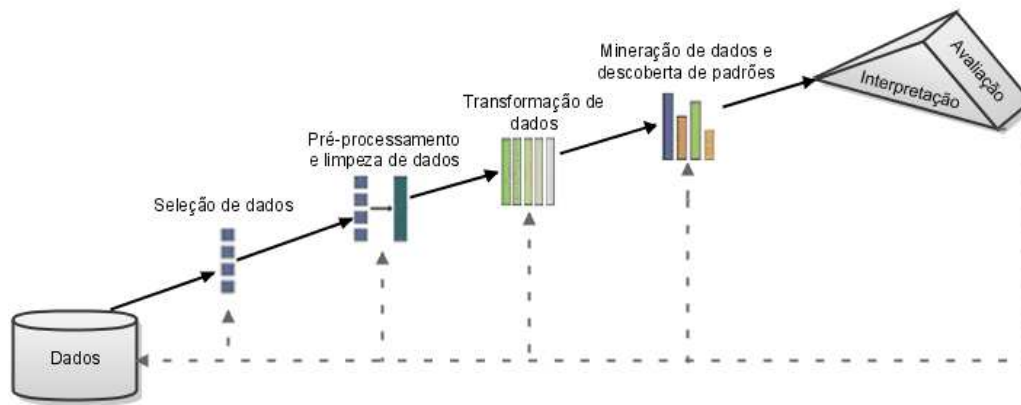


Figura 3.1 - As cinco etapas de KDD. Fonte (FAYYAD et al., 1996a).

Na etapa de seleção dos dados, primeira no processo de descoberta de conhecimento, é feita a seleção do subconjunto dos dados de um domínio que é composto por todas as variáveis (características ou atributos) e registros (casos) que devem ser utilizados na análise. Esta seleção, normalmente, é realizada por um especialista do domínio, e este processo pode incluir a obtenção de dados de uma série de fontes diferentes (*data warehouses*, planilhas, bases de sistemas legados, etc).

A etapa de pré-processamento e limpeza de dados, também é uma parte muito importante, pois a qualidade dos dados é algo determinante para obter eficiência dos algoritmos de mineração de dados. Dentre os importantes passos desta fase estão a eliminação de dados redundantes e inconsistentes, a recuperação de dados incompletos e a avaliação de possíveis *outliers* (dados discrepantes). Também nessa fase, a participação de um especialista do domínio é fundamental.

Já a etapa de mineração de dados, detalhada na seção seguinte, tem como objetivo a descoberta de regras e padrões ligados aos dados. E a última etapa, de interpretação e avaliação de resultados, é quando o conhecimento adquirido através da mineração de dados é apresentado ao especialista do domínio para ser interpretado e avaliado. Na situação em que o resultado não seja satisfatório, o processo deve retornar a algum dos estágios anteriores ou até mesmo recomeçar, conforme pode ser observado no fluxo da Figura 3.1.

A área de mineração de dados tem evoluído para tratar, por exemplo, tipos mais específicos de dados, como mineração em textos (*Text Mining*) (HAN & KAMBER, 2006) que visa descobrir padrões em textos em linguagem natural, e Mineração de Processos (*process mining*) (AALST & WEIJTERS, 2005) que tem como objetivo a descoberta de informações sobre o processo de negócio a partir de análises sobre dados

de execução de sistemas.

Nesse trabalho são utilizados conceitos de e técnicas de mineração de dados e de processos aplicados sobre logs de eventos de sistemas de informação com o objetivo de descobrir regras de negócio. A Seção 3.1 expõe detalhes sobre a mineração de dados e a Seção 3.2 é dedicada à mineração de processos.

3.1 Mineração de Dados

Mineração de dados é uma das fases da descoberta de conhecimento em bases de dados (KDD) (FAYYAD *et al.*, 1996a), e é considerada seu núcleo, pois é a mineração de dados que efetivamente extrai o conhecimento que está implícito nos dados. A mineração de dados é caracterizada pela descoberta de padrões em um determinado conjunto de dados através da aplicação de algoritmos.

Para HAN & KAMBER (2000), mineração de dados é um processo onde métodos inteligentes são aplicados para extrair padrões de dados e WITTEN & FRANK (2000), definem a de mineração de dados como o processo, automático ou semi-automático (maioria dos casos), de descoberta de padrões em dados. Para eles, os padrões descobertos precisam ser significativos e devem trazer algum benefício, usualmente podem ser uma informação preciosa para seu negócio que pode gerar retorno econômico. A definição de TAN *et al.* (2006) para mineração de dados afirma que ela é a tecnologia que mistura métodos tradicionais de análise de dados com algoritmos sofisticados para o processamento de grandes volumes de dados, e segundo a visão de HAND *et al.* (2001), a mineração de dados é a análise de conjunto de dados observacionais, frequentemente em grande volume, para buscar relações insuspeitas e sumarizar os dados em novas formas que são compreensíveis e úteis para o proprietário do dado. Comparando-se estas definições, é possível observar que, de uma forma geral, todas as visões sobre mineração de dados convergem para a descoberta de padrões relevantes em grandes volumes de dados.

As classes de problemas em que técnicas de mineração de dados podem ser aplicadas são denominadas tarefas de mineração. Nas seções seguintes serão expostos detalhes sobre algumas tarefas de mineração de dados mais comuns (LAROSE, 2005), utilizadas na busca e descoberta de conhecimento.

3.1.1 Classificação

A tarefa de classificação busca modelos (funções) que descrevem e distinguem classes ou conceitos (HAN & KAMBER, 2000, LAROSE, 2005). Neste tipo de tarefa cada registro da amostra dos dados é caracterizado por um conjunto de atributos, e deve haver um atributo de classificação (*class label attribute*), que pode ser dividido em classes ou categorias. Por exemplo, seja um atributo “concederEmprestimo” que pode ser classificado em “sim” ou “não”. O algoritmo de mineração de dados examina um grande volume de registros buscando explicitar a relação entre os valores dos demais atributos e cada categoria do atributo de classificação. Como exemplo, pode se descobrir que o atributo “concederEmprestimo” possui valor “sim” quando o atributo “salario_mensal” é maior que “15.000” e o atributo “status_empregado” possui o valor “ativo”.

A classificação é um processo supervisionado, ou seja, a amostra dos dados utilizada deve estar previamente classificada. O processo é dividido em duas etapas (HAN & KAMBER, 2000). Na primeira etapa (aprendizado), uma parte da amostra, denominada base de treinamento (*training data set*), é utilizada como espaço a ser percorrido pelo algoritmo de classificação na busca dos padrões, ou regras de classificação. Posteriormente, na etapa de teste, estas regras encontradas são aplicadas sobre o restante da amostra, denominada de base de testes (*test data set*), para testar a eficácia das regras encontradas e selecionar as que são consideradas “aceitáveis”.

Dentre as possíveis aplicações de tarefas de classificação em organizações encontram-se:

- Determinar se uma particular transação de cartão de crédito é fraudulenta;
- Diagnóstico de doenças;
- Identificar se determinadas movimentações financeiras e operações de compra representam desvio de verba;

As técnicas de mineração utilizadas usualmente para classificação são *k-nearest neighbour* (k vizinhos mais próximos), redes neurais e árvore de decisão. *K-nearest neighbour* é um método baseado em instâncias onde todas estas correspondem a pontos em um espaço n-dimensional (HAN & KAMBER, 2000). As redes neurais são sistemas computacionais baseados em ligações entre nós simples (neurônios), que são ligados formando uma rede de nós (HAN & KAMBER, 2000). As árvores de decisão são estruturas onde cada nó interno representa um teste sobre um atributo e cada ramo

representa um resultado e cada nó folha representa classes (HAN & KAMBER, 2000). O conhecimento representado nas árvores de decisão pode ser extraído e representado na forma de regras de classificação do tipo SE-ENTÃO.

3.1.2 Estimativa

As técnicas de estimativa servem, de forma similar à classificação (Seção 3.1.1), para descobrir modelos ou funções que determinem uma classe, sendo que se diferencia da classificação pelo fato de serem aplicadas sobre uma variável que é numérica ao invés de representar uma categoria ou valor textual. Os modelos são construídos através do uso de registros “completos”, que fornecem uma variável alvo para a qual se pretende estimar o valor, bem como através das variáveis pelas quais se obterá a relação para geração das estimativas (LAROSE, 2005). Por exemplo, existe o interesse em executar a estimativa da pressão sanguínea de um paciente, baseando-se na idade dele, em seu sexo, em seu índice de massa corpórea e no nível de sódio no sangue. O relacionamento da pressão sanguínea com as demais variáveis citadas forneceria um modelo de estimativas, através do qual se pode estimar a pressão de um paciente que acabou de chegar com base nas demais informações sobre ele, ou seja, o modelo pode ser aplicado a novos casos.

O campo de análises estatísticas fornece diversos métodos amplamente utilizados na realização de estimativas. Dentre eles se encontram a estimativa pontual, a estimativa por intervalo de confiança, regressão simples linear e correlação, e regressão múltipla. Na regressão linear os dados são modelados segundo uma equação linear (HAN & KAMBER, 2000). Regressão múltipla é uma extensão da regressão linear que relaciona uma variável a um vetor multidimensional (HAN & KAMBER, 2000).

3.1.3 Predição

As técnicas de predição também são similares às técnicas de classificação, diferenciando-se por seus resultados se referirem a valores futuros (LAROSE, 2005) e, por isso, são frequentemente utilizadas para realizar simulações de cenários alternativos. Dentre exemplos de tarefas de predição se encontram:

- Previsão de produtos no estoque para 6 meses no futuro;
- Previsão do aumento de mortes em acidentes aéreos se o fluxo de vôos diários for aumentado;

- Previsão do próximo campeão mundial de futebol com base nas estatísticas de todas as seleções;
- Previsão da variação do preço das ações de determinada empresa, baseando-se nos últimos dados da empresa e do mercado financeiro;

Qualquer um dos métodos de classificação e estimativa pode ser utilizado para a predição, desde que sob circunstâncias apropriadas. Dentre os métodos possíveis estão tradicionais métodos estatísticos de estimativa de pontos e estimativa por intervalo de confiança, regressão linear simples e correlação, e múltipla regressão, bem como redes neurais, árvores de decisão e métodos “k-nearest neighbor” (k vizinhos mais próximos).

3.1.4 Agrupamento (Clusterização)

Clusterização refere-se ao agrupamento físico ou abstrato de objetos em classes de objetos similares (HAN & KAMBER, 2000). Um cluster é um conjunto de registros onde cada elemento é similar a outro também pertencente ao mesmo cluster e diferente dos elementos de outros clusters. Diferentemente da classificação, na clusterização não existe uma variável alvo para se efetuar a mineração, pois o que se busca é a relação entre todas as variáveis disponíveis, e não há a intenção de descobrir ou estimar valores para as variáveis. O objetivo dos algoritmos de clusterização é buscar a relativa homogeneidade em cada cluster, fazendo com que a similaridade entre os elementos de um mesmo cluster seja maximizada e entre elementos de clusters diferentes seja minimizada.

Dentre as possíveis aplicações de tarefas de clusterização em organizações encontram-se as seguintes (LAROSE, 2005):

- Definição do mercado alvo para o nicho de um determinado produto para um pequeno negócio que não tem grande verba para utilização com marketing;
- No contexto de auditoria contábil, pode ser feita a segmentação de comportamentos financeiros;
- Determinar subgrupos de pacientes com sintomas similares.

3.1.5 Associação

Associação é a descoberta de regras de associações exibindo relações entre

valores de atributos que ocorrem frequentemente juntos em um determinado conjunto de dados (HAN & KAMBER, 2000). Dentre as possíveis aplicações de tarefas de associação em organizações, existe um típico exemplo: a análise do carrinho de compras, onde a análise sobre os hábitos de compra dos clientes é realizada através da busca por associações entre os diferentes itens que os clientes colocam em seu carrinho de compras. Associações descobertas como “sempre que um homem compra CD compra também um livro”, servem como importante fonte de informação para o desenvolvimento de estratégias de marketing ligadas a itens frequentemente comprados pelos seus clientes.

3.2 Mineração de Processos

Assim como a mineração de dados, a mineração de processos é mais uma abordagem para se obter conhecimento não-explícito através da análise de dados. A idéia básica da mineração de processos é a extração de conhecimento a partir de *logs* de eventos gerados por um sistema de informação. Até recentemente, as informações encontradas nestes logs de eventos eram raramente utilizadas para descobrir os processos (AALST & WEIJTERS, 2005) e a mineração de processos visa melhorar isso, provendo técnicas e ferramentas para descobrir processos e informações referentes às execuções destes.

Processo de negócio é um conjunto de atividades relacionadas e organizadas, através do qual uma organização deve ser estruturada, com o objetivo de produzir valor, como por exemplo alcançar algum objetivo estratégico existente. Uma das partes importantes da análise de negócios e da abordagem de processos de negócios são as Regras de Negócio, pois são diretrizes que visam influenciar ou guiar o comportamento do negócio.

Esta seção tem como objetivo apresentar o que é mineração de processos, expondo e explicando detalhes desta abordagem.

3.2.1 Relevância da Mineração de Processos

Muitos sistemas de informação da atualidade armazenam eventos relevantes de forma estruturada como, por exemplo, os sistemas de gerenciamento de workflow, que registram o início e o término de cada atividade, e os ERPs (Enterprise Resource Planning ou Sistemas Integrados de Gestão Empresarial) que registram todas as

transações (ALBERTINI *et al.*, 2007, GOEDERTIER & VANTHIENEN, 2005, AALST & WEIJTERS, 2005).

De uma forma geral, este tipo de sistema não controla as atividades que compõem o processo de determinada organização, mas sempre permitindo algum tipo de liberdade na execução das atividades implementadas nele, em outras palavras, permitindo que sejam executadas ações de forma diferente da que está definida no modelo de processos da organização. Em um ERP, por exemplo, os usuários não precisam seguir o modelo de referência (modelo de processos) completamente, sendo este apenas um guia para execução do trabalho (AALST & WEIJTERS, 2005). É muito freqüente também que existam vários sistemas de informação que dão suporte a atividades distintas de um mesmo processo de negócio.

Nestes cenários, gera-se o interesse por identificar como as pessoas trabalham atualmente. A mineração de processos busca, portanto, identificar como cada usuário executa seu trabalho e prover informação para comparação do modelo de processos especificado pela empresa com o que é efetivamente executado na organização, exibindo discrepâncias que podem ser utilizadas para a melhoria dos processos (AALST & WEIJTERS, 2005).

3.2.2 Uso da Mineração de Processos

A mineração de processos não é uma ferramenta de reengenharia de processos. Seu objetivo é o entendimento do que realmente está acontecendo na organização (GOEDERTIER & VANTHIENEN, 2005, AALST & WEIJTERS, 2005) e, é claro que devido a isso, torna-se importante num processo de reengenharia ou reestruturação dos processos (AALST & WEIJTERS, 2005). Em outras palavras, a mineração de processos é a aplicação de um conjunto de técnicas para identificar informações sobre a execução dos processos que podem ser visualizadas, por exemplo, na forma de um modelo de processos.

Desta forma a mineração de processos fornece uma interface que exhibe aos especialistas do negócio “donos do processo” ou aos gerentes do negócio uma visão real de como os processos estão sendo executados na organização, tornando mais simples a identificação de problemas e de possíveis pontos de melhoria.

3.2.3 Conceitos Sobre Mineração de Processos

A mineração de processos busca identificar os processos reais da organização,

isto é, os que são efetivamente colocados em prática (GOEDERTIER & VANTHIENEN, 2005). Além disso, nem todos os processos da organização podem ser identificados, somente aqueles que são registrados nos sistemas cujos logs serão utilizados na análise.

Na Figura 3.2 pode ser identificado como a mineração de processos se relaciona com outros recursos existentes nas organizações. Nela, pode ser visto que o modelo de processos baseia-se nas informações do negócio de uma determinada organização e, posteriormente, guiam a especificação e a implementação dos sistemas de informação dela. Estes sistemas, por sua vez, registram informações importantes sobre as atividades e eventos que são realizados através deles em logs de eventos que são utilizados no processo de mineração para que seja possível se obter uma visão do que realmente está ocorrendo na organização.

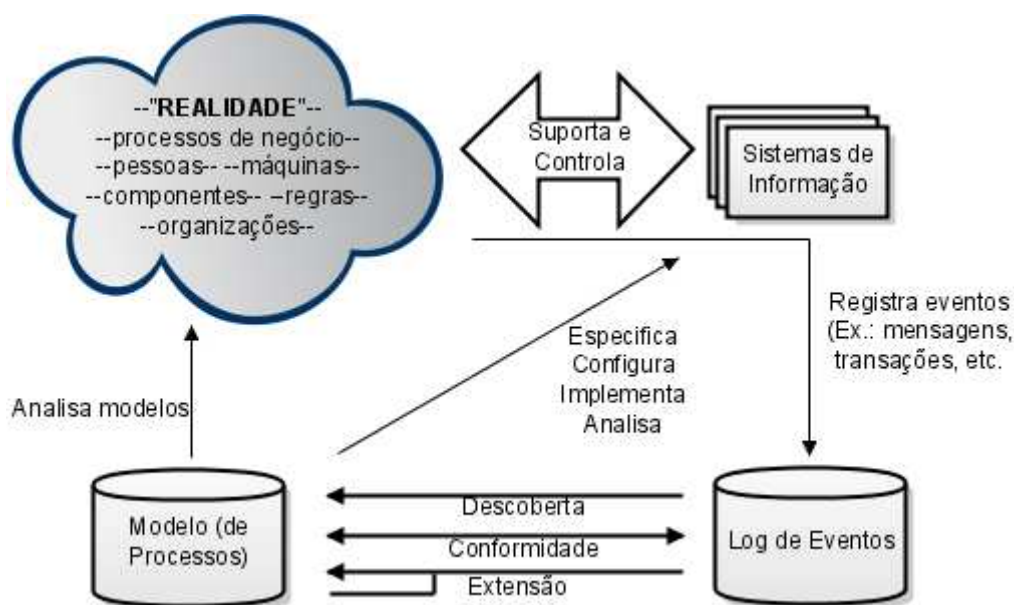


Figura 3.2 - Realidade e o fluxo da mineração de processos (AALST, GÜNTHER, 2007).

3.2.4 Tipos de Mineração de Processos

A mineração de processos pode ser utilizada para a realização de três tipos de ações sobre os processos da organização, com base no conhecimento retirado do log de eventos: Descoberta, conformidade (monitoramento) e extensão (melhoria) dos processos reais da organização (ALBERTINI *et al.*, 2007, GÜNTHER *et al.*, 2007). Abaixo segue uma breve descrição sobre cada uma destas ações:

- Descoberta: Não existe um modelo ou esquema para o processo e através

da mineração de processo o primeiro modelo é obtido. Ele deve ser utilizado para análise do processo da organização e deve ser armazenado para possíveis avaliações sobre a conformidade do processo no futuro;

- Conformidade: Existe um modelo ou esquema para o processo e através da aplicação de mineração de processos é possível verificar se esta documentação existente está sendo respeitada; A análise dos casos de não-conformidade pode expor a necessidade de treinamento para funcionários que não conhecem bem o processo da organização, pode expor falhas em sistemas que controlam o negócio e também podem indicar pontos de desatualização da documentação sobre o processo de negócio (havendo, neste caso, necessidade de extensão);
- Extensão: Existe um modelo ou esquema para o processo e esta documentação é enriquecida através das novas informações identificadas através da mineração de processos;

3.2.5 Premissas para a Execução da Mineração de Processos

Ao se falar em processos e mineração de processos, é necessário conceituar alguns itens. Abaixo segue uma lista destes conceitos e sua descrição:

- Atividades: São passos bem definidos no processo da organização ou, em outras palavras, ações realizadas no sistema ou no negócio que devem ser registradas no log de eventos a cada execução. Seu registro deve ser realizado independentemente se foram realizadas por uma pessoa ou disparadas de forma automatizada;
- Eventos: São as execuções de cada atividade, ou seja, se uma pessoa realizou 5 atividades, devem ser registrados 5 eventos no log, registrando as informações da execução de cada atividade. Cada evento registrado no log de eventos deve respeitar um conjunto de pré-condições que são necessárias para a realização da mineração de processos e serão expostas logo a seguir, nesta mesma seção;
- Casos de execução (ou simplesmente “Casos”): São instâncias de processos de negócio da organização;
- Originadores: São pessoas ou sistemas (no caso de atividades automatizadas) que iniciam a realização de determinada atividade, gerando um novo evento;

- Logs de eventos: Os logs de eventos são a fonte de informação fundamental para a mineração de processos. Eles são registros que armazenam, de forma ordenada no tempo, a sequência de eventos realizados e informações relativas a eles. Por exemplo, logs de auditoria de sistema são logs de eventos.

Existem importantes pré-condições relativas à geração do log de eventos para que a mineração de processos possa ser realizada. Tais pré-condições são referentes a características de formação que o log de eventos deve seguir para que seja possível extrair dele os dados de forma estruturada, executar as transformações necessárias e realizar a efetiva descoberta de conhecimento. Abaixo estão descritas estas características e a Figura 3.3 exemplifica um log bem formado:

- Cada evento no log refere-se a uma única atividade. Mesmo que duas atividades sejam realizadas de forma concorrente, como, por exemplo, a venda de um produto e a baixa dele no estoque, sempre deve haver eventos individuais para cada uma delas;
- Cada evento refere-se, também, a um único caso de execução. Definindo em outras palavras, não pode existir um evento ligado a mais de uma instância de processo;
- Cada evento deve conter um único originador (ou executor);
- Eventos devem possuir marcação de tempo e devem ser totalmente ordenados;

Id do caso	Atividade	Originador	Momento	Dados
(...)	(...)	(...)	(...)	(...)
caso 39	Enviar E-mail	João	23/03/2009 12:03	to=marcos@mail.com, ...
caso 30	Efetuar Registro	Rita	23/03/2009 14:20	name=Ana, idade=20, ...
caso 39	Montar Resposta	Marcos	23/03/2009 17:35	to=joao@mail.com, ...
caso 23	Registrar Participação	Maria	24/03/2009 08:30	evento=palestra BPM, ...
(...)	(...)	(...)	(...)	(...)

Figura 3.3 - Exemplo de log bem formado, seguindo as pré-condições da Mineração de Processos.

Através da análise da Figura 3.3, pode-se identificar a presença de 4 eventos e de 3 casos de execução (casos 39, 30 e 23). Percebe-se também que os 2 eventos do caso 39 são ligados às atividades ‘Enviar E-mail’ e ‘Montar Resposta’, que tiveram sua

execução originada, respectivamente, por 'João' e 'Marcos'. Além disso, os eventos encontram-se ordenados de forma temporal através da coluna 'Momento'.

3.2.6 Perspectivas para a Mineração de Processos

São distinguidas três diferentes perspectivas para a realização da mineração de processos: perspectiva do processo, perspectiva organizacional e perspectiva de casos (GÜNTHER *et al.*, 2007).

- Perspectiva de processos (“Como?”): Possui foco no fluxo de controle, ou seja, na ordenação e organização de atividades. O objetivo da mineração é buscar uma boa caracterização de todos os caminhos possíveis realizados no processo.
 - Ex.: Atividades C e D são sempre executadas em paralelo;
 - Ex.: A média de tempo de execução da atividade B é de 10 minutos;
- Perspectiva organizacional (“Quem?”): Possui seu foco nos originadores dos eventos, em como eles se relacionam e com o que estão envolvidos. O objetivo é encontrar qualquer estrutura na organização através da classificação das pessoas em termos de papéis e unidades da organização, ou exibir relações entre executores individuais (redes sociais).
 - Ex. 1: Júlio e Renata são da mesma equipe;
 - Ex. 2: Renata efetua em média 25 pedidos por dia;
- Perspectiva de caso (“O que?”): Tem o foco voltado para as propriedades de casos. Os casos de execução podem ser caracterizados por seu caminho no processo ou pelos originadores que trabalham sobre eles. Entretanto eles também podem ser caracterizados pelos valores dos dados ou atributos ligados a ele. Por exemplo, se um caso representa um pedido de reabastecimento de estoque, é interessante saber qual é o item que será reabastecido e qual é a quantidade envolvida com o reabastecimento.
 - Ex. 1: Pedidos com valor acima de R\$ 5.000 são encaminhados para Maria;
 - Ex. 2: 70% dos pedidos com valor acima de R\$ 4.000 são entregues em até 3 dias;

Na Figura 3.4 abaixo, podem ser visualizados resultados de mineração de

processos sob a perspectiva organizacional e de processos.

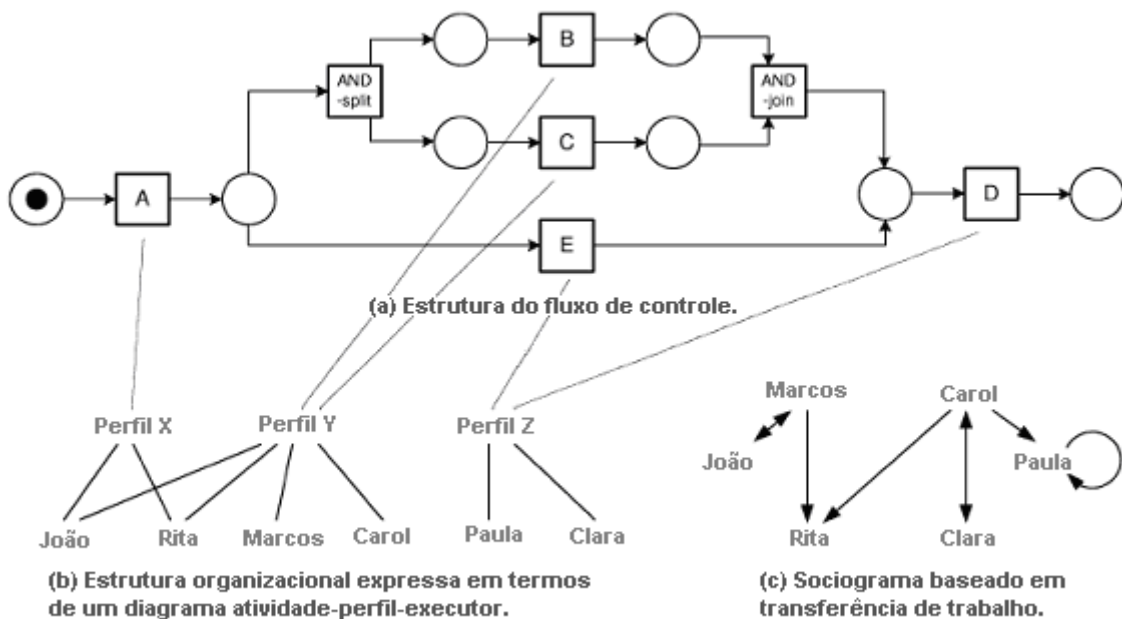


Figura 3.4 - Alguns resultados de mineração baseados nas perspectivas de processos (a) e organizacional (b e c) (AALST & WEIJTERS, 2005).

3.2.7 Framework ProM e o Mining Xml (Mxml)

Ferramentas acadêmicas e comerciais ligadas a BPM (*Business Process Management*) estão sendo desenvolvidas com o objetivo de extrair conhecimento de logs de eventos (por exemplo, logs de transações em ERP's), em outras palavras, para executar mineração de processos (DONGEN *et al.*, 2005). Entretanto estas ferramentas possuem diferentes formatos para leitura e gravação de logs e resultados, o que gera um problema de compartilhamento de informação entre diferentes ferramentas. Em consequência disto, pesquisadores que trabalham sobre novas técnicas de mineração de processos são forçados a construir uma infra-estrutura para testar e executar suas técnicas de forma isolada, ou seja, independente de algum software específico. Neste contexto foi proposto o ProM (PROM, 2008), *Process Mining Framework*, um framework extensível, de código aberto, que provê suporte a uma ampla variedade de técnicas de mineração de processos sob a forma de plug-ins, e está sendo utilizado em muitas abordagens de mineração de processos na literatura (AALST & GÜNTHER, 2007, AALST *et al.*, 2005, MEDEIROS *et al.*, 2007, AALST *et al.*, 2003).

No contexto desta pesquisa, o ProM se apresenta como uma importante ferramenta para auxiliar na identificação das regras de negócios, devido a funcionalidades fornecidas que incluem, por exemplo, a obtenção dos logs de eventos em diferentes formatos (ProM Import) (PROM, 2006), a leitura e facilidade no processamento de registros no padrão Mxml (*Mining Xml*), a disponibilidade de algoritmos de mineração disponíveis no framework sob a forma de plug-ins e a exibição dos resultados da mineração sob diversos formatos. O ProM framework é um software de código aberto que é utilizado em diversos trabalhos na área de mineração de processos descritos na literatura (AALST *et al.*, 2005, ALBERTINI *et al.*, 2007, MEDEIROS *et al.*, 2007).

O ProM framework utiliza o Mxml (*Mining XML*) como formato padrão de entrada para os logs de eventos que são utilizados na execução de diversos algoritmos de mineração de processos. O formato Mxml é uma iniciativa para padronizar o formato de entrada em ferramentas de mineração (AALST *et al.*, 2003), tornando possível que os mesmos logs de eventos sejam compartilhados entre diferentes ferramentas. Na Figura 3.5 pode ser observada uma representação visual do esquema do Mxml (MXML, 2009), com destaque para as principais informações para a execução da mineração de processos.

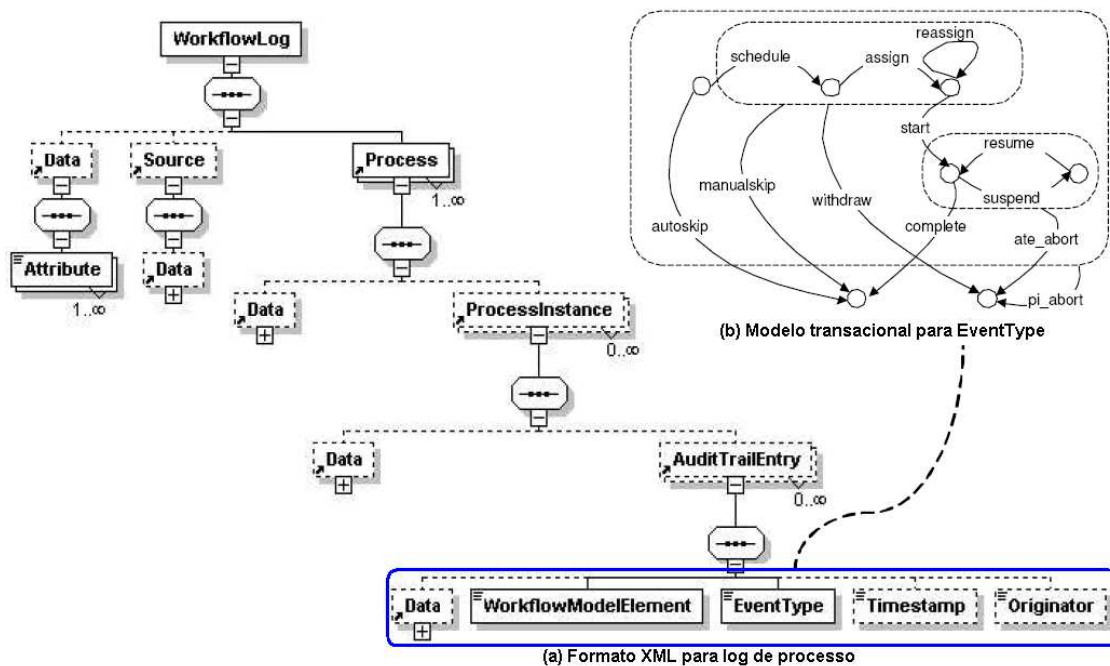


Figura 3.5 - Representação visual do padrão Mxml (MEDEIROS & GÜNTHER, 2005).

Através da Figura 3.5 pode ser observado que nos registros de execuções de um processo (“*Process*”), existem diversas instâncias do processo (“*ProcessInstance*”) e em cada uma dessas instâncias existem os dados de execução de cada evento (“*AuditTrailEntry*”). Em cada evento, são registradas as seguintes importantes informações:

- “*WorkflowModelElement*”: Representa o nome da atividade;
- “*EventType*”: Identifica o passo da atividade no processo. Possui um conjunto de valores permitidos que pode ser visualizado no item b da Figura 3.5;
- “*Timestamp*”: Registra o momento em que o evento é executado;
- “*Originator*”: Identifica quem disparou o evento;
- “*Data*”: Serve para adicionar qualquer outra informação extra ao evento, e é composta por atributos que são nomeados e podem receber valores sem formalismos ou restrições. Em geral, representam campos não obrigatórios para que seja possível efetuar mineração de processos, mas podem conter informações importantes para algumas técnicas de mineração. No contexto deste trabalho este campo é fundamental por fornecer informações contextuais sobre a execução de cada atividade que são utilizadas para a mineração de regras de negócio.

Conforme foi apresentado, a ferramenta de mineração ProM possui como formato de entrada o Mxml, mas diversos sistemas existentes já registram seus eventos em formatos específicos, gerando um gap entre os eventos registrados e o formato Mxml (GÜNTHER & AALST, 2006). Para resolver este problema foi concebido o ProM Import Framework (PROM, 2006), que provê a conversão de diversos formatos de logs de eventos para o formato Mxml e, além disso, ele pode ser estendido para adição de plug-ins para tipos de logs específicos.

No capítulo a seguir será apresentado o método para descoberta de regras de negócio através de mineração e mais adiante o ProM será utilizado como parte importante para a aplicação deste método.

4 Método para Descoberta de Regras de Negócio Através de Mineração

Esse capítulo apresenta a arquitetura e o método propostos para a descoberta de regras de negócio com base na aplicação de conceitos e técnicas de mineração de processos e mineração de dados. A Seção 4.1 apresenta como as regras de negócio influenciam o comportamento dos sistemas de informação, a Seção 4.2 apresenta a arquitetura e a Seção 4.3 apresenta o método para descoberta de regras de negócio através de mineração.

4.1 Regras de Negócio e os Sistemas de Informação

Diante da ausência de documentação sobre as regras de negócio nas organizações e da dificuldade para a execução de um levantamento destas regras, conforme apresentado no Capítulo 2, o desenvolvimento de mecanismos automatizados para descoberta das regras de negócio das organizações vem sendo pesquisado com o objetivo de expor as regras de negócios (PUTRYCZ & KARK, 2007, CRERIE *et al.*, 2008, POO, 1999, LEITE & CERQUEIRA, 1995, SULAIMAN, 2002, WANG *et al.*, 2008, CRERIE *et al.*, 2009, SIQUEIRA, 2002).

Muitos sistemas de informação da atualidade armazenam eventos relevantes de forma estruturada como, por exemplo, os sistemas de gerenciamento de workflow, que registram o início e o término de cada atividade, e os ERP's (*Enterprise Resource Planning* ou Sistemas Integrados de Gestão Empresarial) que registram todas as transações efetuadas (AALST & WEIJTERS, 2005, AALST & GÜNTHER, 2007, GOEDERTIER & VANTHIENEN, 2005). Além disso, na atualidade, os sistemas de informação transacionais, que apóiam as atividades operacionais das organizações, comumente registram algum tipo de histórico com informações referentes a cada evento executado. SULAIMAN (2002) afirma que, os logs de auditoria são armazenados através da inclusão de comandos de registro para rastreamento de determinadas

transações nos sistemas. Além disso, as empresas que possuem capital na bolsa de Nova Iorque precisaram rever e documentar seus logs de auditoria para atender às necessidades da lei *Sarbanes-Oxley* (SOX) (KAARST-BROWN & KELLY, 2005). Esta lei foi criada em resposta a uma série de escândalos contábeis de grandes corporações como Enron e WorldCom, com o objetivo de evitar novos escândalos contábeis no futuro e reconstruir a confiança do público. *Sarbanes-Oxley* aborda os deveres do CEO (*Chief Executive Office*), do CFO (*Chief Financial Officer*), e do auditor, detalhando as obrigações de comunicação, juntamente com as normas e regulamentos para a elaboração dos relatórios financeiros e incluindo impactos sobre sistemas da organização, através de, por exemplo, uma política para retenção de registros para auditoria (KAARST-BROWN & KELLY, 2005), o que indica a necessidade do registro cuidadoso e completo de um log de auditoria para as organizações.

De uma forma geral, os eventos registrados nesse logs de auditoria incluem ações que são realizadas no negócio e que, devido a isso, também sofrem influência das regras de negócio da organização, fazendo com que o conjunto de informações ligado a estes eventos possua características guiadas por estas regras. Segundo Sulaiman (SULAIMAN, 2002) qualquer programa possui, em seu código e em suas estruturas de dados, algum conhecimento sobre o domínio do problema que soluciona, e este conhecimento acaba por se manifestar nas diferentes saídas obtidas para cada entrada específica.



Figura 4.1 - Instâncias de uma atividade onde o valor do atributo “Cliente Antigo?” influencia o valor de “Valor de Desconto” devido à existência de uma regra de negócio.

Por exemplo, suponha um sistema de informação em uma organização, que apóia a atividade “Concluir Pedido de Compra”, da qual algumas instâncias estão ilustradas na Figura 4.1. Esta atividade é guiada pela regra “somente cliente antigo pode

obter desconto” e cada execução sua verifica se o cliente é antigo. Para os casos em que o cliente realmente é antigo, a atribuição de desconto pode ser concedida.

Os procedimentos ligados à regra de negócio citada implicam em mudança nos dados ligados à execução da atividade e, conseqüentemente, nos dados vinculados aos eventos gerados. No exemplo, se o cliente for novo, o valor total do pedido vai ser calculado e registrado levando-se em conta o valor unitário e a quantidade de cada item; se o cliente for antigo, então o valor total do pedido registrado será debitado de um percentual de desconto atribuído no momento da venda. Isso expõe como as regras de negócio, mesmo que não estejam explicitamente documentadas, influenciam o conjunto de informações que estão ligadas aos eventos do negócio e que, em muitos casos, estão implementados nos sistemas de informação que dão suporte ao negócio da organização. Aliado a isto, existe a tendência cada vez maior de direcionar todas as regras para o software (GUILARDUCCI, 2007), porque o software é o elemento mais adaptável de SI e a adaptabilidade a mudanças em regras de negócio é um fator crítico nas empresas modernas. Portanto, o log de eventos dos sistemas de informação torna-se uma importante fonte de informação a partir de onde podem ser extraídas as regras de negócio de forma automatizada, através do tratamento e processamento destes, e podem ser expostas as regras que estão efetivamente implementadas nos sistemas que controlam as atividades realizadas (regras ‘As Is’), de forma que analistas de negócio possam visualizar o que esta acontecendo no cotidiano da organização.

Baseando-se nestas informações, esta seção apresenta uma arquitetura e um método para a descoberta de regras de negócio através da aplicação de técnicas de extração de conhecimento sobre logs de eventos dos sistemas de informação. A arquitetura proposta baseia-se na arquitetura da mineração de processos (AALST & WEIJTERS, 2005) (descrita na Seção 3.2), uma vez que as regras de negócio são parte inerente do processo de negócio. O framework ProM (PROM, 2008) foi estendido para a implementação da arquitetura proposta.

4.2 Arquitetura Proposta

Esta seção visa expor e detalhar uma arquitetura elaborada nessa pesquisa para viabilizar a descoberta de regras de negócio que estão em sistemas da organização através da mineração automatizada dos logs de eventos de Sistemas de Informação. Essa arquitetura, cujos módulos podem ser visualizados na Figura 4.2, representa o

seguinte cenário: em uma organização, diversos sistemas de informação geram logs de eventos que registram dados sobre as operações que são executadas neles. Tais logs serão utilizados como o conjunto de dados sobre os quais serão aplicadas técnicas de mineração, cuidadosamente selecionadas para descoberta de cada tipo de regra de negócio desejado. No entanto, cada sistema de informação pode registrar seu log de eventos de maneiras diferentes, e um exemplo típico é tais dados constarem das bases de dados utilizadas pelo sistema.

A partir deste cenário, existe a necessidade de se uniformizar a representação do log que pode ser realizada de duas formas: os sistemas devem ser alterados para que os logs sejam gerados em determinado formato, o que requer uma imensa carga de trabalho e necessidade de grande investimento por parte da organização; ou, o tratamento do log específico gerado pelo sistema pode ser feito, convertendo-o em um formato padronizado e adequado para a mineração. Esta segunda opção tende a ser menos custosa do ponto de vista do esforço de desenvolvimento, visto que os sistemas em questão podem representar uma cadeia de softwares legados e com alto grau de complexidade em seu código. Além disso, pode ser necessário selecionar um subconjunto dos dados existentes nos logs específicos para compor o log padronizado. O módulo “**Padronizador do log de eventos**” na arquitetura é o responsável por esta seleção e conversão na arquitetura, e recebe como entrada logs de quaisquer sistemas de informação (tornando-se, devido a isso, dependente desses sistemas) e os converte para um formato padronizado.

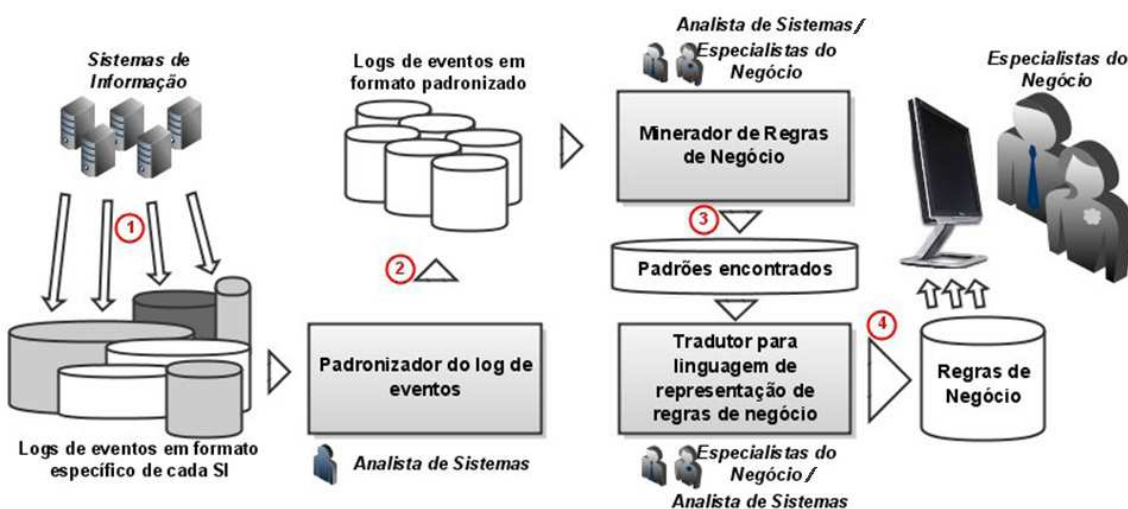


Figura 4.2 – Representação do método de descoberta de regras de negócio. Os retângulos representam os módulos da arquitetura e os círculos numerados representam os passos do método definido nesta pesquisa.

A padronização do log de eventos organiza as informações para que a realização da extração de conhecimento do log seja feita de forma independente de qualquer sistema de informação. O “**Minerador de Regras de Negócio**” é o módulo mais importante da arquitetura, pois é responsável por realizar o processo de descoberta de conhecimento em si. Ele utiliza os logs de eventos convertidos para o formato padrão e executa minerações específicas para descobrir determinado tipo de regra de negócio. As técnicas aplicadas são dependentes dos tipos de regras que serão tratados, pois para cada tipo de regra de negócio podem ser utilizados procedimentos e técnicas diferentes.

O objetivo principal de efetuar a descoberta das regras aplicando-se esta arquitetura é expor aos especialistas do negócio as regras encontradas, configurando-se, desta forma, a necessidade de expor os resultados de forma clara, em um formato que seja familiar aos especialistas. O módulo “Minerador de Regras de Negócio” não trata a forma de exposição dos resultados obtidos, havendo apenas a preocupação em se encontrar os padrões que representam as regras de negócio. Devido a isso, o módulo “**Tradutor para linguagem de representação de regras de negócio**” usa os padrões descobertos como entrada e os converte para um formato de representação de fácil leitura para os especialistas do negócio.

4.3 Método Baseado na Arquitetura Proposta

Com base na arquitetura criada e exposta na Seção 4.2, foi criado nesta pesquisa o método apresentado nesta seção para descoberta e representação de regras de negócio a partir de logs de sistemas de informação. Ele é composto por 4 etapas (Figura 4.2) que são responsáveis pela obtenção da informação nos logs de eventos, pelo tratamento e padronização destas informações, pela descoberta de conhecimento e pela conversão para as regras de negócio que serão apresentadas a um especialista do negócio. Estas 4 etapas são:

1. Registro do log de eventos dos sistemas de informação;
2. Conversão para o formato padronizado;
3. Aplicação de técnicas de mineração para a descoberta de padrões que representem as regras;
4. Tradução das regras para um formato de representação de regras de negócio;

Nas subseções a seguir, será detalhada cada uma destas 4 etapas deste método.

4.3.1 Etapa 1: Registro de Execução e Necessidades dos Logs de Sistemas de Informação

Assim como existem características necessárias para que o log de eventos possa ser utilizado para realizar a mineração de processos, expostas na Seção 3.2.5, o método para descoberta de regras de negócio também estabelece premissas específicas para que se torne possível. Pode ser necessário selecionar certas informações no log de eventos para a realização da descoberta de um determinado tipo de regra de negócio. Apesar da existência de necessidades específicas por tipo de regra de negócio, a principal característica do log associada com a descoberta de regras de negócios é ser independente do tipo de regra que se deseja descobrir: cada evento registrado no log deve conter o máximo de informações ligadas à execução da atividade que o gerou, informações estas denominadas de “**atributos de contexto**”. Nestes atributos devem ser registradas as informações de estado para as variáveis relacionadas com a execução de uma determinada atividade. Por exemplo, abaixo estão algumas atividades e um possível conjunto de atributos de contexto para elas:

- Emprestar Livro: nome do livro, data, locador, atendente;
- Registrar Venda: itens, valor, desconto, vendedor, cliente;
- Enviar E-mail: para, cópia, cópia oculta, assunto, mensagem;

Os atributos de contexto são importantes porque possibilitam a descoberta de padrões indicando que uma determinada regra de uma atividade influencia os valores de um subconjunto de atributos de contexto da mesma atividade. Por exemplo, no conjunto de eventos da atividade “Aprovar Crédito”, que contém os atributos de contexto: “Crédito_foi_aprovado”, “Cliente”, “Valor” e “Documentos_apresentados”, é possível identificar, através de uma análise do log gerado que, por exemplo, “Crédito_foi_aprovado” só recebe o valor “Sim” quando o atributo “valor” é menor do que “2000” e “documentos_apresentados” possui os valores “cpf, identidade e comprovante de renda”, conforme pode visto na Figura 4.3 abaixo, onde o conjunto de atributos de contexto respeita essa regra:

Atividade: Aprovar Crédito				
Evento	Crédito foi aprovado	Cliente	valor	documentos apresentados
1	Não	Daniele	1000	cpf
2	Não	Júlio	2000	cpf, identidade e comprovante de renda
3	Sim	Rita	1900	cpf, identidade e comprovante de renda
4	Não	Maria	2000	cpf
5	Não	Ricardo	2500	identidade
6	Não	Raphael	3000	cpf, identidade e comprovante de renda
7	Sim	Mário	500	cpf, identidade e comprovante de renda
8	Não	Jonas	1800	identidade e comprovante de renda
9	Sim	Milena	200	cpf, identidade e comprovante de renda
10	Não	Telma	1000	

Figura 4.3 - Exemplo de informações de contexto em um subconjunto de log de eventos.

4.3.2 Etapa 2: Conversão para Log de Eventos em Formato Padronizado

Enquanto a primeira etapa do método visa obter as informações de diversas fontes e sistemas de informação, a segunda etapa tem como objetivo resolver o problema causado pela possível heterogeneidade das fontes de informação, uma vez que cada SI da organização pode registrar dados históricos de formas distintas.

Diante desse problema de heterogeneidade dos logs de eventos, seria importante executar o método proposto sem que haja necessidade de alterar cada um dos sistemas que o geram. A abordagem empregada nesta etapa é a conversão do log de eventos gerado em um formato específico de determinado sistema para um formato genérico, que padronize a representação dos logs de eventos. Com base nesse formato padronizado, é eliminada a necessidade de tratar formatos específicos de log de eventos na atividade seguinte, onde o conhecimento sobre o negócio é minerado.

Esse log genérico precisa representar a estrutura básica de um log de eventos, contendo as seguintes informações:

- Nome da atividade: deve possuir a identificação da ação de cujo evento faz parte;
- Momento da execução: deve possuir o registro do momento em que o evento foi realizado, ou seja, deve registrar data e hora.
- Atributos de contexto: os atributos de contexto de cada atividade devem ser armazenados a cada evento com os valores instanciados no momento da execução do evento.

4.3.3 Etapa 3: Descoberta de Padrões e Regras

A terceira etapa do método tem como objetivo a realização da mineração propriamente dita para a descoberta dos padrões que representam as regras de negócio. Ela é independente no que se refere ao sistema de informação que gerou o log de eventos, haja vista que a etapa 2 do método trata especificamente essa questão, fornecendo um log de eventos padronizado.

Nesta etapa o especialista do negócio deve executar a aplicação que realiza o procedimento de descoberta do tipo de regra desejado, indicando os valores para os possíveis parâmetros requeridos pelo tipo de regra de negócio. Para que isso seja possível, um analista de sistema deve ter previamente realizado um trabalho de análise sobre o tipo de regra de negócio que se deseja minerar e desenvolvido mecanismos automatizados para tratar o tipo de regra desejado. Abaixo são expostos os procedimentos definidos para a descoberta de regras do tipo sentenças de ação de autorização, sentenças de ação de condição e sentenças estruturais.

4.3.3.1 Descoberta de Sentenças de Ação de Autorização

As sentenças de ação de autorização se referem a sentenças que indicam que pessoas que possuem determinado perfil na organização podem realizar determinada atividade. Para realizar a descoberta deste tipo de regra com base no conteúdo do log de eventos, é necessário obter o atributo do log que representa o perfil da pessoa na organização, o qual pode ser informado pelo analista de negócio. Portanto, a descoberta de regras do tipo sentenças de ação de autorização requer que as seguintes informações estejam presentes no log de eventos (Figura 4.4):

- “Nome da atividade”;
- “Papel” ou “função” do executor da atividade dentro da organização;

O registro do “nome da atividade” no log de eventos é uma das premissas base da mineração de processos (conforme exposto na Seção 3.2.5) e a “função” é um atributo de contexto requerido especificamente para a descoberta de sentenças de ação de autorização, tornando possível indicar que grupos de pessoas dentro da organização efetivamente executam as atividades existentes no negócio da organização. É importante destacar que essa informação não se refere necessariamente ao papel (role) implementado no SGBD acessado pelo SI, ou ao “grupo de acesso” das pessoas dentro dos sistemas de informação, mas sim à função de trabalho que o indivíduo desempenha na organização.

```

</AuditTrailEntry>
<AuditTrailEntry>
  <Data>
    <Attribute name="Group">
      gerente
    </Attribute>
  </Data>
  <WorkflowModelElement>
    Arquivar reclamacao
  </WorkflowModelElement>
  <EventType>complete</EventType>
  <Originator>John</Originator>
</AuditTrailEntry>
</ProcessInstance>

```

Figura 4.4 - Log de Eventos contendo papel do usuário e nome da atividade em destaque.

4.3.3.2 Descoberta de Sentenças de Ação de Condição

Para que seja possível descobrir regras do tipo sentenças de ação de condição através da aplicação do método proposto, assume-se a existência de um atributo de contexto que indique a forma como o evento foi concluído, em outras palavras, o resultado do evento (*status* do evento), pois esse atributo de contexto servirá para descobrir os fatores que fazem com que determinada atividade seja executada com sucesso ou falha. Na Figura 4.5 segue um exemplo de informações de contexto para a atividade aprovar crédito, onde o atributo “SITUAÇÃO CRÉDITO” indica o status do evento.

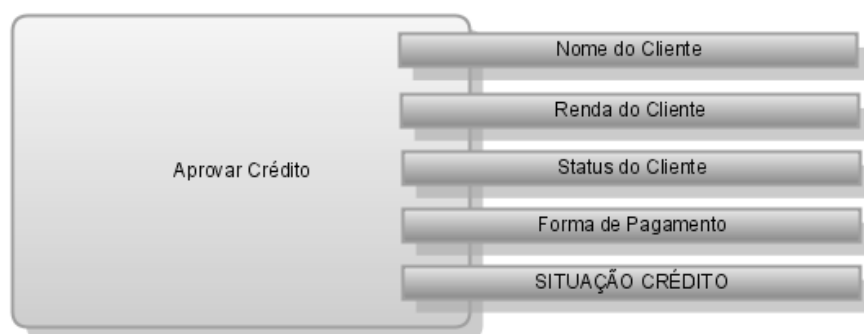


Figura 4.5 - Exemplo de atributos de contexto de atividade.

Através da análise dos valores contidos em cada atributo em um conjunto de execuções do evento, é possível identificar o conjunto de valores possíveis para cada

atributo de contexto (Figura 4.6), de forma similar ao modo como as sentenças de ação de autorização são descobertas (Seção 5.2).

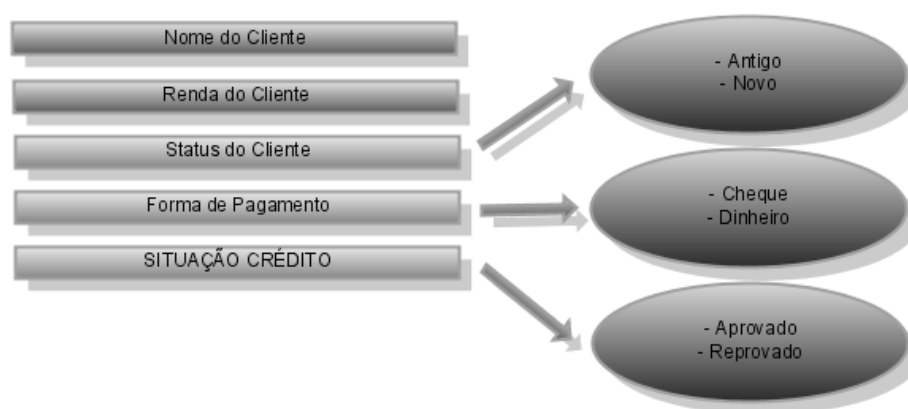


Figura 4.6 - Conjunto de valores possíveis de atributos de contexto.

Cada conjunto “atributo de contexto” + “valor do atributo” é um potencial candidato para representar o conseqüente (o trecho “então...”) de uma regra no formato “se ... então ...”. O trecho “se...” da regra é buscado pelo algoritmo de mineração dentre as possíveis combinações dos demais atributos relacionados com seu contexto, quando se faz uma análise da execução de diversas instâncias da atividade.

Com a análise destas informações pode-se chegar, por exemplo, à seguinte conclusão:

- “SITUAÇÃO CRÉDITO” é aprovada quando:
 - “Forma de Pagamento é Dinheiro” ou;
 - “Forma de Pagamento é Cheque” e “Status do Cliente é Cliente Antigo”.

Neste exemplo pode ser visto que as afirmações “Forma de Pagamento é Dinheiro”, “Forma de Pagamento é Cheque” e “Status do Cliente é Cliente Antigo” são condições para que a afirmação “SITUAÇÃO CRÉDITO é aprovada” seja verdadeira, ou seja, existe uma limitação do contexto de existência de um determinado conceito ou regra de negócio de acordo com outros conceitos e regras existentes no negócio, e esta é a definição de sentenças de ação de condição (Seção 2.3.2), o que demonstra que através da análise dos atributos de contexto das execuções de uma atividade é possível descobrir regras de negócio deste tipo.

4.3.3.3 Descoberta de Sentenças Estruturais

As sentenças estruturais se dividem em termos e fatos, e os fatos podem ser dos seguintes tipos: Atributo, Generalização e Participação. O objetivo da aplicação do método, no contexto deste tipo de regra, é a identificação de relações que exponham, de forma clara a um especialista do negócio, as relações existentes entre conceitos (no caso, atributos de contexto), de forma que este profissional possa ter uma base importante para o mapeamento deste tipo de regra em sua organização. Mas, de forma geral, as características expostas nessa seção forçam a identificação de fatos do tipo atributo e termos que compõem os fatos. Por exemplo, no log de registro de vendas são registrados os seguintes atributos de contexto:

- Nome do vendedor;
- Matrícula do vendedor;
- Nome do cliente;
- CPF;
- Data de nascimento do cliente;
- Valor da compra.

Com base nestes atributos, deve ser possível identificar que em todas as vezes que uma determinada instância de “Nome do vendedor” aparece no log, o mesmo valor para a instância “Matrícula do vendedor” é registrado, e este é um comportamento padrão para os registros destes atributos nos eventos registrados. Desta forma se abstrai que estas duas informações fazem parte de algum conceito do negócio, ou seja, são atributos de um conceito do negócio, e ambas também são termos. Elas podem ser expressas da seguinte forma:

- “Conceito X” tem atributo “Nome do vendedor”;
- “Conceito X” tem atributo “Matrícula do vendedor”;

E da mesma forma, outros atributos podem se relacionar com outro conceito:

- “Conceito Y” tem atributo “Nome do cliente”;
- “Conceito Y” tem atributo “CPF”;
- “Conceito Y” tem atributo “Data de nascimento do cliente”;

O especialista do negócio deve avaliar todas as regras encontradas e deve identificar, de acordo com os atributos de contexto relacionados, quais são os conceitos que foram descobertos. Analisando os dados destas 2 listas acima, ele deve identificar que “Conceito X” é o vendedor e que “Conceito Y” é um cliente.

Como esse tipo de regra visa buscar relações entre os elementos da organização, a única necessidade referente à execução do primeiro passo do método é o registro do máximo de informações de contexto de cada atividade, com o objetivo de obter mais informações e obter mais relações entre atributos de contexto dentro de determinada atividade.

4.3.4 Etapa 4: Tradução para Linguagem de Representação de Regras de Negócio

Os padrões (que representam regras de negócio) descobertos na etapa 3 são resultado da aplicação de procedimentos e técnicas desenvolvidos individualmente para cada tipo de regra de negócio. Devido a isso, o formato do resultado da mineração também varia dependendo dos procedimentos e técnicas empregados e, conseqüentemente, essa diferença precisa ser tratada. Além disso, a execução da mineração pode gerar dados que não sejam de fácil entendimento pelos especialistas do negócio, de modo que é importante que os padrões obtidos na mineração do log sejam convertidos ou traduzidos para uma linguagem que possa ser facilmente interpretada pelos especialistas do negócio e que possua recursos para representar os diferentes tipos de regras de negócio.

Nesta quarta e última etapa do método, as regras de negócio descobertas devem ser convertidas para um formato de representação de regras de negócio, de forma que o especialista de negócio possa entendê-la.

Os especialistas de negócio devem validar as regras encontradas e traduzidas. Caso o resultado da descoberta não seja satisfatório, o especialista, dependendo dos procedimentos definidos na etapa de mineração das regras, pode alterar parâmetros para a execução da mineração, re-executando a etapa 3 do método para verificar se o resultado final da descoberta é modificado.

Esta pesquisa, conforme já discutido, apresenta a aplicação do método para 3 tipos de regras de negócio (sentenças de ação de autorização, de condição e derivações), e, posteriormente, o método pode ser aplicado para os demais tipos de regras. No próximo capítulo, será exposta a forma como estes três tipos de regra foram tratados, com os detalhes de tecnologia empregados.

5 Implementação do Método

A partir das definições apresentadas no Capítulo 4, foi realizada implementação do método com base no framework ProM (PROM, 2008) e nos conceitos de mineração de processos. O objetivo deste capítulo é expor os detalhes desta implementação, explicando etapa a etapa como o método pode ser aplicado e os recursos que foram desenvolvidos e utilizados.

5.1 A Implementação do Método

Para o método proposto, foram definidos algoritmos e ferramentas utilizadas para colocá-lo em prática (Figura 5.1). Foram empregados conceitos de mineração de processos sobre o método, de forma que a execução dos algoritmos de mineração ficou sob a responsabilidade do framework ProM (PROM, 2008). Por consequência disto, o padrão de representação do log de eventos adotado foi o Mxml (*Mining Xml*) (MXML, 2009) e conversão do log de eventos foi executada com o apoio da ferramenta Prom Import (PROM, 2006). A representação das regras de negócio foi feita com base na especificação do metamodelo SBVR (*Semantics of Business Vocabulary and Business Rules*) (OMG, 2008).

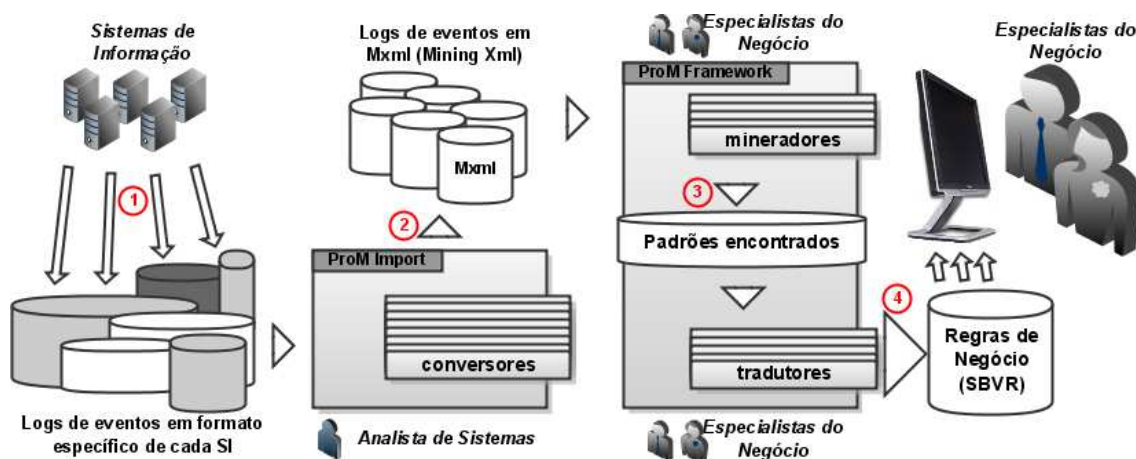


Figura 5.1 - O método para descoberta de regras de negócio com os conceitos e ferramentas utilizados na implementação.

Como a etapa 1 do método não necessita de implementação, nas seções seguintes serão apresentados detalhes das etapas do método, exceto a etapa 1.

5.1.1 Implementação da Etapa 2

Para realizar a carga de informações no ProM é necessário que o log de eventos do sistema de informação, do qual se pretende extrair as regras, esteja em um formato suportado pelo framework e, caso não esteja, existe a possibilidade de utilizar a ferramenta ProM Import (PROM, 2006) que possui recursos para a realização da importação. No ProM Import, além de haver a possibilidade de conversão de alguns formatos específicos de logs (como, por exemplo, logs de ferramentas de controle de versão de software) para o formato Mxml (formato padrão de entrada para ferramentas de mineração de processos, conforme apresentado na Seção 3.2.7), existe uma extensão para a realização de importação de dados armazenados em um banco de dados relacional (EindhovenTU, 2009) cuja estrutura deve ser a estrutura exposta no modelo da Figura 5.2.

Esta possibilidade de conversão para Mxml através da importação de dados é particularmente importante no contexto do método proposto nessa pesquisa porque economiza o esforço para a criação um plug-in no ProM ou no ProM Import para ler um possível novo formato de log de eventos, e, além disso, esse recurso pode ser utilizado para acrescentar novos atributos de contexto que não estejam presentes no log de eventos gerado no passado pela aplicação, mas que estejam disponíveis na base de

dados, como, por exemplo, buscar o sexo do executor de cada evento em uma tabela que armazene as informações pessoais de cada usuário do sistema.

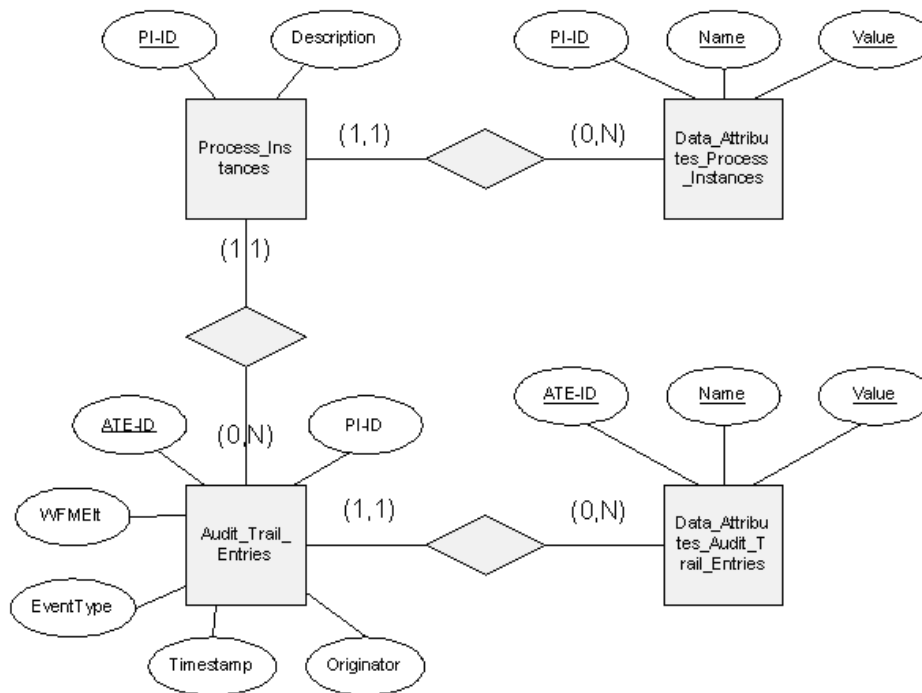


Figura 5.2 - DER para a conversão para Mxml (EindhovenTU, 2009, MALLENS, 1997).

5.1.2 Implementação dos recursos necessários para a Etapa 3 e 4

Para a realização da 3ª etapa, é necessário que um analista de sistemas realize uma análise sobre as características do tipo de regra de negócio a ser descoberta, e, com base nestas características, desenvolva um conjunto de procedimentos e selecione as técnicas de mineração de dados, que são adequadas ao problema, para que se torne possível extrair as regras. Este conjunto de procedimentos identificados para cada tipo de regra deve ser disponibilizado para que os especialistas do negócio possam executar esta terceira etapa do método sempre que necessitarem, extraindo as regras que eles desejam. O ProM framework foi adotado para a realização desta etapa. Nesta pesquisa foram desenvolvidos plug-ins para 2 tipos de regras de negócio com base nas seguintes atividades:

- Análise de pré-processamentos necessários sobre o log de eventos;

- Seleção das técnicas e procedimentos de mineração de dados que devem ser aplicados, com base nas características do tipo de regra e das informações presentes no log;
- Definição e implementação do plug-in para a realização da mineração como um todo. Essa atividade é essencial para que os especialistas do negócio consigam realizar minerações de forma independente. O desenvolvimento destas atividades em um aplicativo é extremamente recomendado para que o analista de negócio utilize seu tempo apenas na análise dos resultados e para que seja reduzida a possibilidade de ocorrer alguma falha humana no momento da aplicação desta importante etapa do método.

Por fim, os padrões encontrados nos procedimentos de mineração devem ser traduzidos, na etapa 4, para um único formato de representação, com o objetivo de facilitar a leitura por parte dos analistas de negócio. Para isso foi selecionada uma representação baseada no padrão SBVR (OMG, 2008), devido ao fato de ter se tornado um padrão OMG (OMG, 2009) e por possuir características que facilitam tanto o entendimento por parte dos analistas de negócio quanto o processamento automatizado das regras. Essa representação foi baseada nas estruturas apresentadas na Seção 2.4.3.

Através do ProM os especialistas do negócio podem fazer a carga do Mxml com os dados das execuções dos sistemas e chamar o plug-in que realiza a descoberta de determinado tipo de regra (Figura 5.3). Antes da execução do algoritmo é possível que o especialista possa selecionar alguns parâmetros ligados à execução do algoritmo, dependendo de cada implementação.

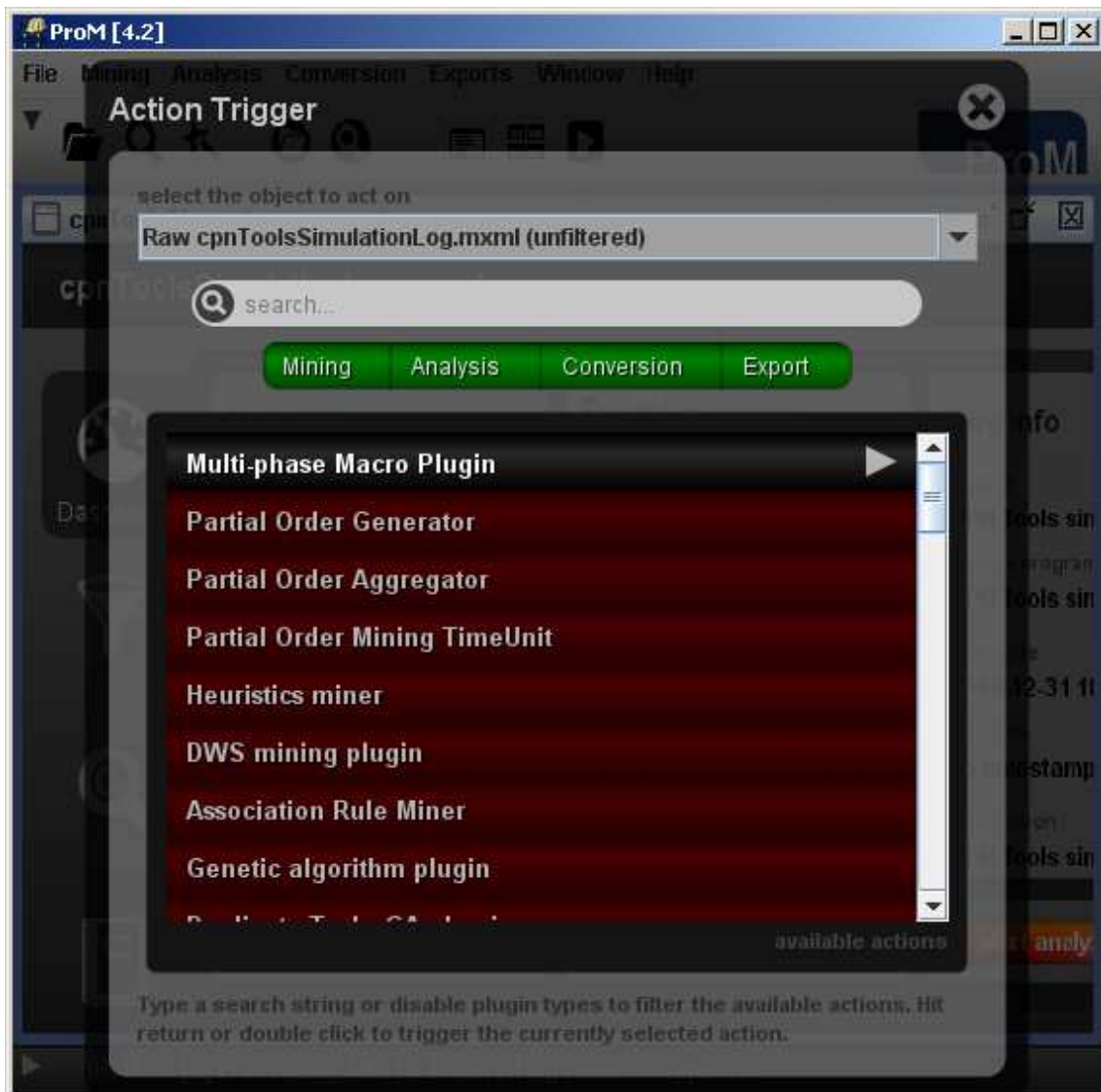


Figura 5.3 - Detalhe da tela do ProM: Tela de seleção do algoritmo (Plug-in).

Nas subseções seguintes serão apresentados os procedimentos da aplicação deste método para 3 tipos de regras de negócio: sentenças de ação de autorização, sentenças de ação de condição e sentenças estruturais. Serão discutidos exemplos e a aplicação deste passo será detalhada.

5.1.2.1 Aplicação do Método para Descoberta de Sentenças de Ação de Autorização

Sentenças de ação de autorização, como exposto na Seção 2.3.2, são regras que restringem quem pode executar determinada ação. A seguir será apresentado como as etapas do método desenvolvido nessa pesquisa foram aplicadas para este tipo de regra e também será exposto um exemplo da execução do método.

Para a realização da etapa 3 do método proposto, foi criado um plug-in para o framework ProM (PROM, 2008) que carrega os dados e analisa as informações

presentes em cada execução. Como as sentenças de ação de autorização são regras que indicam as funções das pessoas na organização que podem realizar determinada atividade e o log de eventos foi organizado de forma que em cada evento gerado contenha diretamente a informação da função ou perfil de quem realizou a atividade, as informações necessárias estão expostas no log de eventos de forma direta, não havendo necessidade da utilização de algum algoritmo de mineração de dados ou de processos. Diante disso, o plug-in desenvolvido realiza uma busca sequencial nas instâncias de execução e identifica, para cada atividade presente no log, quais perfis de usuários (representadas pelo atributo “Group” na Figura 4.4) a executaram.

Para a realização da última etapa do método, a tradução para linguagem de representação de regras (baseado em SBVR) para a representação das sentenças de ação de autorização, foi definida a seguinte estrutura:

- “indicação do tipo de permissão”: caso somente um grupo possa realizar a atividade, de acordo com a análise do log de eventos, é usada a representação “It is obligatory that...” para indicar que somente um grupo de usuários pode realizar a atividade; se, no entanto, mais de um grupo tenha executado a atividade, a representação é feita por “It is permitted that...”;
- “nome da atividade”: indica o nome da atividade em análise;
- “indicação de condição para ação”: é o trecho que faz a ligação entre a atividade e o grupo que a executa, limitando a execução da atividade ao grupo. É representada por “... be run only if...”;
- “nome que indica o grupo no log de eventos”: palavra que representa o conceito grupo no log;
- “nome do grupo”: nome que indica uma instância de grupo no log;

Com a aplicação desta estrutura sobre as regras encontradas, são descritas regras como as expostas abaixo:

*It is permitted that AVALIAR RECLAMACAO be run only
if GRUPO is GERENTE;*

*It is permitted that AVALIAR RECLAMACAO be run only
if GRUPO is CORDENADOR;*

It is obligatory that ENVIAR AVISO DE APROVACAO be run only

if GRUPO is GERENTE;

5.1.2.2 Exemplo de Sentenças de Ação de Autorização Descobertas

Para exemplificar um cenário de uso do método para a descoberta de regras de negócio do tipo sentenças de ação de autorização, foi gerado um Mxml com base no processo fictício exposto na Figura 5.4. As instâncias contidas no Mxml possuíam as seguintes características:

- 3 perfis de usuários foram definidos (Role0, Role1 e Role2), para representar o papel do executor dentro da organização;
- O perfil associado à execução de cada instância da atividade “FileFine” foi selecionado randomicamente dentre os perfis existentes;
- Os perfis das demais atividades do processo foram definidos da seguinte forma:
 - Todas as instâncias de “SendBill” foram executadas por usuários “Role0”;
 - Todas as instâncias de “ProcessPayment” foram executadas por usuários “Role0”;
 - Todas as instâncias de “CloseCase” foram executadas por usuários “Role0”;
 - Todas as instâncias de “SendReminder” foram executadas por usuários “Role2”;

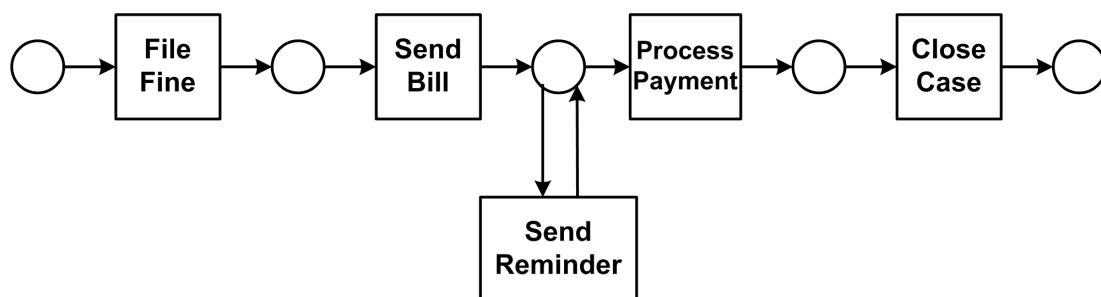


Figura 5.4 - Parte do processo base utilizado como exemplo (MEDEIROS & GÜNTHER, 2005).

O resultado da execução do plug-in trouxe como resultado a identificação das seguintes regras:

*It is obligatory that SendReminder be run only
if GRUPO is role2;*

*It is obligatory that CloseCase be run only
if GRUPO is role0;*

*It is obligatory that ProcessPayment be run only
if GRUPO is role0;*

*It is obligatory that SendBill be run only
if GRUPO is role0;*

*It is permitted that FileFine be run only
if GRUPO is role0;*

*It is permitted that FileFine be run only
if GRUPO is role1;*

*It is permitted that FileFine be run only
if GRUPO is role2;*

O Capítulo 6 apresentará uma avaliação da aplicação do método e discutirá com mais detalhes os resultados encontrados.

5.1.2.3 Aplicação do Método para Descoberta de Sentenças de Ação de Condição

Sentenças de ação condicionais, como exposto na Seção 2.3.2, restringem a existência de determinada regra de negócio de acordo com a aplicação de outras regras de negócio, em outras palavras, uma determinada regra vai ser válida para o negócio quando outras regras forem aplicadas. Nesta seção está exposta a forma como foi realizada a aplicação do método desenvolvido para a descoberta desse tipo de regra nos logs de eventos.

Conforme exposto na Seção 3.1.1, classificação é o processo de busca por um conjunto de modelos (funções) que descrevem e distinguem classes ou conceitos, o que se enquadra no objetivo de descobrir modelos que descrevam através de limitações a existência de determinada regra de negócio. Além disso, o processo de classificação é composto por 2 conjuntos de dados (base de treinamento e base de testes) que podem ser representados como subconjuntos do conjunto de instâncias de execução de cada atividade que será analisada, e os atributos utilizados como classificadores são os atributos de contexto presentes nas atividades. Desta forma, a classificação foi utilizada nesta pesquisa como técnica de mineração de dados para a descoberta de sentenças de ação de condição.

Para a realização da classificação no plug-in desenvolvido no ProM, foi utilizado o algoritmo J48, implementação em Java baseada no tradicional algoritmo C45 presente na biblioteca WEKA (2009) de mineração de dados. No plug-in criado, o Mxml contendo o log de eventos é processado e é feito um agrupamento dos eventos por atividade, ou seja, são reunidos todos os eventos da mesma atividade. Para cada atividade são realizados então os seguintes passos no plug-in:

- É obtida a lista de todos os valores possíveis para cada atributo de contexto da atividade, dado este necessário para a mineração no J48;
- O Algoritmo J48 é executado utilizando cada um dos atributos de contexto formados por texto como atributos classificadores (class label attribute), os valores numéricos não podem ser utilizados por restrição do algoritmo de classificação, vide seção 3.1.1;
- Os padrões encontrados são convertidos em regras conforme exposto abaixo.

A aplicação da etapa 4 do método proposto nesse trabalho sobre o resultado da mineração no plug-in, ou seja, a conversão para SBVR, foi baseada na construção das restrições sobre cada caminho existente na árvore de decisão resultante da mineração de dados. Por exemplo, para a árvore da atividade CREDIT_STATUS (Figura 5.5), são geradas as seguintes regras:

```
R1: if payment_type is money
    then CREDIT_STATUS is APPROVED;
R2: if payment_type is credit and client_type is New_Client
    then CREDIT_STATUS is DISAPPROVED;
R3: if payment_type is credit and client_type is Old_Client
    then CREDIT_STATUS is APPROVED;
```

```
=== Classifier model (full training set) ===

J48 pruned tree
-----

payment_type = money: APPROVED (1028.0)
payment_type = credit
|  client_type = New_Client: DISAPPROVED (500.0)
|  client_type = Old_Client: APPROVED (472.0)
```

Figura 5.5 - Árvore de decisão obtida através do algoritmo J48.

Observe que há a possibilidade, não tratada no presente trabalho, de combinar as regras R1 e R3, gerando a regra abaixo:

*R1: if payment_type is money or (payment_type is credit and client_type is Old_Client)
then CREDIT_STATUS is APPROVED;*

5.1.2.4 Exemplo de Sentenças de Ação de Condição Descobertas

Para executar um exemplo de aplicação do método para sentenças de ação de condição, foi utilizado como base um modelo utilizado em pesquisas na área de mineração de processos (MEDEIROS & GÜNTHER, 2005) (Figura 5.4).

Com base neste modelo foi gerado um Mxml contendo instâncias que tiveram os valores dos atributos de contexto guiados por regras de controle sobre os dados de cada execução. Dentre essas regras estão as seguintes:

As atividades “SendBill” e “CloseCase” tiveram os seguintes atributos de contextos registrados:

- Role: indica se o perfil do executor da atividade na organização é “role0”, “role1” ou “role2” (a atividade “SendBill” é realizada sempre por pessoas do perfil “role0”);
- Client_type: indica se o cliente é “Old_Client” ou “New_Client”;
- Payment_Type: registra o tipo de pagamento utilizado (“credit” ou “money”);
- Amount: registra o valor total da transação. Os valores possíveis são: "130", "210", "200", "145", "1000", "1250", "515", "3100", "350", "45", "10000", "125", "3000", "180", "1450", "5050", "900", "1430", "1700", "2000" ou "5000";
- Credit_Status (somente na atividade “SendBill”): registra se o status da operação foi “APPROVED” ou “DISAPPROVED”.
- Payment Status (somente na atividade “CloseCase”): registra se o status da operação de pagamento foi “OK”, “Denied” ou “Failure”;
- Delivery_released (somente na atividade “CloseCase”): registra se a entrega do produto foi efetuada, através dos valores “YES” e “NO”;

No Apêndice A podem ser visualizados exemplos de funções que foram criadas na ferramenta *CPN Tools* (CPNTOOLS, 2007) para forçar o comportamento de alguns dos atributos de contexto citados acima, de acordo com os valores dos outros atributos de contexto em cada uma das instâncias simuladas do processo. Ao executar isso, torna-se viável avaliar se é possível extrair regras esperadas de um log de eventos.

Com base nisso foi possível identificar algumas regras, como as regras abaixo:

```
if payment_type is money  
    then CREDIT_STATUS is APPROVED ;  
if payment_type is credit and client_type is New_Client  
    then CREDIT_STATUS is DISAPPROVED;  
if payment_type is credit and client_type is Old_Client  
    then CREDIT_STATUS is APPROVED;  
  
if Payment_status is Failure  
    then delivery_released is NO;  
if Payment_status is OK  
    then delivery_released is YES;  
if Payment_status is Denied  
    then delivery_released is NO;
```

Nestas regras descritas acima, é possível observar que a relação entre os atributos “payment_type”, “Client_type” e “CREDIT_STATUS” foi encontrada assim como a ligação entre “delivery_released” e “Payment_status”. No Capítulo 6 serão apresentadas uma avaliação da aplicação do método e uma avaliação sobre os resultados encontrados.

5.1.2.5 Aplicação do Método para Descoberta de Sentenças Estruturais

Sentenças de ação estruturais, como exposto na Seção 2.3.1, são regras que expressam algum aspecto estrutural da organização como um conceito ou um relacionamento entre conceitos. A seguir será apresentado como as etapas do método de descoberta de regras de negócio foram aplicadas para este tipo de regra e também será exposto um exemplo da execução do método.

Para este tipo de regra não chegou a ser desenvolvido um plug-in no ProM framework, mas foi desenvolvida uma sequência de atividades que representam os passos que deverão ser executados em um possível plug-in. Existem 2 parâmetros de configuração para a execução desse plug-in:

- Parâmetro 1: Indica o percentual dos valores das instâncias de determinado atributo que será considerado na análise. Por exemplo, se esse valor é de “15” e existem “100” valores diferentes para o atributo em análise, isso significa que 15% de 100 será o valor dos atributos a ser considerado na análise;
- Parâmetro 2: Indica o número mínimo que deve ser utilizado. Caso o valor selecionado pela aplicação do parâmetro 1 seja menor que o valor definido no parâmetro 2, o valor do segundo parâmetro passa a vigorar para a análise do atributo. Por exemplo, se o parâmetro 1 possui valor “10”, o parâmetro 2 possui valor “8” e existirem 30 valores para o atributo, o valor definido para a análise do atributo é 8, pois 10% de 30 correspondem a apenas 3 valores. A aplicação do parâmetro 2 é condicionada pela quantidade de valores distintos de cada atributo, visto que para os casos onde exista uma quantidade menor de valores distintos que a quantidade imposta por esse parâmetro, todos os valores do atributo são utilizados na análise, pois não é possível obter a quantidade de valores definida no parâmetro.

Os seguintes os seguintes procedimentos foram desenvolvidos nessa pesquisa para realizar a identificação de sentenças estruturais:

- Para cada atributo de contexto:
 - Descobrir o conjunto de valores distintos para o atributo, considerando todos os eventos. Por exemplo, um atributo “STATUS” pode possuir os valores “OK”, “EM PROCESSAMENTO” e “FALHA”;
 - Selecionar aleatoriamente um subconjunto desses valores. Para isso, são utilizados os dois parâmetros definidos para a descoberta de sentenças estruturais, descritos anteriormente, nesta mesma seção;
 - Para cada valor desse subconjunto:

- Selecionar todos os eventos da atividade que contém o valor;
- Executar associação para o conjunto de eventos selecionados;
- Obter o conjunto de atributos relacionados pertencentes ao resultado mais relevante da associação;
- Para cada elemento do conjunto de atributos relacionados:
 - Efetuar novamente o procedimento;
 - Verificar, no resultado da associação, se a relação mais relevante encontrada expõe o atributo originalmente selecionado para análise, o que indica que estes dois atributos são fortemente associados.

Ao final deste procedimento, é verificado se a relação encontrada entre dois atributos é recíproca. Para isso, se o procedimento acima é aplicado ao atributo A e é encontrada relação com o atributo B, quando a análise for realizada para o atributo B deve ser encontrada relação com o atributo A. No contexto desse procedimento, quando isso ocorre, a relação encontrada entre estes dois atributos é denominada “relação forte” e ambos, portanto, são “fortemente associados”.

Para representação em SBVR, segundo a quarta etapa do método exposto nessa pesquisa para a descoberta de regras, deve ser feita uma abstração no que diz respeito às relações encontradas: Se a aplicação desse procedimento descrito acima resultar na descoberta de dois grupos de atributos associados, um deles contendo, por exemplo, “matricula”, “nome” e “cargo” e o segundo contendo “número” e “andar”, a representação deles deve indicar que cada grupo de atributos é ligado a um conceito do negócio, não identificado, mas que se sabe a relação existente entre seus atributos, conforme exemplo abaixo:

“Conceito1” has “matricula”;

“Conceito1” has “nome”;

“Conceito1” has “cargo”;

“Conceito2” has “número”;

“Conceito2” has “andar”.

Esse conjunto de informações deve ser analisado pelos especialistas do negócio com o objetivo de formalizar as regras e identificar o conceito cuja estrutura foi encontrada.

No Capítulo 6 serão apresentadas uma avaliação da aplicação do método e uma avaliação sobre os resultados encontrados.

5.1.2.6 Aplicação do Método Sobre Outros Tipos de Regras de Negócio

Neste trabalho o método desenvolvido foi aplicado para as regras do tipo sentenças de ação de autorização, sentenças de ação de condição e sentenças estruturais. Nesta seção será discutida a possibilidade de aplicação do método para outros tipos de regras de negócio, visando explicar como o método pode ser usado para a descoberta de outros tipos de regras nas organizações, a saber: sentenças de ação de restrição de integridade, derivação por cálculo matemático e inferência.

A descoberta de sentenças de ação do tipo restrição de integridade requer na etapa 1 do método proposto, adicionar como informação de contexto no log de eventos o status da execução da atividade para a instância. Com base nisso um conjunto de minerações deve ser realizado através da aplicação de classificação, utilizando como atributo classificador o atributo de status e para cada execução deve ser selecionado somente um atributo de contexto (uma execução). Após a realização, deve ser verificado qual o conjunto de valores possíveis do atributo para o caso em que o status da atividade indicar sucesso na execução (esse valor deve ser parametrizado para a execução da mineração).

Nas regras de negócio de derivação do tipo “cálculo matemático”, técnicas de tarefas de regressão podem ser aplicadas sobre atributos de contexto numéricos, com o objetivo de obter as fórmulas de formação dos valores. No entanto a regressão linear é eficaz para a descoberta de fórmulas simples relacionando os atributos, para fórmulas mais complexas é necessário buscar outra solução. Regras do tipo derivação por inferência tendem a ser descoberta através de estruturas do SGBD, pois são representadas por relações entre conceitos dependentes e seus atributos.

6 Avaliação da proposta

Esse capítulo tem como objetivo apresentar os resultados da avaliação realizada com base no método criado nesta pesquisa e apresentado no Capítulo 4 para obter regras de negócio com base nos logs de sistemas de informação.

Na Seção 6.1 será apresentado o método para realização da avaliação, e na Seção 6.2 é feito um detalhamento do cenário da avaliação. A Seção 6.3 apresenta o modelo de processo de negócio que foi utilizado nessa avaliação e as Seções 6.4 e 6.5 são dedicadas, respectivamente, à forma de geração do log de eventos e ao resultado da execução da mineração.

6.1 Método de Avaliação

Um fator fundamental para que esse método possa ser aplicado é que o log de eventos seja bem formado e registre as informações de contexto de cada atividade realizada. Com base nesse fator, para efetuar a avaliação exposta nesse capítulo, foi construído um ambiente simulado, controlado e adaptado a partir da documentação um processo real, visto que uma abordagem para a execução de testes com o objetivo de avaliar algoritmos e técnicas de mineração de processo é a utilização de logs simulados (MEDEIROS & GÜNTHER, 2005), pois dessa forma se obtêm logs controlados sem imperfeições e ruídos que são comuns em logs de ambientes reais.

Além disso, a utilização de logs simulados proporciona um prévio conhecimento do que ocorre em cada atividade realizada, de forma que o resultado da aplicação do método deve refletir as regras que guiam as atividades controladas. Esse ambiente controlado permite a comparação do que deveria ser encontrado com o que efetivamente foi encontrado através da aplicação do método. Foi baseando-se nisso que o método de descoberta de regras de negócio foi avaliado e os detalhes dessa avaliação são expostos nas seções seguintes.

6.2 Cenário da Avaliação

Devido à dificuldade de acesso às informações de um log de eventos em uma organização real e pela necessidade de se obter um ambiente controlado para avaliar a aplicabilidade do método, optou-se pela adoção da criação de um processo fictício, sobre o qual foram feitas simulações de execuções para obter e registrar dados em um log. Este processo foi construído, no entanto, com base em um documento de especificação de um processo real de uma grande empresa petrolífera. Todas as informações presentes no processo de negócio utilizado na avaliação (exposto com detalhes na seção seguinte) foram alteradas com o intuito de proteger o caráter confidencial das informações, de forma que os passos, e em especial as informações envolvidas com os passos, foram mascaradas através de simplificação do fluxo e alteração de informações de atributos de contexto. Além disso, é omitida a especificação do significado de cada atributo de contexto relacionado às atividades do processo.

É importante ressaltar que a aplicação do método independe do mapeamento dos processos de negócios da organização. O motivo da utilização deste processo de negócio é somente trazer para um ambiente mais próximo da realidade possível os dados gerados de forma automática através de simulação.

6.3 O Processo de Negócio Utilizado

O modelo da Figura 6.1 representa o modelo de processos que foi utilizado na avaliação do método. Ele é composto por 9 atividades relacionadas com a requisição de estimativas de reservas.

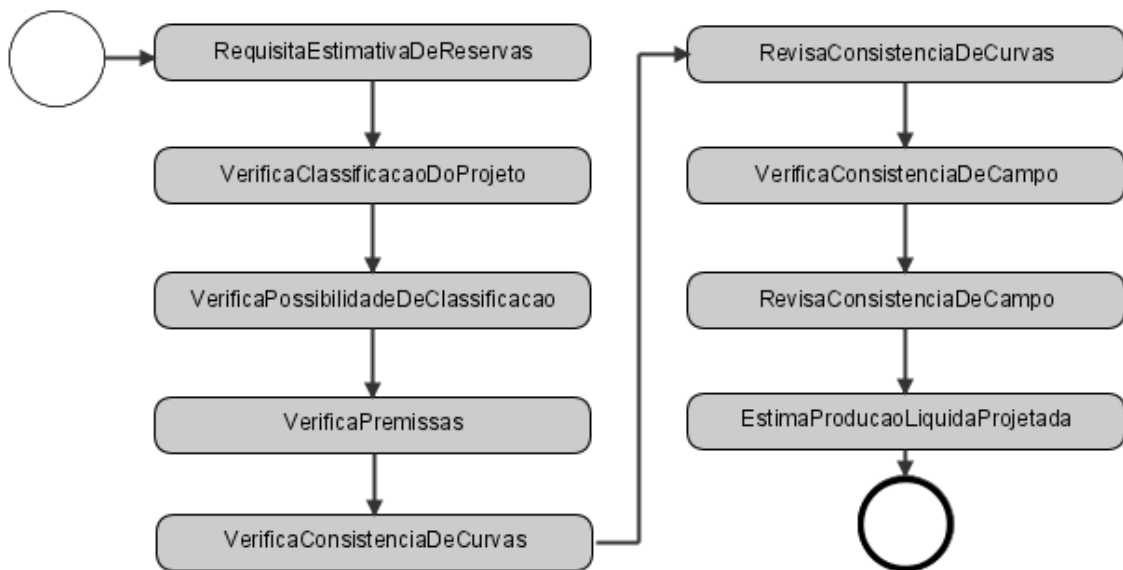


Figura 6.1 - Processo de Negócio adaptado utilizado na avaliação.

A seguir, serão expostos os atributos de contexto de cada atividade do processo adaptado. Por questões de confidencialidade, a descrição dos atributos foi omitida, bem como atributos não relevantes para essa avaliação.

A primeira atividade é a “RequisitaEstimativaDeReservas” que representa o registro do pedido para a realização da estimativa de reservas. Esta atividade possui como atributos de contexto os seguintes itens:

- Chave;
- Grupo;
- Exercício;
- Ano de Exercício;
- Exercício Referencia;
- Código do Campo;
- Nome do Campo;
- Un;

A segunda atividade é a “VerificaClassificacaoDoProjeto” que representa a etapa de análise sobre os dados de um projeto da organização com o objetivo de obter a classificação deste. Esta atividade possui como atributos de contexto os seguintes itens:

- Chave;
- Grupo;

- Exercício;
- Nome do Projeto;
- Código do Campo;
- Classificação do Projeto;
- Projeto já Classificado;

“VerificaPossibilidadeDeClassificacao” é a terceira atividade do processo, que representa a etapa onde o projeto é analisado para verificar se é possível classificar o projeto ou não. Esta atividade possui como atributos de contexto os seguintes itens:

- Chave;
- Grupo;
- Código do Campo;
- Nome do Projeto;
- Projeto é Classificável;

“VerificaPremissas” é a quarta atividade do processo, que representa a etapa onde diversas informações sobre a organização, sobre a unidade regional e sobre o campo (informações denominadas de premissas) são verificadas para identificar se elas estão preenchidas e indicar o local onde estão faltando informações. Esta atividade possui como atributos de contexto os seguintes itens:

- Chave;
- Grupo;
- Un;

A quinta atividade é a “VerificaConsistenciaDeCurvas” que representa a análise de consistência sobre os dados das curvas que é uma representação de um item do projeto. Esta atividade possui como atributos de contexto os seguintes itens:

- Grupo;
- Código do Campo;
- Un do Campo;
- Projeto;
- Somatório dos Pontos da Curva é Negativo;
- ExisteZpEsgotadaComCurvaDeProducaoDeFluido;

- SituacaoCurvas;

“RevisaConsistenciaDeCurvas” é a sexta atividade que representa a revisão da consistência sobre os dados das curvas que é uma representação de um item do projeto. Esta atividade possui como atributos de contexto os seguintes itens:

- Grupo;
- Somatório dos Pontos da Curva é Negativo;
- ExisteZpEsgotadaComCurvaDeProducaoDeFluido;
- SituacaoCurvas;
- SituacaoFinalCurvas;
- Justificativa;
- Projeto é Liberado;

“VerificaConsistenciaDeCampo” é a sétima atividade. Ela representa a análise da consistência de um campo com base nos seus dados. Esta atividade possui como atributos de contexto os seguintes itens:

- Grupo;
- Código do Campo;
- Un do Campo;
- Campo Possui Projeto Inconsistente;
- Situação Campo;

“RevisaConsistenciaDeCampo” é a oitava atividade, que representa a revisão sobre a análise realizada sobre a consistência de um campo. Esta atividade possui como atributos de contexto os seguintes itens:

- Grupo;
- Código do Campo;
- Campo Possui Projeto Inconsistente;
- Situação Campo;
- Situação Final Campo;
- Justificativa;
- Projeto é Liberado;

A nona e última atividade do processo é a “EstimaProducaoLiquidaProjetada” que representa o cálculo da estimativa de produção líquida de determinado campo. Esta atividade possui como atributos de contexto os seguintes itens:

- Chave;
- Grupo;
- Exercício;
- Ano de Exercício;
- Exercício Referencia;
- Código do Campo;
- Nome do Campo;
- Un;
- Produção Acumulada do Dia;
- Produção Acumulada do Dia Anterior;
- Produção Acumulada dos Últimos Dois Dias;
- Produção Líquida Projetada;
- Tempo Para Produzir;

6.4 Geração do Log de Eventos

Conforme já exposto, a geração de logs simulados em um ambiente controlado foi a abordagem selecionada para avaliar o método proposto nessa pesquisa, sendo realizada a geração do log de eventos através da modelagem do processo descrito na Seção 6.3 na ferramenta CPN Tools (CPNTOOLS, 2007, MEDEIROS & GÜNTHER, 2005) (Figura 6.2) e da simulação de execuções do processo modelado.

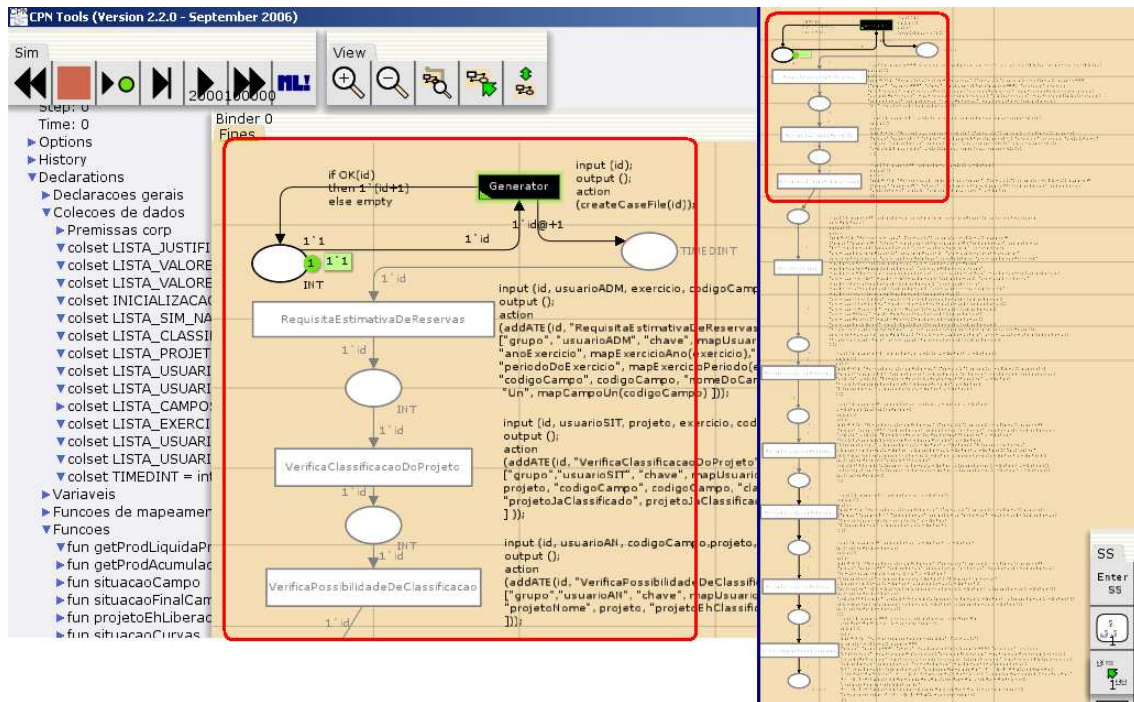


Figura 6.2 - Detalhe de parte do processo adotado nessa avaliação e modelado na ferramenta CPN Tools (CPNTOOLS, 2007).

O CPN Tools é uma ferramenta desenvolvida pela Universidade de Aarhus na Dinamarca, primeiro sob a designação de Design/CPN e posteriormente sob a designação atual, que tem funcionalidades para edição, simulação e análise das chamadas CP Nets (Coloured Petri Nets) (CPNETS, 2002), que é uma linguagem gráfica para projeto, especificação, simulação e verificação de sistemas.

Para o registro dos atributos de contexto e para o registro correto das regras definidas no processo de negócio, foi necessário um trabalho de construção de diversas fórmulas e outras estruturas em um modelo na ferramenta CPN Tools (conforme exemplificado no Apêndice A). Por exemplo, o conceito Campo possui 3 características: código do campo, nome do campo e unidade regional do campo. Foi necessário construir um controle no modelo para que todas as vezes que uma determinada instância de campo apareça no log de eventos, os mesmos valores para estas características devem aparecer, para manter a consistência do log gerado (Figura 6.3).

idEvento	codigoCampo	nomeCampo	unCampo	idEvento	codigoCampo	nomeCampo	unCampo
1	C1	nomeA	SP	1	C1	nomeA	SP
2	C1	nomeB	SC	2	C1	nomeA	SP
3	C2	nomeC	RJ	3	C2	nomeB	RJ
4	C1	nomeD	ES	4	C1	nomeA	SP
5	C1	nomeA	PA	5	C1	nomeA	SP
6	C2	nomeB	PR	6	C2	nomeB	RJ
7	C2	nomeC	DF	7	C2	nomeB	RJ
8	C1	nomeD	PE	8	C1	nomeA	SP

(a) (b)

Figura 6.3 - Representação de instâncias de uma atividade qualquer, exemplificando a falta de consistência dos atributos entre as instâncias (a) e a manutenção da consistência (b).

Ainda com base nesse exemplo do conceito Campo, durante cada execução da simulação no CPN Tools, um primeiro valor (código do campo) é selecionado aleatoriamente dentro de um escopo fechado que representa o conjunto de todos os campos da organização (Figura 6.4) e os demais valores são produzidos através dos resultados de fórmulas igualmente definidas para garantir a consistência dos dados gerados em instâncias diferentes. Esses mesmos procedimentos precisaram ser realizados para todos os atributos de contexto que representam uma informação que caracteriza um conceito e que não deve mudar em instâncias diferentes.

```

▼ colset LISTA_PROJETOS = subset STRING with ["alfa1","alfa2","beta1","beta3","beta4"
▼ colset LISTA_USUARIOS_SIC = subset STRING with ["Felipe","Romulo","Daniele","Sonia
▼ colset LISTA_USUARIOS_SIT = subset STRING with ["Marco","Adriana","Felipe","Romulo
▼ colset LISTA_CAMPOS = subset STRING with ["ORION1", "ORION2", "ORION3",
"SATURNO1", "SATURNO2", "SG1", "SG2", "SG3", "SG4", "UNI1", "UNI2", "UNI3", "UNI4",
"NIT1", "NIT2", "NIT3", "NIT4", "NIT5", "RIO1", "RIO2", "BR1", "BR2", "BR3", "BR4", "BR5"];
▼ colset LISTA_EXERCICIOS = subset STRING with ["EXE2004", "EXE2005", "EXE2006", "
▼ colset LISTA_USUARIOS_GRUPO_ADM = subset STRING with ["Ricardo", "Decio", "Rogeri
▼ colset LISTA_USUARIOS_GRUPO_AN = subset STRING with ["Marco", "Adriana", "Cristian

```

Figura 6.4 - Representação de listagens na ferramenta CPN Tools, com destaque para a lista de código de campos.

Já atributos de contexto que não possuem dependências com relação a outros atributos possuem seus valores selecionados de forma aleatória, variando dependendo da instância de execução. E nos casos em que existia alguma regra já definida na própria documentação do processo utilizado como base, foram criadas fórmulas especificamente para obter o valor a cada instância de execução. Por exemplo, para um projeto obter o estado de “projeto liberado” ele deve possuir o atributo “situacaoFinal”

com valor “CONSISTENTE” ou possuir justificativa com um valor “BoaJustificativa”, “OtimaJustificativa” ou “JustificativaRazoavel” (Figura 6.5).

```
▶ fun getProdLiquidoProjetoAvia(projetoAvia, tempoAtualProjeto) = projetoAvia div temp
▶ fun getProdAcumuladaDoisDias
▶ fun situacaoCampo
▶ fun situacaoFinalCampo
▼ fun projetoEhLiberado(situacaoFinal, justificativa) =
  if situacaoFinal = "CONSISTENTE" then "SIM"
  else (if justificativa = "BoaJustificativa" then "SIM"
        else (if justificativa = "OtimaJustificativa" then "SIM"
              else (if justificativa = "JustificativaRazoavel" then "SIM"
                    else ("NAO"))));
▶ fun situacaoCurvas
▶ fun situacaoFinalCurvas
```

Figura 6.5 - Representação de funções na ferramenta CPN Tools, com destaque para a função que define se um projeto é liberado.

Através da execução da simulação, foram gerados 8.000 arquivos (totalizando quase 60MB de logs), um para cada instância de execução do processo, e cada um destes contendo o conjunto de eventos de cada instância com seus atributos de contexto. O log de eventos total continha 72.000 eventos. A Figura 6.6 ilustra um trecho do log gerado.

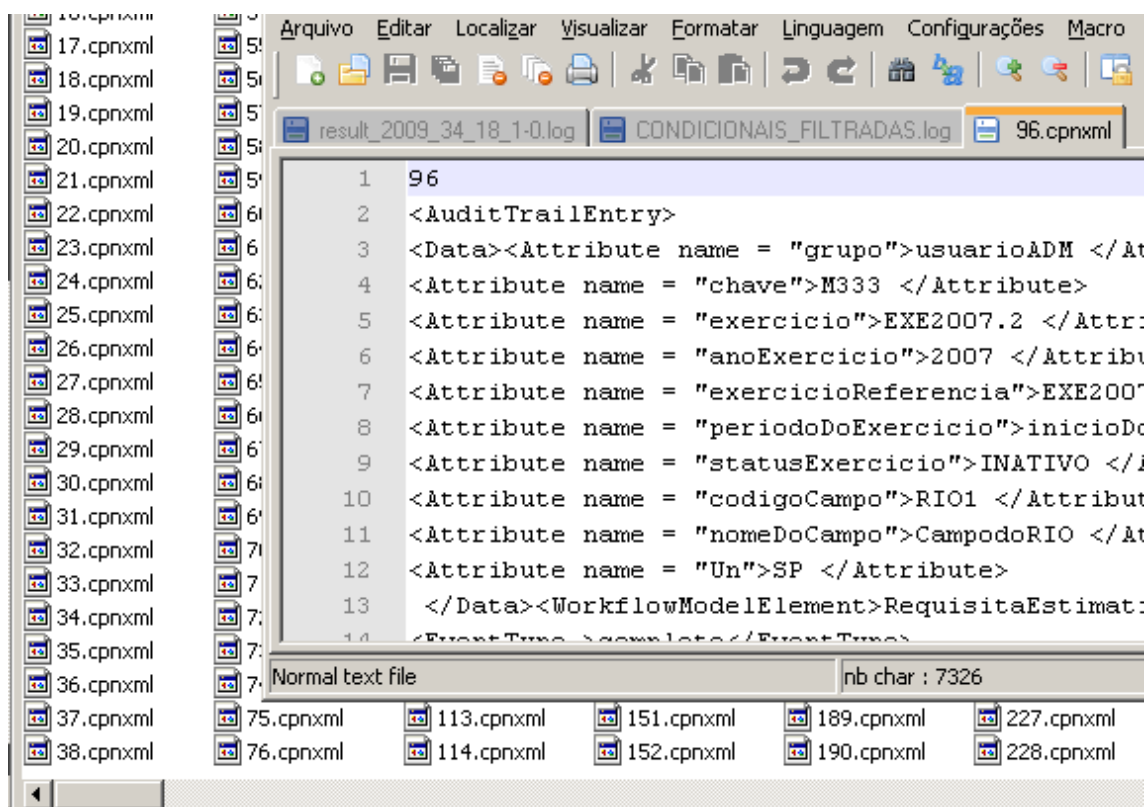


Figura 6.6 - Em destaque exemplo do log gerado e, ao fundo, uma parte dos arquivos gerados pela simulação.

6.5 Aplicação do Método e Resultados Obtidos

Com a aplicação do plug-in para conversão de logs da ferramenta CPN Tools para Mxml existente no ProM Import (MEDEIROS & GÜNTHER, 2005), foi gerado um Mxml com mais de um milhão de linhas, o que expressa a necessidade de mineração para se extrair o conhecimento. Após isso foi feita a aplicação dos plug-ins do ProM para a descoberta de sentenças de ação de autorização e sentenças de ação de condição, e o procedimento para a descoberta de sentenças estruturais também foi executado. Os resultados obtidos são expostos a seguir.

6.5.1 Resultados para as Sentenças de Ação de Autorização

No processo simulado cada uma das atividades foi direcionada, segundo a especificação do modelo de processos utilizado como base, a um determinado grupo de usuários, e após a simulação e a execução do plug-in desenvolvido no ProM, foram encontradas as seguintes regras:

*It is obligatory that REQUISITAESTIMATIVADERESERVAS be run only
if GRUPO is USUARIOADM;*

*It is obligatory that VERIFICACCLASSIFICACAODOPROJETO be run only
if GRUPO is USUARIOSIT;*

*It is obligatory that VERIFICAPOSSIBILIDADEDECLASSIFICACAO be
run only
if GRUPO is USUARIOAN;*

*It is obligatory that VERIFICAPREMISSAS be run only
if GRUPO is USUARIOAN;*

*It is obligatory that VERIFICACONSISTENCIADECURVAS be run only
if GRUPO is USUARIOSIC;*

*It is obligatory that REVISACONSISTENCIADECURVAS be run only
if GRUPO is USUARIOAN;*

*It is obligatory that VERIFICACONSISTENCIADECAMPO be run only
if GRUPO is USUARIOSIT;*

*It is obligatory that REVISACONSISTENCIADECAMPO be run only
if GRUPO is USUARIOAN;*

*It is obligatory that ESTIMAPRODUCAOLIQUIDAPROJETADA be run
only
if GRUPO is USUARIOADM;*

6.5.2 Resultados para as Sentenças de Ação de Condição

Conforme exposto na Seção 6.4, existem casos em que funções que são responsáveis pela formação de valor de um determinado atributo de contexto foram incluídas no modelo na ferramenta CPN Tools. A seguir são expostas algumas destas funções e são apresentados resultados importantes puderam ser observados através da simulação.

A Figura 6.7 apresenta 3 funções que foram utilizadas para a definição de 3 atributos de contexto da atividade “RevisaConsistenciaDeCurvas”: “Situação das Curvas”, “Situação Final das Curvas” e “Projeto é liberado”. Para a definição de “Situação das Curvas”, a função “situacaoCurvas” recebe duas variáveis como parâmetro que indicam se “Somatório dos Pontos da Curva é Negativo” e se “ExisteZpEsgotadaComCurvaDeProducaoDeFluido” possuem valor “SIM” ou “NAO”.


```

▼ fun projetoEhLiberado(situacaoFinal, justificativa) =
  if situacaoFinal = "CONSISTENTE" then "SIM"
  else (if justificativa = "BoaJustificativa" then "SIM"
        else (if justificativa = "OtimaJustificativa" then "SIM"
              else (if justificativa = "JustificativaRazoavel" then "SIM"
                    else ("NAO"))));
▼ fun situacaoCurvas(cond1, cond2) = if cond1 = "SIM"
  then "INCONSISTENTE"
  else ( if cond2 = "SIM" then "INCONSISTENTE"
        else "CONSISTENTE");
▼ fun situacaoFinalCurvas(cond, curvasSituacao) =
  if curvasSituacao = "CONSISTENTE" then "CONSISTENTE"
  else ( if cond = "SIM" then "INCONSISTENTE"
        else ( if cond = "NAO" then "CONSISTENTE"
              else "INCONSISTENTE"));

```

Figura 6.7 - Funções incluídas no modelo criado na ferramenta CPN tools, para controle de atributos de alguns contextos.

O atributo de contexto “Situação Final das Curvas” obtém seu valor da função “situacaoFinalCurvas”, através do valor da situação da curva original e de um segundo valor aleatório. Já o atributo “Projeto é liberado” utiliza o valor encontrado na “Situação Final das Curvas” e o valor do atributo de contexto ”justificativa” para definir seu valor. Com base nessas regras, a mineração expôs as seguintes regras definindo “Situação Curvas”:

```

if ExisteZpEsgotadaComCurvaDeProducaoDeFluido is NAO
and SomatorioDosPontosDasCurvasEhNegativo is NAO
    then SituacaoCurvas is CONSISTENTE;
if ExisteZpEsgotadaComCurvaDeProducaoDeFluido is NAO
and SomatorioDosPontosDasCurvasEhNegativo is SIM
    then SituacaoCurvas is INCONSISTENTE;
if ExisteZpEsgotadaComCurvaDeProducaoDeFluido is SIM
    then SituacaoCurvas is INCONSISTENTE;

```

Essas próximas regras foram encontradas definindo “Projeto é liberado”:

```

if justificativa is 0
    then projetoEhLiberado is SIM;
if justificativa is JustificativaPessima
    then projetoEhLiberado is NÃO;
if justificativa is JustificativaRuim
    then projetoEhLiberado is NÃO;
if justificativa is OtimaJustificativa

```

then projetoEhLiberado is SIM;
if justificativa is JustificativaRazoavel
then projetoEhLiberado is SIM;
if justificativa is BoaJustificativa
then projetoEhLiberado is SIM;

No caso do atributo de contexto “Situação Final das Curvas”, foram encontradas as regras expostas a seguir:

if justificativa is 0
then SituacaoFinalCurvas is CONSISTENTE;
if justificativa is JustificativaPessima
then SituacaoFinalCurvas is INCONSISTENTE;
if justificativa is JustificativaRuim
then SituacaoFinalCurvas is INCONSISTENTE;
if justificativa is OtimaJustificativa
then SituacaoFinalCurvas is INCONSISTENTE;
if justificativa is JustificativaRazoavel
then SituacaoFinalCurvas is INCONSISTENTE;
if justificativa is BoaJustificativa
then SituacaoFinalCurvas is INCONSISTENTE;

Para a atividade “RevisaConsistenciaDeCampo”, a formação das funções é similar à formação das funções das regras expostas acima. Consequentemente a estrutura das regras seguiu um padrão bem similar como pode ser visto abaixo. Para o atributo “Situação do Campo”, foram encontradas as seguintes regras:

if CampoPossuiProjetoInconsistente is SIM
then SituacaoCampo is INCONSISTENTE;
if CampoPossuiProjetoInconsistente is NÃO
then SituacaoCampo is CONSISTENTE;

Para o atributo “Projeto é liberado”, foram encontradas as seguintes regras:

if justificativa is JustificativaRuim
then projetoEhLiberado is NÃO;
if justificativa is OtimaJustificativa

then projetoEhLiberado is SIM;
if justificativa is JustificativaRazoavel
then projetoEhLiberado is SIM;
if justificativa is 0
then projetoEhLiberado is SIM;
if justificativa is BoaJustificativa
then projetoEhLiberado is SIM;
if justificativa is JustificativaPessima
then projetoEhLiberado is NÃO;

E o atributo de contexto “Situação Final Campo”, obteve como definição às seguintes regras:

if justificativa is JustificativaRuim
then SituacaoFinalCampo is INCONSISTENTE;
if justificativa is OtimaJustificativa
then SituacaoFinalCampo is INCONSISTENTE;
if justificativa is JustificativaRazoavel
then SituacaoFinalCampo is INCONSISTENTE;
if justificativa is 0
then SituacaoFinalCampo is CONSISTENTE;
if justificativa is BoaJustificativa
then SituacaoFinalCampo is INCONSISTENTE;
if justificativa is JustificativaPessima
then SituacaoFinalCampo is INCONSISTENTE;

6.5.3 Resultados para as Sentenças Estruturais

A mineração das regras estruturais foi realizada com base nos procedimentos descritos na Seção 5.4. Através deste procedimento, foram encontrados 4 elementos que agrupam alguns atributos de contexto. Abaixo estão representados estes elementos referentes à estrutura do negócio que foram encontrados:

“Elemento1” has “anoExercicio”;
“Elemento1” has “exercicio”;
“Elemento1” has “exercicioReferencia”;
“Elemento2” has “codigoCampo”;

“Elemento2” has “nomeDoCampo”;
“Elemento3” has “CampoPossuiProjetoInconsistente”;
“Elemento3” has “SituacaoCampo”;
“Elemento4” has “Grupo”;
“Elemento4” has “SituacaoCampo”.

6.5.4 Avaliação dos Resultados

Essa seção visa fazer uma análise crítica sobre os resultados obtidos com a realização da simulação descrita nesse capítulo. Desta forma, são apresentados abaixo os pontos positivos e negativos da aplicação do método para os três tipos de regras de negócios para os quais o método foi implementado: sentenças de ação de autorização, sentenças de ação de condição e sentenças estruturais.

Na aplicação do método para descoberta de sentenças de ação de autorização sobre os dados simulados, foram extraídas todas as regras esperadas e não foram encontrados problemas. Os resultados obtidos mostram indícios de que a aplicação do método para este tipo de regra de negócio é viável de ser aplicada a um cenário real. Além disso, o método propõe a realização de uma busca exaustiva para a descoberta deste tipo de regra que, pela sua simplicidade, pode apresentar um custo muito alto em situações reais. Apesar de reconhecidamente importante, o aspecto de desempenho dos algoritmos de descoberta não foi avaliado no presente trabalho. Isto, no entanto, não impacta a aplicabilidade do método, uma vez que tipicamente estes procedimentos não serão executados com frequência, e nem em tempo (ou no mesmo ambiente) de execução dos sistemas de informação que estão sendo analisados.

Para as regras do tipo sentenças de ação de condição, o resultado também foi positivo, uma vez que as regras esperadas foram encontradas. No entanto, a quantidade de padrões de dados encontrados com a realização da classificação que não correspondiam a sentenças de ação de condição foi elevada (o documento gerado com a especificação das regras em SBVR de cerca de 500 páginas). Observou-se também que a acurácia dos resultados (métrica que avalia os modelos de classificação a partir da porcentagem de predições corretas que o modelo executou sobre o total de predições realizadas), obtida diretamente dos resultados da aplicação do algoritmo J48 (WEKA, 2009), foi alta para os padrões que representavam as sentenças de ação de condição (maior que 90%) e baixa para os inúmeros padrões que não se referiam a regras (menor que 50%). Conclui-se, portanto, que o estabelecimento de um valor mínimo de corte

para a identificação das regras que devem ser apresentadas ao especialista do negócio poderia guiar o algoritmo a obter apenas as regras significativas.

Ainda com relação à descoberta das sentenças de ação de condição, outro problema identificado é a descoberta de mais de uma regra, onde era esperada apenas uma, pois ao realizar a mineração de 2 atributos de contexto que se relacionam, são descobertas duas regras que representando a relação entre eles e que devem ser analisadas pelos analistas de negócio (Figura 6.8). O problema com relação a essa regra é somente a adição de informação desnecessária para o especialista do negócio.

```
@ Usando como classificador: SituacaoCampo
if CampoPossuiProjetoInconsistente is SIM then SituacaoCampo is INCONSISTENTE
if CampoPossuiProjetoInconsistente is NAO then SituacaoCampo is CONSISTENTE

@ Usando como classificador: CampoPossuiProjetoInconsistente
if SituacaoCampo is INCONSISTENTE then CampoPossuiProjetoInconsistente is SIM
if SituacaoCampo is CONSISTENTE then CampoPossuiProjetoInconsistente is NAO
```

Figura 6.8 - Duas regras relacionando os mesmos atributos de contexto.

Com relação à execução do método para a descoberta das sentenças estruturais, pode ser observado que a realização da descoberta é viável, pois foram descobertos 4 conceitos contendo 9 atributos relacionados. No entanto é importante ressaltar que a análise dos resultados por parte dos especialistas do negócio é essencial. Por exemplo, das sentenças estruturais encontradas, as duas que estão repetidas logo abaixo e definem um elemento denominado “Elemento 4”, foram encontradas com base na mineração sobre o log de eventos mas não representam uma regra real do negócio. Na verdade enquanto a regra afirma que os atributos de contexto “Grupo” e “SituacaoCampo” pertencem a um único elemento do negócio, eles pertencem respectivamente ao usuário do sistema e ao “Campo” referenciado na atividade.

“Elemento4” has “Grupo”;

“Elemento4” has “SituacaoCampo”;

Além disso, ocorreram problemas de desempenho associados à execução da mineração de dados através de associação nas atividades com muitos atributos de contexto. O passo da realização da mineração de dados em si não pode ser efetuada completamente para a atividade “verificarPremissas” que é a atividade que mais continha atributos de contexto. Isso sinaliza que a aplicação do método para este tipo de

regras em ambientes reais pode ser comprometida. Neste caso, será necessária a realização de pré-processamento nos dados para reduzir o número de atributos de contexto para a realização de associação.

6.5.5 Conclusões

A avaliação demonstrou que o método para a descoberta de regras de negócio pode ser viável em ambientes reais, mas, no entanto, será necessário fazer evoluções no que se refere ao ajuste dos resultados expostos aos especialistas do negócio (no caso das sentenças de ação de condição) e ao desempenho (no caso das sentenças estruturais). Um importante passo futuro seria a elaboração de um estudo de caso com base em dados reais.

7 Conclusão

Este trabalho abordou o problema da descoberta e representação (semi-) automatizadas de regras de negócio através de logs de eventos de sistemas de informação. Isto ocorre com base nos registros das execuções dos sistemas de informação, que armazenam comportamento gerado pela presença das regras de negócio, e a partir deles é possível extrair padrões que identificam as regras de negócio correntes na organização. A relevância deste tema é decorrente da importância das regras de negócio para as organizações e da importância de se buscar nos sistemas de informação as regras que realmente estão guiando as atividades na organização.

Os principais resultados obtidos foram a criação de uma proposta de arquitetura e de um método para a descoberta de regras de negócio, bem como da implementação deste método para 3 tipos distintos de regras de negócio: sentenças de ação de autorização, sentenças de ação de condição e sentenças estruturais.

A seguir são apresentadas comparações desse método com outros trabalhos, as contribuições do trabalho e os trabalhos futuros.

7.1 Comparações com Outros Trabalhos

Rozinat e Aalst (ROZINAT & AALST, 2006) utilizam mineração de processo para encontrar o que denominam de “pontos de decisão” (*decision points*), que podem ser consideradas sentenças de ação de condição. No entanto, o foco desta abordagem é sempre sobre o ponto de vista do fluxo das atividades, enquanto no caso da proposta apresentada nesta pesquisa, a avaliação é feita sobre os atributos de contexto de cada atividade, independente da relação da atividade com as outras atividades do workflow, o que torna o resultado mais amplo no contexto das sentenças de ação de condição.

SULAIMAN (2002) apresenta uma abordagem para a descoberta de fluxos de trabalho e regras de negócio baseando-se em dados gerados pelos Sistemas de

Gerenciamento de Banco de Dados (SGBDs). Segundo o próprio autor apresenta, essa abordagem é muito dependente da aplicação-alvo, pois os dados endógenos são gerados pelos próprios SGBDs e a tarefa de extração desses dados precisa ser feita “sob medida”. Neste ponto a abordagem apresentada nessa dissertação é vantajosa, visto que o passo de obtenção de dados pode inclusive estar pronto na ferramenta ProM Import adotada na implementação do método. Além disso, o autor afirma que os logs de auditoria são armazenados através da inclusão de comandos de registro para rastreamento de determinadas transações nos sistemas, enquanto os logs de SGBDs são considerados, muitas vezes, custosos. Este fator gera uma limitação para a proposta de SULAIMAN e é um fator positivo para aplicação do método desenvolvido e proposto nessa dissertação, visto que ele aponta para a tendência de registro de logs de auditoria, que são logs de eventos de atividades realizadas.

SIQUEIRA (2002) propôs a extração de regras de negócio de sistemas legados em ambiente Unisys, visando resolver o problema de divergência entre as regras documentadas no início do sistema para as que efetivamente passam a existir no sistema após anos de sua utilização. Para alcançar seu foram desenvolvidas heurísticas específicas para o ambiente Unisys, tornando a abordagem também específica para o ambiente onde se aplica (sistemas em COBOL e banco DMSII). Nesse ponto, o método proposto nessa pesquisa pode ser aplicado a sistemas de informação desenvolvidos em qualquer ambiente, não é dependente de uma tecnologia específica.

7.2 Contribuições

As principais contribuições deste trabalho são listadas a seguir.

Arquitetura proposta: foi proposta uma arquitetura baseada na extração de conhecimento de logs de eventos de sistemas de informação, com o objetivo de se extrair regras de negócios e expô-la para especialistas do negócio.

Método proposto: com base nessa arquitetura proposta, foi desenvolvido um método composto de etapas para que seja possível realizar a mineração das regras de negócio. Foram definidas tarefas de mineração de dados mais adequadas para descobrir regras de negócio de cada tipo. Versões preliminares do método proposto constam em CRERIE *et al.* (2008) e CRERIE *et al.* (2009).

Implementação do método: O método foi implementado para três tipos de

regras de negócio: sentenças de ação de autorização, sentenças de ação de condição e sentenças estruturais, aplicando-se algoritmos existentes a plug-ins desenvolvidos como extensão ao framework ProM.

Resultados da avaliação do método: Uma avaliação do método foi realizada com o objetivo de verificar se a extração de regras de negócio é efetuada e analisar os resultados de sua aplicação. Os resultados disponíveis mostraram a viabilidade do método em descobrir e representar regras de negócio em SBVR a partir de logs de eventos de sistemas de informação.

7.3 Limitações e Trabalhos Futuros

O escopo desta dissertação não incluiu a implementação da descoberta de todos os tipos de regras de negócio.

O método proposto assume que existam logs de eventos disponíveis para extração de conhecimento. Além disso, as regras descobertas refletem apenas a correlação entre as informações registradas no log.

Diante da inexistência de log de determinado sistema, devem ser realizadas alterações ligadas à execução dele para que se passe a armazenar esse log. Para o caso em que apenas uma determinada informação essencial não seja registrada, é possível buscá-la na base de dados que o sistema utiliza e incluí-la nos eventos presentes no log de execuções, mas caso não haja registro algum na base de dados, a descoberta se torna inviável. É, portanto, extremamente importante para a aplicação desta abordagem, que todas as informações relacionadas à execução de cada atividade sejam incluídas como atributos de contexto, pois são estas informações que serão utilizadas para descoberta das regras de negócio. Portanto, se um sistema de informação não registra um log de eventos de qualidade, a qualidade final da mineração é certamente impactada.

Outra limitação existente possui relação com a abrangência do método, no contexto dos tipos de regras de negócio existentes. Nesta dissertação foi apresentada a implementação do método para 3 tipos. A proposta de como descobrir os demais tipos de regras foi delineada, mas ficou fora do escopo da avaliação realizada.

Além disso, a proposta desta dissertação limita-se à descoberta de regras de negócio que estejam automatizadas em sistemas de TI. Por exemplo, utilizar o crachá nas dependências da organização pode ser uma regra de negócio que não é tratada computacionalmente. É evidente que não é possível o registro de todo tipo de

conhecimento humano explicitamente (SULAIMAN, 2002) *apud* (DAVENPORT & PRUSAK, 1998).

Outra limitação encontrada foi o problema de desempenho encontrado durante a avaliação da descoberta de sentenças estruturais. Um problema causado pela quantidade de atributos de contexto pode restringir a realização da descoberta de regras de negócios deste tipo e deve ser analisado futuramente.

Outra possível aplicação desse método em um trabalho futuro, refere-se não à descoberta das regras de negócio simplesmente, mas à descoberta de eventos que indiquem violações às regras encontradas (*outliers*).

Por fim, a união do método apresentado nesse trabalho com abordagens que buscam informações em bases de dados, como o trabalho do SULAIMAN (2002) e NAVATHE *et al.* (1995), pode representar uma importante evolução na pesquisa de descoberta de regras de negócio.

7.4 Considerações Finais

De uma forma geral, o método proposto, segundo suas características, se apresenta com uma tendência de ser mais abrangente, por não depender de uma tecnologia específica. Isto é um ponto positivo, mas por outro lado é necessário que a premissa de que exista o registro do log de eventos para que a aplicação do método possa ser realizada. A realização da avaliação apresentada nesta dissertação mostrou que a aplicação do método é viável para realizar a descoberta de regras ao aplicá-lo sobre um log de eventos baseado em um cenário real.

Referências

AALST, W. M. P. van der, Dongen, B. F. van, Herbst, J. *et al.*, 2003, “Workflow Mining: A Survey of Issues and Approaches”, *Data and Knowledge Engineering*, vol. 47, 2, pp.237-267.

AALST, W. M. P. van der, DE BEER, H. T., VAN DONGEN, B. F., 2005, “Process mining and verification of properties: An approach based on temporal logic”, In: *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005*, pp. 130-147, vol. 3760, Agia Napa, Cyprus, outubro, 2005.

AALST, W. M. P. van der, GÜNTHER, C. W., 2007, “Finding Structure in Unstructured Processes: The Case for Process Mining”, In: *ACSD Proceedings of the Seventh International Conference on Application of Concurrency to System Design*, pp. 3-12, Washington DC., USA, IEEE Computer Society, 2007.

AALST, W.M.P. van der, WEIJTERS, A. J. M. M., 2005, “Process Mining”, In Dumas, M.; Aalst, W.M.P. van der; Hofstede, A.H.M. ter (eds.), *Process-Aware Information Systems: Bridging People and Software through Process Technology*, USA, Wiley & Sons.

ALBERTI, M., CHESANI, F., GAVANELLI, M. *et al.*, 2007, “A Computational Logic-based Approach to Verification of IT Systems”, In: *Proceedings of the 14th*

Annual Workshop of HP Software University Association, pp. 115-125, Munich, Germany, Julho, 2007.

AMGHAR, Y., MEZAINÉ, M., FLORY, A., 2003, “Modeling of business rules for active database application specification”, In: SIAU, K. (ed), *Advanced topics in database research*, vol. 1, pp. 135-156, Hershey, PA, USA, IGI Publishing, 2003.

ARFF, 2009, *Attribute-Relation File Format*, <http://www.cs.waikato.ac.nz/~ml/weka/arff.html>, acessado em janeiro de 2009.

BEZERRA, F., WAINER, J., 2008, “Fraud Detection in Process Aware Systems”, In: *Anais do XIV Simpósio Brasileiro de Sistemas Multimídia e Web, II Workshop de Gestão de Processos de Negócio (WBPM)*, vol. 2, Vila Velha, ES, Brasil, Outubro, 2008.

BRG (Business Rules Group), 2000, *Defining Business Rules ~ What Are They Really?*, Rev. 1.3, 2000, http://www.businessrulesgroup.org/first_paper/BRG-whatIsBR_3ed.pdf, acessado em junho de 2008.

BRG (Business Rules Group), 2003, *Business Rules Manifesto*, Versão 2.0, <http://www.businessrulesgroup.org/brmanifesto.htm>, acessado em dezembro de 2008.

BRG (Business Rules Group), 2009, Business Rules Group Home Page, <http://www.businessrulesgroup.org>, acessado em janeiro de 2009.

CHEN, P.P., 1976, “The Entity-Relationship Model: Toward a Unified View of Data”, In: *ACM Transactions on Database Systems (TODS)*, vol.1, issue 1, pp.9-36, Framingham, MA, USA, Março, 1976.

CPNTOOLS, 2007, *Computer Tool for Coulored Petri Nets*, <http://wiki.daimi.au.dk/cpntools/cpntools.wiki>, acessado em julho de 2008.

CPNETS, 2002, CPNets Home Page, <http://www.daimi.au.dk/CPnets/intro/>, acessado em agosto de 2009.

CRERIE, R., BAIÃO, F., SANTORO, F. M., 2008, “Identificação de Regras de Negócio utilizando Mineração de Processos”, In: *Anais do XIV Simpósio Brasileiro de Sistemas Multimídia e Web, II Workshop de Gestão de Processos de Negócio (WBPM)*, vol. 2, Vila Velha, ES, Brasil, Outubro, 2008.

CRERIE, R., BAIÃO, F., SANTORO, F. M., 2009, “Discovering Business Rules through Process Mining”, In: *Enterprise, Business-Process and Information Systems Modeling, Workshop on Business Process Modeling, Development, and Support (BPMDS) at CAISE 2009*, Springer, vol. 29, pp. 136-148.

DATE, C. J., 2000, *What Not How: The Business Rules Approach to Application Development*, 1ª edição, Addison-Wesley, New Jersey, USA, 2000.

DAVENPORT, T H., PRUSAK, L., 1998, *Conhecimento Empresarial*, 14ª edição, Ed. Campus, Rio de Janeiro, Brasil.

DIAS, F., MORGADO, G., OSCAR, P., *et al.*, 2006, “Uma Abordagem para a Transformação Automática do Modelo de Negócio em Modelo de Requisitos”, In: *Anais do WER06 - Workshop em Engenharia de Requisitos*, pp 51-60, Rio de Janeiro, RJ, Brasil, Julho, 2006.

DONGEN, B. van, MEDEIROS A., VERBEEK, H., *et al.*, 2005, “The ProM Framework: A New Era in Process Mining Tool Support”, In: Ciardo, G., Darondeau, P. (Eds.), *Application and theory of Petry Nets 2005*, vol. 3536, pp. 444-454.

EindhovenTU, 2009, *User manual for converting data from a Microsoft Access Database to the ProM MXML format*, <http://prom.win.tue.nl/research/wiki/promimport/tutorials>, acessado em 23/05/2009.

FAYYAD, U. M., Piatetsky-Shapiro, G., SMYTH, P., *et al.*, 1996, *Advances in Knowledge Discovery and Data Mining*, Menlo Park, Califórnia, USA, AAAI/MIT Press.

FAYYAD, U., PIATETSKY-SHAPIRO, G., PADHRAIC, S., 1996, “The kdd process for extracting useful knowledge from volumes of data”, *Communications of the ACM*, vol. 39, issue 11, pp.27-34.

GOEDERTIER, S., VANTHIENEN, J., 2005. “Rule-based business process modeling and execution”, In: *CTIT Workshop Proceeding Series*, International IEEE EDOC Workshop on Vocabularies, Ontologies and Rules for the Enterprise (VORTE), pp. 67-74.

GUILARDUCCI, G. A., 2007, *Uma Abordagem para Tratamento de Regras de Negócio em Sistemas de Informação*, M.Sc., Instituto de Informática, Universidade Federal de Goiás, Goiânia.

GÜNTHER, C., RINDERLE, S., REICHERT, M., *et al.*, 2007, “Using Process Mining to Learn from Process Changes in Evolutionary Systems”, *International Journal of Business Process Integration and Management, Special Issue on Business Process Flexibility*, 3, pp. 61-78.

GÜNTHER, C. W., AALST, W. M. P. van der, 2006, “A generic import framework for process event logs”, In: Eder, J., Dustdar, S. (Eds.), *Business Process Management Workshop, Proceedings BPM 2006 International Workshops*, Pp81-92.

HALLE, B. V., 2002, *Business Rules Applied: Building Better Systems Using the Business Rules Approach*, 1ª ed., New York, NY, USA, John Wiley & Sons.

HAN, J., KAMBER, M., 2000, *Data Mining: Concepts and Techniques*, 1ª edição, Morgan Kaufmann, USA.

HAN, J., KAMBER, M., 2006, *Data Mining: Concepts and Techniques*, 2ª edição, Morgan Kaufmann, USA.

HAND, D., MANNILA, H., SMYTH, P., 2001, *Principles of data Mining*, Massachusetts, USA, MIT Press.

HORROCKS, I., PATEL-SCHNEIDER, P., F., BOLEY, H. *et al.*, 2004, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. W3C Member Submission. <http://www.w3.org/Submission/SWRL/>, acessado em julho de 2008.

KAARST-BROWN, M., KELLY, S., 2005, “IT governance and Sarbanes-Oxley: the latest sales pitch or real challenges for the IT function?”, *Proceedings of the 38th Hawaii International Conference on System Sciences*, Hawaii, USA.

KAMADA, A., 2006, *Execução de serviços baseada em regras de negócio*. D.Sc., Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Brasil.

KOVACIC, A., GROZNIK, A., 2004, “The business rule-transformation approach”, In: *Information technology interfaces*, vol.1, pp. 113-117, Cavtat, CROATIE, junho, 2004.

LAROSE D. T., 2005, *Discovering Knowledge in Data: An Introduction to Data Mining*, 1ª edição, John Wiley & Sons, New Jersey, USA, 2005.

LEITE, J. C. S. P., CERQUEIRA, P., 1995, “Recovering business rules from structured analysis specifications”, In: Proceedings of the WCRE95, IEEE Computer Society Press, pp.61-70.

MALLENS, P., 1997, “Business Rules-Based Application Development”, Database Newsletter, vol. 25, 1997.

MARTINS, A. E., 2006, *Em direção à captura e representação sistemática das definições dos termos das regras de negócio*, M.Sc., Núcleo de Computação Eletrônica, Instituto de Matemática, Universidade Federal do Rio de Janeiro.

MEDEIROS, A. K. A., 2006, *Genetic Process Mining*, Ph.D., Eindhoven Technical University, Eindhoven, Holanda.

MEDEIROS, A. K. A., PEDRINACI, C., AALST, W. M. P. van der *et al.*, 2007, “An Outlook on Semantic Business Process Mining and Monitoring”. In: On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops, OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, OTM Academy Doctoral Consortium, MONET, OnToContent, ORM, PerSys, PPN, RDDS, SSWS, and SWWS 2007, vol. 2, pp. 1244-1255, Vilamoura, Portugal, Novembro, 2007.

MEDEIROS, A. K., GÜNTHER, C.W., 2005, “Process Mining: Using CPN Tools to Create Test Logs for Mining Algorithms”, In: K. *Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2005)*, vol. 576, Aarhus, Dinamarca, outubro, 2005.

MEDEIROS, A. K A., AALST, W. M. P. van der, PEDRINACI, C., 2008, "Semantic Process Mining Tools: Core Building Blocks", In: *16th European Conference in Information Systems (ECIS)*, pp. 1953-1964, Galway, Irlanda, junho, 2008.

MITCHELL, T. M., 1997, *Machine Learning*, USA, McGraw-Hill.

MIRABETE, J. F., 2003, *Processo Penal*, 15ª edição, São Paulo, Brasil, Editora Atlas.

MORGADO, G. P., DIAS, F. G., MARTINS, A., E. *et al.*, 2004, "Um Ambiente para Modelagem Organizacional Baseado em Regras de Negócio", *Simpósio Brasileiro de Sistemas de Informação*, Porto Alegre, Brasil, 2004.

MORGADO, G. P., GESSER, I, SILVEIRA, D. S., *et al.*, 2007, "Práticas do CMMI como Regras de Negócio", *Revista Produção*, vol. 17, n. 2, pp.383-394.

MORGAN, T., 2002, *Business rules and information systems: aligning IT with business goals*, 1ª edição, Addison Wesley Professional, Unisys Series, Boston, MA, USA.

MXML, 2009, Mxml schema, <http://is.tm.tue.nl/research/processmining/WorkflowLog.xsd>, acessado em 11/04/2009.

NAVATHE, S., TANAKA, A., MADHAVAN, R., *et al.*, 1995, "A Methodology for Application Design Using Active Database Technology", *Final Technical Report, Rome Laboratory, Griffiss Air Force Base, New York*, março, 1995.

OMG, 2003, *Business Semantics of Business Rules - Request For Proposal*, *OMG Document*, <http://www.omg.org/docs/br/03-06-03.pdf>, acessado em 20/02/2009.

OMG, 2006. *Business motivation Model (BMM) Specification*, <http://www.omg.org/docs/dtc/06-08-03.pdf>, acessado em abril de 2008.

OMG, 2006, *Meta Object Facility (MOF) Core, Version 2.0*, <http://www.omg.org/spec/MOF/>, acessado em fevereiro de 2009.

OMG, 2006, *Object Constraint Language, OMG Available Specification, Version 2.0*, <http://www.omg.org/docs/formal/06-05-01.pdf>, acessado em julho de 2008.

OMG, 2005, *MOF 2.0/XMI Mapping Specification, v2.1*, <http://www.omg.org/docs/formal/05-09-01.pdf>, acessado em fevereiro de 2009

OMG, 2008, *Semantics of Business Vocabulary and Rules*, <http://www.omg.org/spec/SBVR>, acessado em dezembro de 2008.

OMG, 2009, *Object Management Group*, www.omg.org, acessado em fevereiro de 2009.

OMG, 2009, *Model Driven Architecture page*, <http://www.omg.org/mda/>, acessado em 10/04/2009.

PETRUSEL, R., 2009, “Collaborative Virtual Enterprise Environment and Decision Mining”, *Informatica Economica Journal*, vol. 13, Issue 2, pp. 59-67, Romênia.

PHILLIPS, T., 2002, *Versata Method Handbook*, v. 1.4, Oakland, Califórnia, Versata Inc, disponível em Versata Inc. 2101 Webster Street 8th Floor, Oakland, CA, USA.

POO, D. C. C., 1999, “Events in Use Cases as a Basis for Identifying and Specifying Classes and Business Rules”, *Proceedings of the Technology of Object-Oriented Languages and Systems. Technology of Object-Oriented Languages and Systems*, 1999.

PROM Import, 2006, *Swiss army knife for event logs*, <http://prom.win.tue.nl/tools/promimport>, acessado em agosto de 2008.

PROM, 2008, *The ProM Framework*, <http://prom.win.tue.nl/tools/prom/>, acessado em abril de 2008.

PUTRYCZ, E., KARK, A. W., 2007, “Recovering Business Rules from Legacy Source Code for System Modernization”, In: Paschke, A., Biletskiy, Y. (eds.), *Advances in Rule Interchange and Applications, International Symposium, RuleML 2007*, vol. 4824/2007, pp. 107-118, Orlando, Florida, USA, Novembro, 2007.

RAM, S., KHATRI, V., 2005, “A comprehensive framework for modeling set-based business rules during conceptual database design”, In: *Information Systems*, Volume 30, Issue 2, pp.89-118, Oxford, UK, Abril, 2005.

ROSS, R. G., 2000, “Expressing Business Rules”, In: *ACM SIGMOD Record*, vol. 29 , Issue 2, pp.515-516, New York, NY, USA, Junho, 2000.

ROSS, R. G., 2001, “The BRS Rule Classification Scheme”, *DataToKnowledge Newsletter* (29:5), <http://www.BRCommunity.com/a2001/b086.html>, acessado em 12/11/2008.

ROSS, R. G., 2003, *Principles of the Business Rule Approach*, Boston, MA, USA, Addison-Wesley Information Technology.

ROZINAT, A., MANS, R. S., SONG, M., *et al.*, 2009, “Discovering simulation models”, *Information Systems*, vol. 34, n.3, pp. 305-327.

ROZINAT, A., AALST, W. M. P. van der, 2006, “Decision Mining in ProM”, In: Dustdar *et al.* (Eds.), *Lecture Notes in Computer Science: Business Process Management (BPM 2006)*, vol. 4102, pp. 420-425, Berlin, Alemanha.

RUBIANES, C. J., 1985, *Manual de derecho processal penal*, 6ª edição, Buenos Aires, Depalma.

RULEML, 2009, *The Rule Markup Initiative*, <http://ruleml.org/>, acessado em 23/03/2009.

SIQUEIRA, J. R. B., 2002, *Source Inspector - Uma Ferramenta para Extração de Regras de Negócio em Sistemas Legados*, M.Sc., Universidade Federal do Rio de Janeiro, Rio de Janeiro.

SULAIMAN S. J. A., 2002, *Mineração de dados endógenos*, D.Sc., Universidade Federal do Rio de Janeiro, Rio de Janeiro.

TAN, P., STEINBACH, M., KUMAR, V., 2006, *Introducion to Data Mining*, 1ª edição, Pearson Addison-Wesley, 2006.

WANG, C., ZHOU, Y., CHEN, J., 2008, “Extracting Prime Business Rules from large legacy system”, In: *CSSE Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, vol. 2, pp. 19-23.

WEIDEN, M., HERMANS, L., SCHREIBER, G., *et al.*, 2002, “Classification and Representation of Business Rules”, *European Business Rules Conference*, Junho, 2002.

WEKA, 2009, *WEKA Home Page*, <http://www.cs.waikato.ac.nz/ml/weka/>, acessado em janeiro de 2009.

WITTEN, I. H., FRANK, E., 2000, *Data Mining: practical machine learning tools and techniques*, 1ª edição, Morgan Kaufmann Publishers, San Francisco, CA, USA, 2000.

ZIMBRÃO, G., MARANDA, R. A., SOUZA, J. M., *et al.*, 2002, “FalaOCL: Uma ferramenta para Parafrapear OCL”, In: *Anais do Simpósio Brasileiro de Engenharia de Software*, vol. 16, pp. 390-395, Rio Grande do Sul, Brasil, outubro, 2002.

Apêndice A – Geração de dados com a ferramenta CPN Tools

A ferramenta *CPN Tools* (CPNTOOLS, 2007), é uma ferramenta onde se pode criar e executar simulação de workflows representados nas chamadas *CP Nets* (*Coloured Petri Nets*) (CPNETS, 2002), que é uma linguagem gráfica para projeto, especificação, simulação e verificação de sistemas. *CPN Tools* foi adotada como importante ferramenta na pesquisa de mineração de processos ao ser utilizada para gerar log de simulação que pode ser convertido para o formato Mxml pela ferramenta *ProM Import* (MEDEIROS & GÜNTHER, 2005).

Através dessa ferramenta é possível forçar a ocorrência de determinados comportamentos e gerar um log controlado que pode ser utilizado para verificar se as regras embutidas no comportamento forçado na simulação são refletidas na mineração realizada sobre o log de eventos. Abaixo estão alguns exemplos de como funções podem ser geradas em um modelo para a realização de simulação:

```
fun IS_OK_CREDIT(client_type, payment_type) =  
  if (client_type = "Old_Client") then "APPROVED"  
  else  
    if payment_type = "money" then "APPROVED"  
    else "DISAPPROVED";
```

O objetivo desta formula acima é forçar que em todas as execuções, somente os clientes antigos (`client_type = "Old_Client"`) ou clientes novos que efetuam pagamento com dinheiro (`payment_type = "money"`) passam a ter o atributo de contexto “Credit_Status” com valor “APPROVED”. Já os atributos de contexto “Payment_Status” e “Delivery_released” possuem seus valores gerados de acordo com as seguinte definições:

```
fun GET_PAYMENT_STATUS(paymentType, paymentStatus, randomStatus) =  
  if payment_type = "money" then (  
    paymentStatus = "OK";)  
  else (  
    paymentStatus = randomPaymentStatus;);
```

```
fun IS_DELIVERY_RELEASED(paymentStatus) =  
  if paymentStatus = "OK" then "YES"  
  else "NO";
```

A primeira fórmula faz com que “Payment_Status” tenha sempre valor “OK” quando o “Payment_Type” for “money” e valor aleatório (dentro do escopo de valores estipulado) quando “Payment_type” possuir outro valor. E a última fórmula apresentada faz com que sempre que “Payment_Status” assumir valor igual a “OK”, “Delivery_Release” assume o valor “YES”, em outros casos este atributo assume o valor de “NO”.