UNIVERSIDADE METODISTA DE PIRACICABA

FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA

MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

UMA ONTOLOGIA PARA INSERIR CONHECIMENTO HUMANO EM FERRAMENTAS DE MINERAÇÃO DE DADOS

EDMAR AUGUSTO YOKOME

ORIENTADORA: PROFA. DRA. FLÁVIA LINHALIS ARANTES

PIRACICABA, SP
2011

UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

UMA ONTOLOGIA PARA INSERIR CONHECIMENTO HUMANO EM FERRAMENTAS DE MINERAÇÃO DE DADOS

EDMAR AUGUSTO YOKOME

ORIENTADORA: PROFA. DRA. FLÁVIA LINHALIS ARANTES

Dissertação apresentada ao Mestrado em Ciência da Computação, da Faculdade de Ciências Exatas e da Natureza, da Universidade Metodista de Piracicaba – UNIMEP, como parte dos requisitos para obtenção do Título de Mestre em Ciência da Computação.

PIRACICABA, SP
2011

UMA ONTOLOGIA PARA INSERIR CONHECIMENTO HUMANO EM FERRAMENTAS DE MINERAÇÃO DE DADOS

Autor: Edmar Augusto Yokome
Orientadora: Profa. Dra. Flávia Linhalis Arantes

Dissertação de Mestrado apresentada em 01 de junho de 2011, à Banca Examinadora constituída dos Professores:

_____

Profa. Dra. Flávia Linhalis Arantes
UNICAMP

_____

Prof. Dr. Ivan Rizzo Guilherme
UNESP

_____

Profa. Dra. Marina Teresa Pires Vieira
UNIMEP

DEDICATÓRIA

Gostaria de dedicar este trabalho primeiramente a Deus, a força maior, que me deu a vida, saúde e força para cursar o mestrado e me abençoou na grande quantidade de viagens que fiz ao longo desses dois anos.

Aos meus pais e irmãos que sempre estiveram presentes em todos os momentos.

À Ciência da Computação que me fez refletir o tanto que esta área é complexa e diversificada.

Aos meus amigos que de certa forma fiquei afastado neste período que me dediquei ao mestrado.

Às minhas sobrinhas (Myallyn, Mayumi e Ana Heloisa), em especial a Aninha que foi meu escape do stress.

Às Artes Marciais e aos esportes que tanto gosto, mas que por uma causa maior fiquei afastado.

Aos ex-coordenadores (Eleonilda, Rogério e Vivian) da UEG – UnU Santa Helena de Goiás no período de 2007 a 2008, no qual fiz parte dessa equipe. Trabalhamos e divertimos muito, foi um bom período que não votará mais.

AGRADECIMENTOS

Primeiramente gostaria de agradecer à professora Dr[a] Flávia Linhalis Arantes pela orientação, que aceitou ser a minha orientadora em um período complicado, me introduziu uma nova área da Ciência da Computação e mesmo depois de deixar de ser professora do programa de mestrado, continuou a me orientar, sempre me apoiando e incentivando, gostaria de registrar minha eterna gratidão.

Gostaria de agradecer ao programa de mestrado em Ciência da Computação da UNIMEP por ter sido um dos escolhidos. Aos professores (Dr Plínio, Dr[a] Marina, Dr Victor, Dr Luiz Eduardo, Dr[a] Ana Estela e Dr[a] Cecília), pelas suas aulas super proveitosas (foi de longe a melhor coisa que fiz para minha vida profissional, em especial as professoras Marina e Ana Estela que contribuiram muito com o crescimento deste trabalho ao participarem da minha banca de qualificação), aos colegas de mestrado (Rodrigo, Etianne, José Edielson, Carlos, João Paulo, Ku Hai Chiang, Márcio, Isaias e Regina) pelos momentos difíceis e alegres; às funcionárias (Rosa e Dulce) que sempre estavam à disposição para nos atender. E à aluna de iniciação cientifica (Mirela) da UNIMEP pela grande ajuda que me deu na ferramenta Kira.

Às diversas instituições de ensino por onde passei e a seus funcionários, professores e colegas, que foram responsáveis pela minha formação, entre as instituições estão: Colégio Modelo (Ensino Fundamental), Colégio Vital de Oliveira (Ginásio), Escola Paroquial de 1^0 e 2^0 Grau (Colegial), UEG (Curso Superior) e Faculdade FAR (Pós-Graduação Latu Sensu), onde também fiz grandes amizades.

Gostaria de fazer um agradecimento especial à instituição UEG - UnU de Santa Helena que, além de permitir que eu fizesse um curso superior, me deu oportunidade de trabalho. E seus funcionários, professores e alunos, em especial a professora Dilça (Coordenadora do Curso) que aceitou a ser coordenadora para que eu pudesse cursar o mestrado e sempre me apoiou e incentivou. As ex-Diretoras Ereni (em especial, por ter me chamado para fazer parte do grupo de funcionários da UEG) e Maria Lúcia e o atual Diretor prof Dr Luis Carlos pelo incentivo. E por fim ao corpo docente e discente do curso de Sistemas de Informação que entenderam minha constante ausência ao longo desses anos.

E por fim ao meu ex-orientador de especialização o prof Msc. Fabian e a profa Msc. Dulcinéia que fizeram a carta de recomendação para o mestrado.

“O valor das coisas não está no tempo em que elas duram,

mas na intensidade com que acontecem.

Por isso existem momentos inesquecíveis,

coisas inexplicáveis e pessoas incomparáveis".

(Fernando Pessoa)

Faça da pedra de tropeço, um degrau de subida. Transforme cada fato negativo, em uma experiência positiva.

Bruce Lee

Quando alguém está querendo aprender, o conselho de uma pessoa experiente vale mais do que anéis de ouro ou jóias de ouro puro. Pv 25:12

RESUMO

Ontologias vêm sendo utilizadas amplamente em pesquisas na área da Ciência da Computação, inclusive na mineração de dados. Este trabalho apresenta o desenvolvimento de uma ontologia para o domínio de mineração de dados, cujo objetivo é fornecer uma terminologia comum que pode ser compartilhada e processável por ferramentas de mineração de dados. O principal diferencial da ontologia desenvolvida é identificar pontos onde o conhecimento humano se torna necessário, onde a partir desta característica é possível utilizar metodologias orientadas ao domínio, como a D^3M, e com a utilização desta metodologia é possível obter uma mineração mais interativa entre a máquina e o minerador de dados. Como produto gerado a partir desta ontologia é proposto uma arquitetura para ferramentas de mineração de dados, levando em consideração a metodologia D^3M, onde a partir desta é possível desenvolver ferramentas de mineração de dados orientadas ao domínio.

Palavras-Chaves:
Ontologia de Domínio, Mineração de Dados, Metodologias para Mineração de Dados, Conhecimento de Domínio, Ferramentas de Mineração de Dados.

ABSTRACT

Ontologies have widely been used in Computer Science research, including data mining. This works presents the development of a domain ontology for data mining, which aims to provide a common terminology that can be shared and computed by data mining tools. The main feature of the ontology is to identify where human knowledge is required. With this feature it is possible to use domain-oriented methodologies, such as D^3M, to obtain a more interactive data mining between the machine and the data miner expert. As product generated from this ontology we proposed an architecture for data mining tools. The architecture takes into account the D^3M methodology, aiming the developement of domain-oriented data mining tools.

Key Words:
Domain Ontology, Data Mining, Data Mining Methodologies, Domain Knowledge, Data Mining Tools.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

CRISP-DM: *Cross-Industry Standard Process for Data Mining* – Processo Padrão Intersetorial para Mineração de Dados

D^3M: *Domain Driven Data Mining* - Mineração de Dados Orientada ao Domínio

DAML: *DARPA Agent Markup Language* – Linguagem de Marcação para Agentes da DARPA

DDID-PD: *Domain-driven in-depth pattern discovery* – Descoberta de Padrões em Profundidade Orientada ao Domínio

DM: *Data Mining* - Mineração de Dados

DMO: *Data Mining Ontology* – Ontologia de Mineração de Dados

KDD: *Knowledge Discovery in Databases* - Descoberta do Conhecimento em Bases de Dados

KIF: *Knowledge Interchange Format* – Formato de Troca de Conhecimento

OCML: *Operational Conceptual Modeling Language* – Linguagem de Modelagem Conceitual Operacional

ODE: *Ontology Design Environment* – Ambiente para Projeto de Ontologias

OIL: *Ontology Interchange Language* – Linguagem para Intercâmbio de Ontologias

OKBC: *Open Knowledge Based Connectivity* – Conhecimento Aberto Baseado em Conectividade

OWL: *Ontology Web Language* – Linguagem de Ontologias para Web

RDF: *Resource Descriptor Framework* – Framework para Descrição de Recursos

SWRL: *Semantic Web Rule Language* – Linguagem de Regras para Web Semântica

WEKA: *Waikato Environment For Knowledge Analysis* - Ambiente Waikato para Análise do Conhecimento

XML*: eXtensible Markup Language* - Linguagem de Marcação Extensível

1 INTRODUÇÃO

1.1 Contextualização

A computação nos últimos anos evoluiu de maneira surpreendente. Uma das consequências dessa evolução foi que as bases de dados cresceram de forma inimaginável e ficaram também mais heterogêneas (por exemplo: banco de dados multimídia, de texto, temporais e outros (HAN & KAMBER, 2006).

Com essa diversidade e aumento de volume nas bases de dados, ficou praticamente impossível fazer uma análise de forma manual nos dados. Devido a essa questão, surgiu um novo ramo na ciência da computação chamado mineração de dados (DM, do inglês *Data Mining*), cujo propósito é encontrar padrões interessantes em bases de dados.

Essa necessidade de encontrar padrões interessantes nas bases de dados é uma tentativa de descobrir algo que possa auxiliar o analista de dados em uma tomada de decisão. Por exemplo: quais produtos podem ser vendidos em conjunto em um supermercado, dado o histórico de vendas realizadas? Se o processo de mineração de dados conseguir chegar a um determinado padrão, poderá ajudar a definir uma estratégia de *marketing* para o supermercado.

A tarefa de descoberta de conhecimento em base de dados é conhecida como KDD (*Knowledge Discovery in Databases*). A mineração de dados é uma das etapas que faz parte do processo de KDD. Esse processo consiste de uma série de passos de transformação, pré-processamento e pós-processamento dos resultados da mineração de dados (TAN et al., 2009).

Na tentativa de disciplinar o processo de KDD, surgiram algumas metodologias, dentre elas a CRISP-DM (*Cross-Industry Standard Process For Data Mining*). De acordo com essa metodologia, o ciclo de vida de um projeto de mineração de dados é dividido em seis fases: entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e desenvolvimento (CHAPMAN et al., 2000). CRISP-DM é uma metodologia bem ampla e detalhada, e é usada para fornecer orientações em relação a como as várias fases de um projeto de mineração de dados podem ser executadas (SHARMA & OSEI-BRYSON, 2008).

Ao seguir CRISP-DM ou outras metodologias orientadas aos dados, é constatado que o processo de KDD é realizado de forma sistematizada. Isto é, o usuário escolhe, prepara e entra com os dados, adota um algoritmo de mineração de dados e define como será a visualização dos padrões gerados, porém não há uma interferência durante o processo de aplicação das técnicas de mineração de dados, onde são feitas somente nas fases anteriores e posteriores a esta aplicação (CAO & ZHANG, 2006).

Metodologias orientadas aos dados como CRISP-DM tratam o processo de mineração como algo isolado e baseado na tentativa e erro. Nessas metodologias, questões relacionadas ao objetivo do negócio e a aplicação de técnicas de mineração de dados têm pouco apoio durante seu processo, sendo tratados de maneira isolada. Como resultado, o conhecimento descoberto com a mineração pode não ser interessante para o negócio (ou problema) em questão.

Metodologias orientadas ao domínio do negócio, como D^3M, vem, aos poucos, sendo utilizadas na mineração de dados. O objetivo da metodologia D^3M é diminuir a distância entre a mineração realizada na academia e na indústria, onde são retratadas partes do mundo real. Na academia são elaborados estudos sistemáticos com o objetivo de aumentar a eficiência dos resultados produzidos com a mineração de dados e na indústria é necessário que os resultados gerados forneçam resultados satisfatórios que possam ser aplicados (CAO & ZHANG, 2006). Esta metodologia se baseia em elementos chaves como restrições de contexto; integração do conhecimento do domínio do negócio; cooperação entre humanos e máquinas durante o processo de mineração; e refinamento iterativo dos resultados.

Para concretizar um projeto de mineração de dados há várias ferramentas, entre elas estão: Kira (Mendes, 2009), WEKA[1], *Bramining*[2], dentre outras. Estas ferramentas permitem fazer uma análise de forma automática, algo que se fosse feito manualmente tornaria um projeto de mineração de dados inviável. Porém, por mais que estas ferramentas sejam eficientes, o conhecimento humano é exigido em várias fases do processo de mineração de dados.

Para possibilitar a utilização de conhecimento humano ou conhecimento de domínio, é preciso representar formalmente a terminologia do domínio. O conceito de ontologias se adequa bem a esse propósito.

[1] www.cs.waikato.ac.nz. Acesso: 09 jun. 2011.
[2] www.graal-corp.com.br. Acesso: 09 jun. 2011.

Uma ontologia é comumente utilizada como uma estrutura que representa o conhecimento sobre uma determinada área (ou domínio) por meio de conceitos relevantes e relacionamentos entre eles (FALBO et al., 1998). Em outras palavras, as ontologias podem representar a semântica necessária para descrever determinado domínio de aplicação. Portanto, as ontologias podem ser peças fundamentais para viabilizar metodologias de mineração de dados orientadas ao domínio, como D^3M, e inserir conhecimento humano durante o processo de mineração de dados.

1.2 Motivação

Apesar da existência de metodologias e ferramentas para auxiliar na tarefa de descoberta de informação, esse processo não é simples, pois há várias etapas a serem seguidas, e muitas das vezes são encontradas barreiras difíceis de serem solucionadas, como, por exemplo, a melhor forma de fazer a limpeza de dados, que tarefa de mineração de dados utilizar, como analisar os resultados produzidos, entre outros. Essas barreiras se tornam ainda maiores e mais evidentes diante da necessidade de se inserir conhecimento de domínio no processo de mineração.

A motivação deste trabalho é aliar ontologias ao processo de mineração de dados para guiar o processo de mineração de dados, onde é considerado as fases do KDD e também a semântica do domínio do problema. Com isso, pretende-se ajudar a inserir conhecimento humano no processo de mineração de dados realizado por ferramentas de mineração.

1.3 Objetivos

O objetivo deste trabalho é criar uma ontologia para o domínio de mineração de dados que irá guiar o minerador de dados durante o processo de KDD em ferramentas de mineração de dados. A ontologia deverá servir de base para uma arquitetura para ferramentas de mineração de dados onde é levado em consideração a metodologia D^3M.

Os resultados e contribuições esperadas com o desenvolvimento desse trabalho são:

- Desenvolver uma ontologia de domínio para mineração de dados, chamada daqui por diante de Meta-DM, cujo objetivo é guiar o processo de descoberta do conhecimento nas diversas etapas desse processo;
- Identificar as etapas onde o conhecimento humano se torna necessário e inserir tarefas da metodologia D³M, cujo objetivo é uma mineração de dados mais interativa; e
- Propor uma arquitetura para ferramentas de mineração de dados, com base na ontologia desenvolvida e na metodologia D^3M.

A arquitetura baseada em ontologias e na metodologia D^3M vem contribuir com o estado da arte na área de semântica em mineração de dados no sentido de inserir conhecimento humano e de domínio durante o processo de mineração de dados realizado em ferramentas de mineração.

1.4 Metodologia

Para o desenvolvimento deste trabalho foram realizadas as seguintes tarefas:

- Estudo das principais etapas do processo de KDD e da metodologia CRISP-DM;
- Realização de mineração de dados em ferramentas desse domínio;
- Classificação das tarefas essenciais na mineração de dados que devem fazer parte da ontologia Meta-DM;
- Utilização de metodologias para construção de ontologias, como Noy e McGuiness e METHONTOLOGY para o desenvolvimento da ontologia;
- Implementação da ontologia proposta em uma ferramenta de desenvolvimento de ontologias – Protégé;

- Verificação e avaliação da ontologia por meio da máquina de inferência Pellet e instanciação da ontologia;
- Identificação dos pontos onde o conhecimento humano se torna necessário na ontologia;
- Estudo e identificação das tarefas da metodologia D³M;
- Proposta de uma arquitetura para ferramentas de mineração de dados, baseado na ontologia Meta-DM e na metodologia D³M.

1.5 Trabalhos Relacionados

Para o desenvolvimento da ontologia Meta-DM foram pesquisados outros trabalhos existentes na literatura, que tratam do desenvolvimento de ontologias para o domínio da mineração de dados. Uma das ontologias encontradas foi parcialmente utilizada na elaboração da ontologia proposta. A seguir é apresentada uma breve descrição desses trabalhos e também é feito um comparativo com a proposta da ontologia Meta-DM.

Com a ontologia de Sharma e Osei-Bryson (2008) é representada a fase de entendimento do négócio, uma das etapas da metodologia CRISP-DM (Chapman et al., 2000). De acordo com análise feita nesta ontologia, os autores fizeram o levantamento de questões relacionadas ao entendimento do negócio cujo objetivo é ajudar o minerador no entendimento do negócio. Diferente da ontologia proposta por Sharma e Osei-Bryson (2008), a Meta-DM tem como objetivo representar todas as etapas de um projeto de mineração de dados. A Figura 1 apresenta o diagrama da ontologia de Sharma e Osei-Bryson (2008).

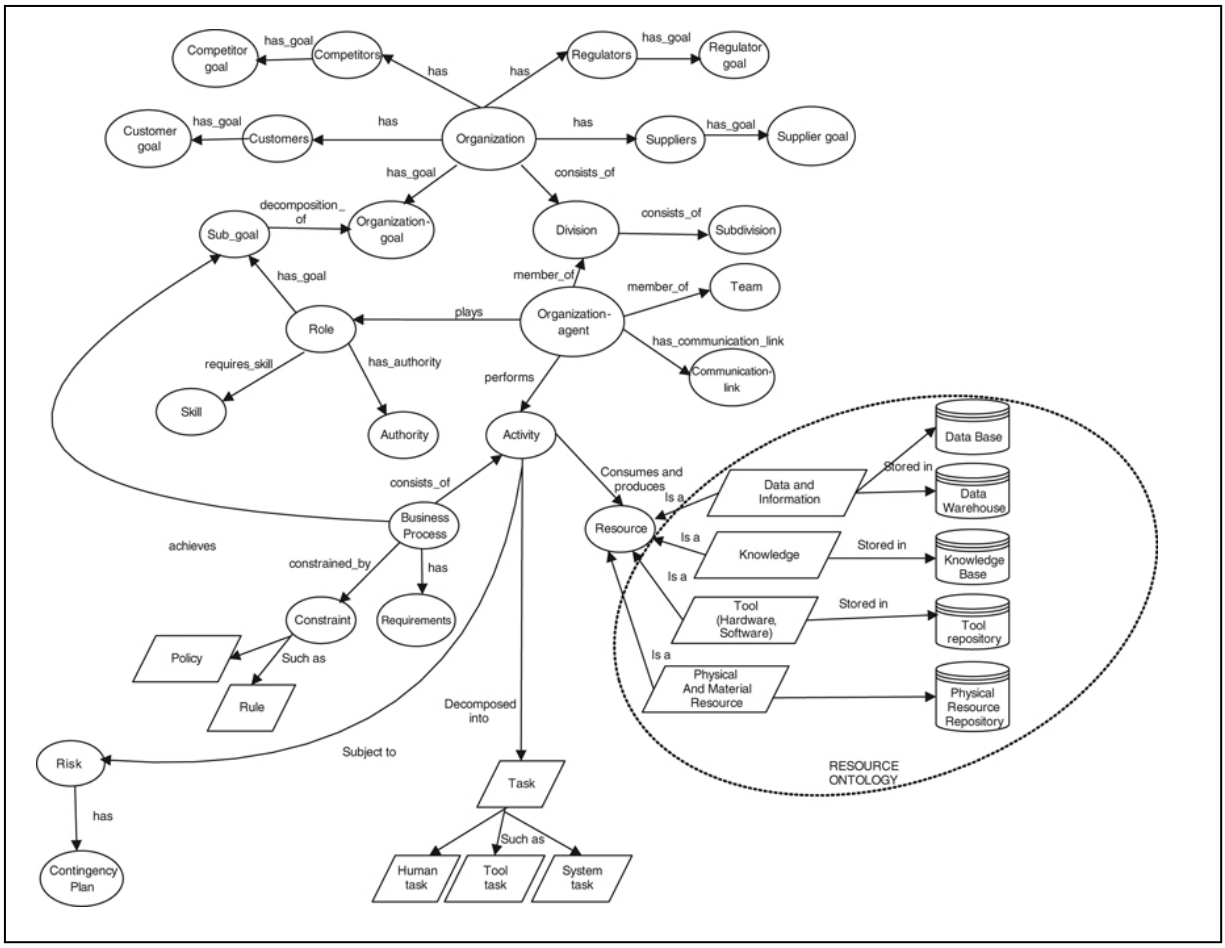


Figura 1: ONTOLOGIA DE SHARMA E OSEI-BRYSON
Fonte: Sharma e Osei-Bryson (2008)

Sharma e Osei-Bryson (2008) identificam que a fase de entendimento do negócio é composta por 67 fases, a Figura 1 apresenta uma visão parcial da ontologia de Sharma e Osei-Bryson. Esta é uma ontologia interessante para restringir o contexto do domínio do problema. Ela é novamente mencionada no Capítulo 5, onde há mais discussão sobre o assunto.

A *DM Ontology*, desenvolvida por Zheng e Li (2008), é uma ontologia que contempla a mineração de dados aplicada ao negócio de *marketing* para uma empresa de financiamento, isto é, essa ontologia foi desenvolvida para um problema específico de mineração de dados. O propósito da ontologia Meta-DM é contemplar as etapas da mineração de dados como um todo, sem se voltar a um problema de mineração em particular. Entretanto, outras ontologias podem ser utilizadas juntamente com a Meta-DM para prover conhecimento de domínio necessário para resolver determinado problema de mineração de dados.

A ontologia DMO proposta por Brezany, Janciak e Tjoa (2008) foi desenvolvida com o intuito de guiar um projeto de mineração de dados em grade. Seu maior objetivo é realizar a mineração de dados utilizando serviços semânticos espalhados pela web. Para isso, utiliza a ontologia OWL-S para descrever serviços web semânticos. A ontologia utiliza os conceitos das fases do KDD, entretanto tem objetivos diferentes da ontologia Meta-DM. A ontologia DMO descreve serviços semânticos com OWL-S, enquanto a ontologia Meta-DM tem o objetivo de guiar o processo de mineração de dados em uma ferramenta de mineração, sem considerar a existência de serviços.

A ontologia OntoDM de Panov et al. (2008) tem como objetivo criar um conjunto de definições de termos para o domínio de mineração de dados, como, por exemplo, tipo de dados, conjunto de dados, tarefas de mineração de dados, algoritmos de mineração de dados, dentre outros. Desse modo, projetos de desenvolvimento de ontologias para esse domínio podem utilizar suas definições, onde é evitado ambiguidades na interpretação de alguma definição do domínio. O objetivo de servir como terminologia comum para o domínio de mineração de dados é também contemplado na Meta-DM, entretanto, a OntoDM não representa, formalmente, a necessidade de conhecimento humano no processo de KDD, característica essencial para a utilização da metodologia D^3M, abordada neste trabalho.

A ontologia de Pinto e Santos (2009) utiliza alguns conceitos da ontologia DMO de Brezany, Janciak e Tjoa (2008) e foi desenvolvida com o intuito de contemplar exclusivamente as fases do KDD seguindo a metodologia METHONTOLOGY de Fernandez-Lopez et al. (1997). O trabalho de Pinto e Santos foi o que mais se assemelhou com a ontologia proposta neste trabalho, tanto que alguns conceitos dessa ontologia foram utilizados na Meta-DM. Entretanto, a ontologia Meta-DM leva também em consideração a metodologia CRISP-DM (Chapman et al., 2000) e tem o objetivo de ser uma ontologia para ferramentas de mineração de dados, onde são identificados os momentos onde o conhecimento humano se faz necessário, com o intuito utilizar metodologias orientadas ao domínio como a D^3M, com o objetivo de buscar resultados mais significativos em um projeto de mineração de dados. A Figura 2 apresenta uma ilustração parcial da ontologia de Pinto e Santos (2009).

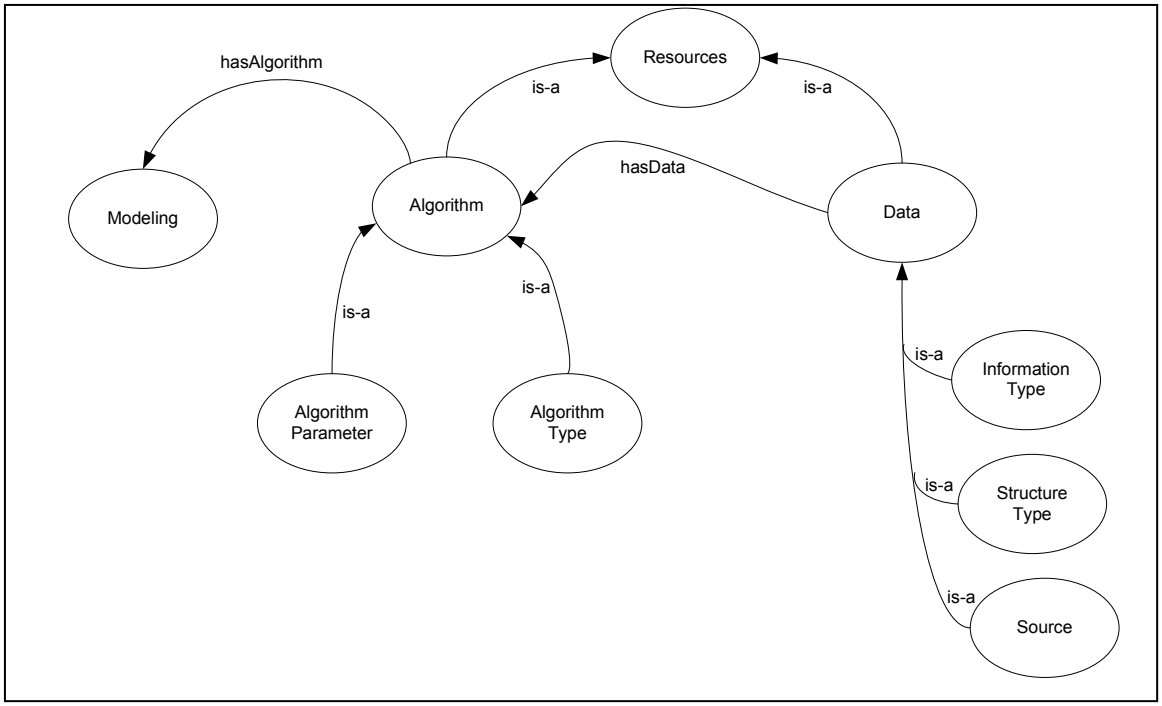


Figura 2: PARTE DA ONTOLOGIA DE PINTO E SANTOS
Fonte: Pinto e Santos

1.6 Organização do Trabalho

Esta monografia está estruturada da seguinte forma:

- O Capítulo 2 apresenta uma revisão bibliográfica sobre descoberta de conhecimento em bases de dados (KDD do inglês *Knowledge Discovery in Databases*), algumas metodologias para o desenvolvimento de um projeto de mineração de dados e algumas ferramentas para auxiliar esse processo.
- O Capítulo 3 faz um estudo bibliográfico do tema ontologias, onde são descritos alguns pontos que foram essenciais para desenvolver este trabalho.
- O Capítulo 4 é sobre o desenvolvimento da ontologia Meta-DM, onde é feito uma descrição detalhada do ciclo de vida utilizado para o desenvolvimento da ontologia;
- O Capítulo 5 faz a descrição da proposta de uma arquitetura para ferramentas de mineração de dados, onde é levado em consideração a ontologia Meta-DM e as tarefas da metodologia

D³M. Neste capítulo é apresentado ainda um cenário de uso da arquitetura, que procura ilustrar sua utilização para solucionar um problema de mineração de dados;

- O Capítulo 6 apresenta as conclusões, contribuições e trabalhos futuros.

2 MINERAÇÃO DE DADOS

2.1 Considerações Iniciais

A mineração de dados é uma área da ciência da computação que tem como objetivo encontrar padrões interessantes em bases de dados. O surgimento dessa área é justificado pela necessidade de encontrar informações úteis em bases de dados de forma eficiente, onde são utilizados em conjunto técnicas, métodos e ferramentas desse domínio.

Este capítulo tem o intuito de abordar alguns conceitos desse domínio que são considerados importantes para o desenvolvimento do tema proposto neste trabalho. Assim, no decorrer deste capítulo são abordados os seguintes assuntos:

- O processo de descoberta do conhecimento em base de dados, onde são expostos: características, fases, tarefas, técnicas e algoritmos de mineração de dados;
- Metodologias para elaboração de projetos de mineração de dados, em especial as metodologias CRISP-DM e D^3M, que são as mais importantes para o desenvolvimento deste trabalho; e
- Ferramentas utilizadas na mineração de dados.

2.2 Descoberta de Conhecimento em Base de Dados

A humanidade presencia a era da informação, ou seja, quem detém o conhecimento tem uma ampla vantagem em relação aos outros. Para poder buscar cada vez mais eficiência em suas atividades, as empresas e instituições de vários setores ao longo do tempo (desde quando a informática conquistou espaço nas diversas áreas do conhecimento) têm investido em sistemas de informação para poder otimizar e agilizar as atividades rotineiras. Com o desenvolvimento de sistemas de informação cada vez mais complexos, o barateamento no

armazenamento de dados e computadores cada vez mais poderosos, foi gerado grandes bases de dados, e a cada dia crescem ainda mais.

A Ciência da Computação está em constante busca para empregar recursos computacionais nas mais diferentes áreas, do modo mais rápido e confiável possível. Ao constatar o crescimento acentuado nas bases de dados e a necessidade de analisar essas bases, foi desenvolvida uma ramificação da computação chamada mineração de dados (no final da década de 80), cuja idéia principal seria descobrir padrões interessantes nas bases de dados. Han e Kamber definem mineração de dados como: “extração ou mineração do conhecimento em um grande amontoado de dados”. (HAN & KAMBER, 2006, p. 05, tradução nossa).

Assim, a mineração de dados foi desenvolvida com a finalidade de analisar e retirar padrões interessantes a partir de uma base de dados, pois para o ser humano seria praticamente inviável fazer uma análise manualmente.

A Figura 3 ilustra as etapas do processo de descoberta de conhecimento em bases de dados (KDD do inglês *Knowledge Discovery in Databases*).

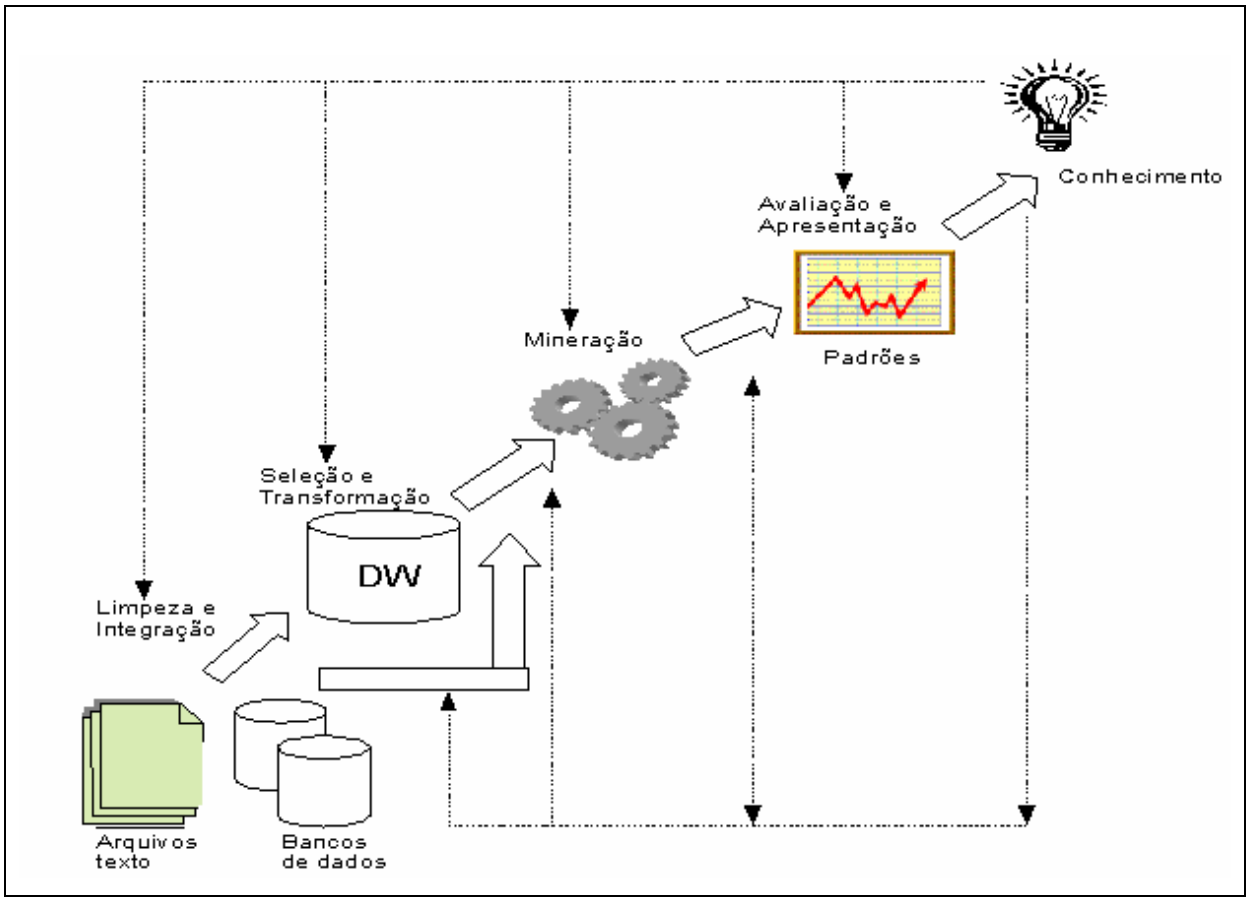


Figura 3: PROCESSO DE DESCOBERTA DO CONHECIMENTO (KDD)
Fonte: Han & Kamber (2006, pg: 6, tradução nossa)

O Quadro 1 descreve brevemente cada uma das fases do processo KDD segundo Han & Kamber (2006, v.2 p.7, tradução nossa):

Quadro 1: DESCRIÇÃO DAS FASES DO PROCESSO DE KDD (HAN & KAMBER, 2006)

Fase	Descrição da fase
Limpeza de Dados	Remover ruídos e inconsistências dos dados.
Integração dos dados	Combinação de múltiplas fontes de dados.
Seleção de dados	Dados relevantes para a tarefa de análise são selecionados da base de dados.
Transformação de dados	Dados são transformados ou consolidados dentro das formas apropriadas para mineração.
Mineração de dados	Aplicações de métodos inteligentes são utilizadas para extrair padrões dos dados.
Padrão de avaliação	Identificar os padrões verdadeiramente interessantes.
Apresentação do conhecimento	Técnicas de visualização e representação são usadas para apresentar o conhecimento minerado para o usuário.

As etapas citadas consistem em trabalhar a base de dados de modo que possa ser empregado o processo de KDD. Em resumo, as atividades a serem seguidas são: obter uma base de dados, preparar os dados, aplicar a MD e por fim mostrar e avaliar os padrões encontrados.

A mineração de dados é um campo interdisciplinar que envolve várias ciências do conhecimento, entre elas estão: tecnologia de banco de dados, aprendizagem de máquina, estatística, ciência da informação, entre outras. Dessa forma, para conseguir atingir um determinado objetivo é essencial que a equipe envolvida na tarefa de encontrar padrões interessantes a partir da base de dados tenha diferentes habilidades; além de ser essencial o trabalho em equipe.

Entre as aplicações que podem se beneficiar com a realização da mineração de dados estão:

- Planejar instalação de novas filiais de empresa/lojas;
- Realizar promoções;
- Combinar itens de propaganda;

- Planejar estratégias de periódicos de campanhas de marketing.

2.2.1 Arquitetura Típica de um Sistema de MD

Uma arquitetura típica de um sistema de MD é apresentada na Figura 4, segundo Han & Kamber (2006):

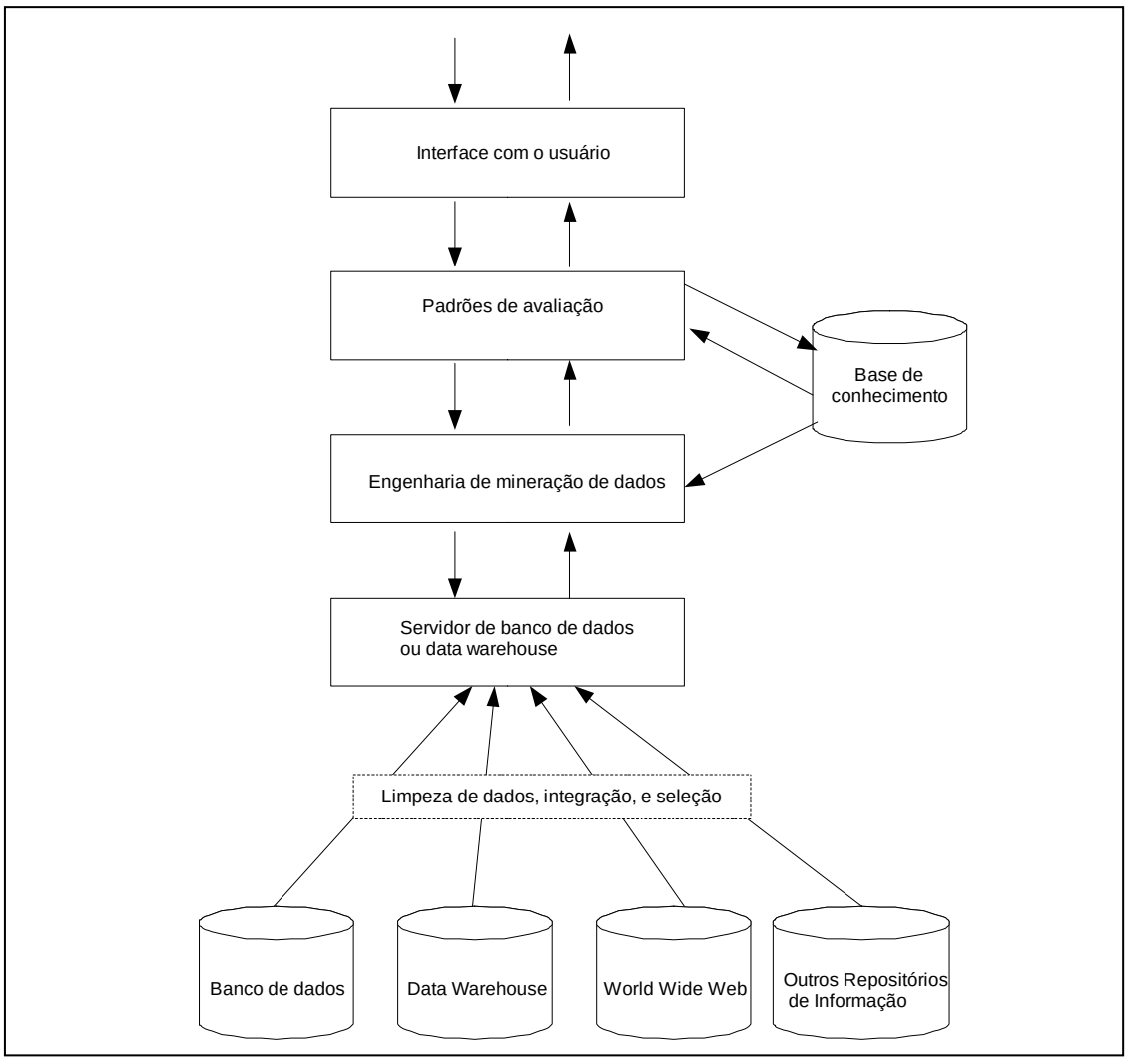


Figura 4: ARQUITETURA TÍPICA DE UM SISTEMA DE MD
Fonte: Han & Kamber (2006, pg: 08, tradução nossa)

A descrição para essa arquitetura é feita da seguinte forma, segundo Han e Kamber (2006):

- Banco de Dados, Data *Warehouse*, Word Wide Web e outros repositórios de informação: É um ou um conjunto de banco de dados, data *warehouse*, planilhas, ou outros repositórios de

informação. A limpeza de dados ou técnicas de integração de dados pode ser executada sobre esses dados;

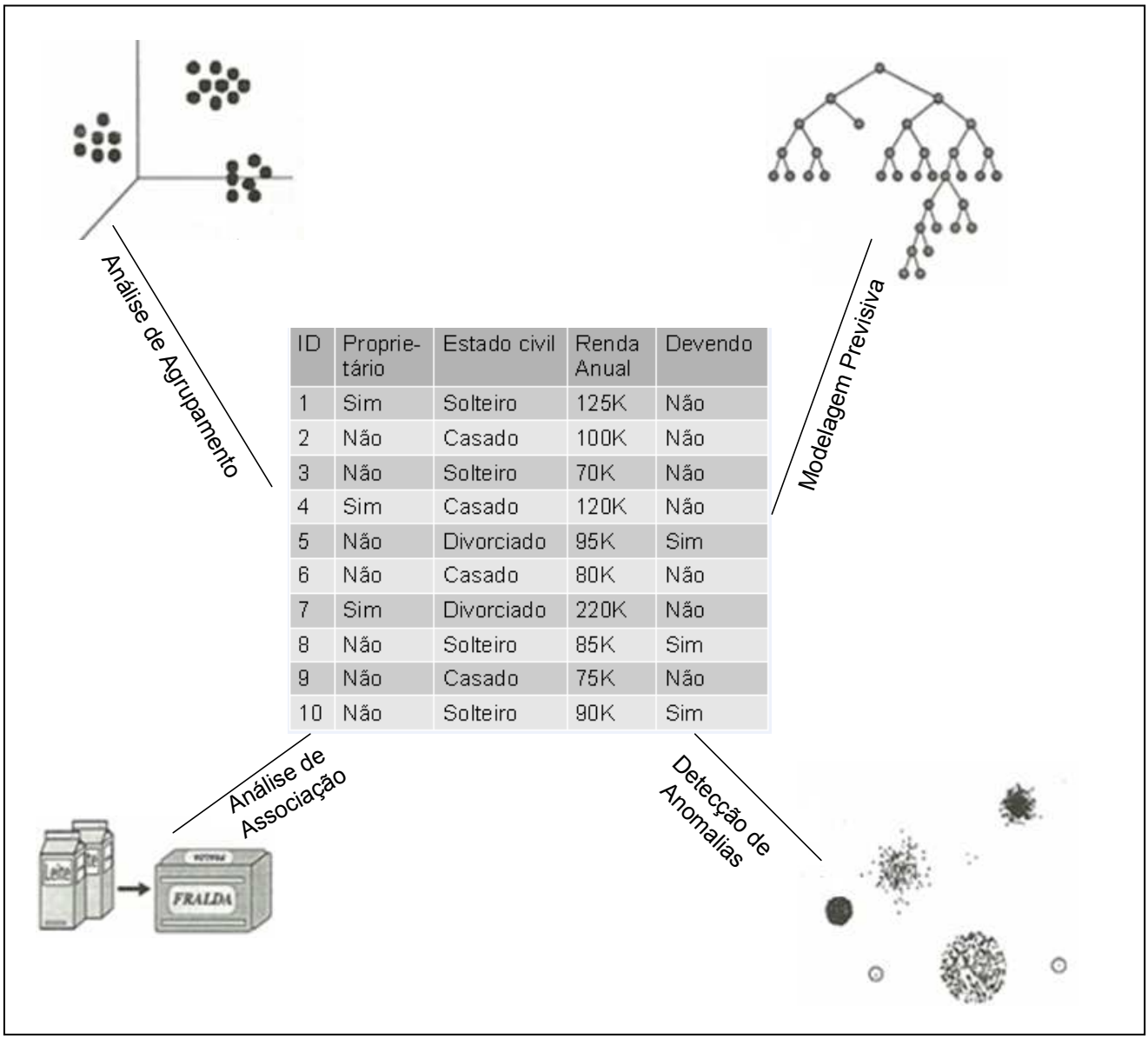
- Servidor de banco de dados ou *data warehouse*: É responsável por gerenciar os dados, com base na solicitação do usuário de mineração de dados;
- Base de Conhecimento: Este é o domínio do conhecimento que é usado para guiar as pesquisas ou avaliar os interesses dos padrões resultantes. Tais conhecimentos podem incluir conceitos hierárquicos, usados para organizar atributos ou valores dos atributos dentro de diferentes níveis de abstração;
- Engenharia de mineração de dados: É um conjunto de módulos funcionais para tarefas tais como caracterização, associação e análises de correlação, classificação, predição, análise de cluster, análise de *outilier*[3] e análise de evolução;
- Módulo de avaliação de padrões: Este módulo emprega medidas de interesse e interação com os módulos de mineração de dados;
- Interface com o usuário: Este módulo faz a comunicação entre usuários e a execução do processo de mineração de dados, onde é permitido ao usuário interagir com este processo, e especificar uma consulta de mineração de dados ou tarefa.

Ao realizar uma tarefa nesta arquitetura típica de um sistema de mineração de dados procura-se empregar as técnicas de mineração de dados e assim poder encontrar informações valiosas para uma determinada tarefa, conforme a necessidade a ser suprida.

2.2.2 Tarefas da Mineração de Dados

A Figura 5 apresenta as tarefas que são realizadas na mineração de dados:

[3] Anomalias



ID	Proprie-tário	Estado civil	Renda Anual	Devendo
1	Sim	Solteiro	125K	Não
2	Não	Casado	100K	Não
3	Não	Solteiro	70K	Não
4	Sim	Casado	120K	Não
5	Não	Divorciado	95K	Sim
6	Não	Casado	80K	Não
7	Sim	Divorciado	220K	Não
8	Não	Solteiro	85K	Sim
9	Não	Casado	75K	Não
10	Não	Solteiro	90K	Sim

Figura 5: AS TAREFAS CENTRAIS DA MD
Fonte: Tan et al. (2009, pág: 9)

Conforme apresentado na Figura 5 as tarefas de mineração de dados são: análise previsiva, análise de agrupamento, análise de associação e detecção de anomalias.

O Quadro 2 faz uma breve descrição para cada uma dessas tarefas segundo Tan et al. (2009):

Quadro 2: DESCRIÇÃO DAS TAREFAS DE MINERAÇÃO DE DADOS, (Tan et al. 2009)

Tarefa de Mineração de Dados	Descrição
Modelagem Previsiva	Refere-se à tarefa de construir um modelo para a variável alvo como uma função das variáveis explicativas. Há dois tipos de tarefas de modelagem de previsão: classificação, a qual é usada para variáveis discretas, e regressão, que é usada para variáveis alvo contínuas. Exemplo: na Figura 6 é apresentada uma árvore de decisão onde dada uma informação no banco de dados, estes dados poderiam ser classificados até chegar a uma informação desejada.
Análise de Associação	É usada para descobrir padrões que descrevam características altamente associadas dentro dos dados. Exemplo: Na Figura 5 há uma associação entre {Fraldas} → {Leite}, que sugere que os clientes que compram fraldas tendem a comprar leite.
Análise de Agrupamento	Procura encontrar grupos de observações intimamente relacionadas de modo que observações que pertençam ao mesmo grupo sejam mais semelhantes entre si do que com as que pertençam a outros grupos. Exemplo: na Figura 5 compras que tenham semelhanças umas com as outras, poderiam ser agrupadas assim como as pessoas que compraram leite e fralda.
Detecção de Anomalias	É a tarefa de identificar observações cujas características sejam significativamente diferentes do resto dos dados. Exemplo: na Figura 5 dados que não pertence a nenhuma característica estabelecida seriam considerados anomalias.

Para cada tarefa de mineração de dados há um conjunto de algoritmos específicos a ser aplicado, e estes utilizam determinadas técnicas de mineração de dados que são abordados na seção 2.2.3.

2.2.3 Técnicas de Mineração de Dados

Ao utilizar-se uma tarefa de MD, algumas técnicas são usadas para concretizá-las. Entre estas técnicas, Santos e Ramos (2009) destacam as seguintes: árvores de decisão, regras de associação, regressão linear, redes neurais, algoritmos genéticos e vizinhos mais próximos. A seguir é feita uma breve descrição de cada uma delas.

Árvores de Decisão: Santos e Ramos (2009, p. 132) definem essa técnica da seguinte maneira.

> As árvores de decisão, como o próprio nome indica, são constituídas por estruturas em árvores que representam um conjunto de decisões. Os algoritmos dessa técnica permitem gerar regras de classificação dos dados, baseados nas informações armazenadas na base de dados. A Figura 6 apresenta um exemplo dessa estrutura.

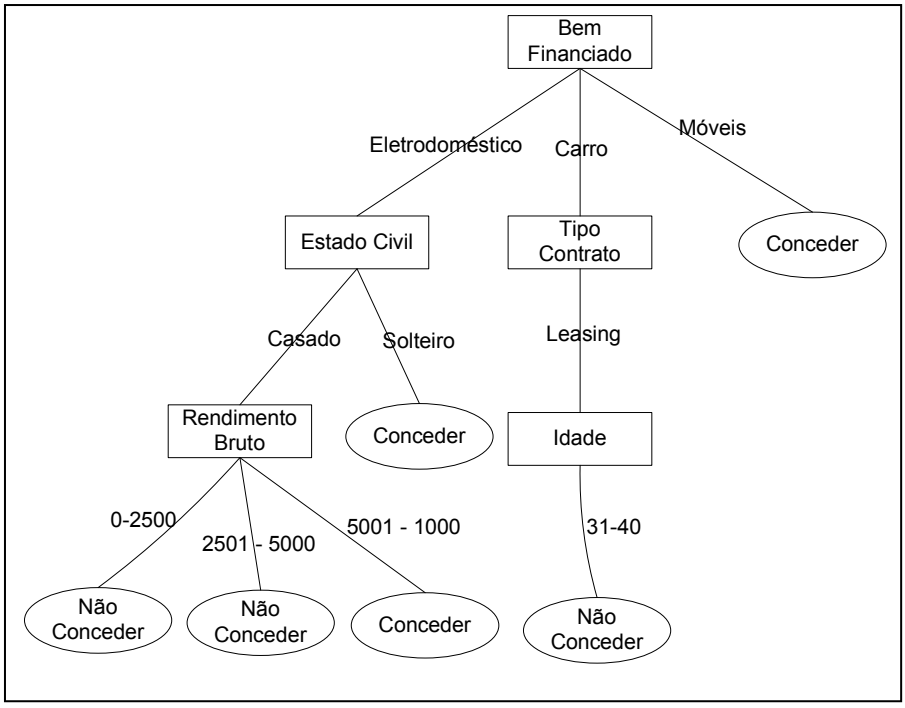


Figura 6: ÁRVORE DE DECISÃO
Fonte: Santos e Ramos

Na Figura 6 é apresentado um exemplo prático da utilização de uma árvore de decisão, cujo objetivo é conceder ou não crédito a uma determinada pessoa, onde é levado em consideração o tipo de bem a ser financiado, o estado civil, o tipo de contrato, o rendimento bruto e a idade.

Santos e Ramos (2009) abordam que as árvores de decisão podem ainda ser representadas por um conjunto de regras, como apresentado a seguir:

```
Se Bem Financiado = “Eletrodoméstico” e Estado Civil = “Casado” e
Rendimento Bruto = “5001 – 10000” Então “Conceder”.
```

Santos e Ramos (2009) destacam que cada folha da árvore dá origem a uma regra, sendo o seu conteúdo apresentado na parte consequente da regra.

Regras de Associação: Santos e Ramos (2009, p. 135) definem esta técnica da seguinte maneira.

> O objetivo dessa técnica é identificar regras que relacionam uma conclusão (por exemplo, a compra de um produto) com um conjunto de condições (por exemplo, a compra de outros produtos), permitindo encontrar relacionamentos entre os atributos existentes numa base de dados, sendo representado na forma de uma regra: Se X então y ou “X => Y”.

A Figura 7 apresenta um exemplo de uma base de dados e a utilização de regras de associação para gerar associação entre os elementos.

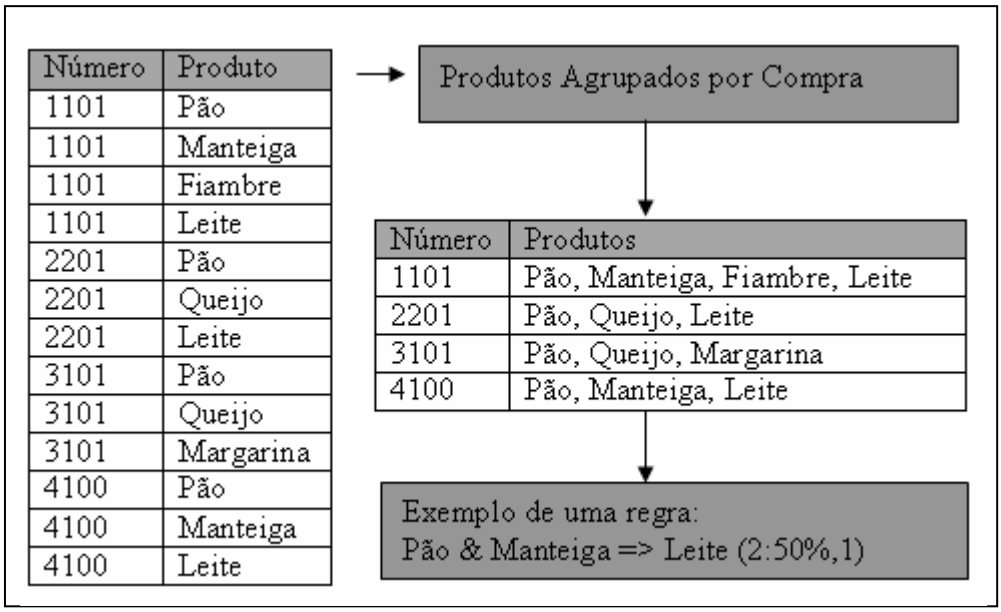


Figura 7: PROCESSO DE INDUÇÃO DE REGRAS DE ASSOCIAÇÃO
Fonte: Santos e Ramos (2009)

A Figura 7 apresenta um conjunto de dados associado à compra de produtos, estes produtos são agrupados por compra e a partir desse agrupamento são geradas regras. No exemplo foi gerada a regra: Pão & Manteiga => Leite (2:50%,1) indica que os clientes que compram o produto Pão juntamente com o produto manteiga, tende a comprar o produto Leite, esta regra apresenta um suporte de 50%, o que significa que metade dos registros analisados pertencem a referida regra.

Regressão Linear: Santos e Ramos (2009, p. 136) definem esta técnica da seguinte maneira.

> Esta técnica é utilizada sempre que se pretende prever uma variável com valores contínuos. Os dados são modelados aproximando-os a uma linha reta. A forma mais simples de regressão é apresentada através de uma equação com duas variáveis, X e Y, tal que: $Y = \alpha + \beta X$, onde X representa a variável independente, Y a variável dependente calculada a partir de X, e α e β os coeficientes da regressão.

Redes Neurais Artificiais: Santos e Ramos (2009, p. 138) definem esta técnica da seguinte forma.

> Redes neurais artificiais são sistemas classificatórios modelados segundo o funcionamento do sistema nervoso humano. Estes sistemas são compostos por um conjunto de unidades, organizadas em níveis. As unidades (nós) encontram-se conectadas através de ligações, nas quais têm associado um peso. As unidades que constituem uma rede encontram-se agrupadas em três grupos: unidades de entradas encarregadas de receber os dados (atributos) a analisar; unidade de saída que transmitem os sinais à saída da rede; e um número ilimitado de níveis intermediários (ou níveis ocultos) que contêm as unidades intermediárias.

A Figura 8 apresenta um exemplo de uma rede neural artificial.

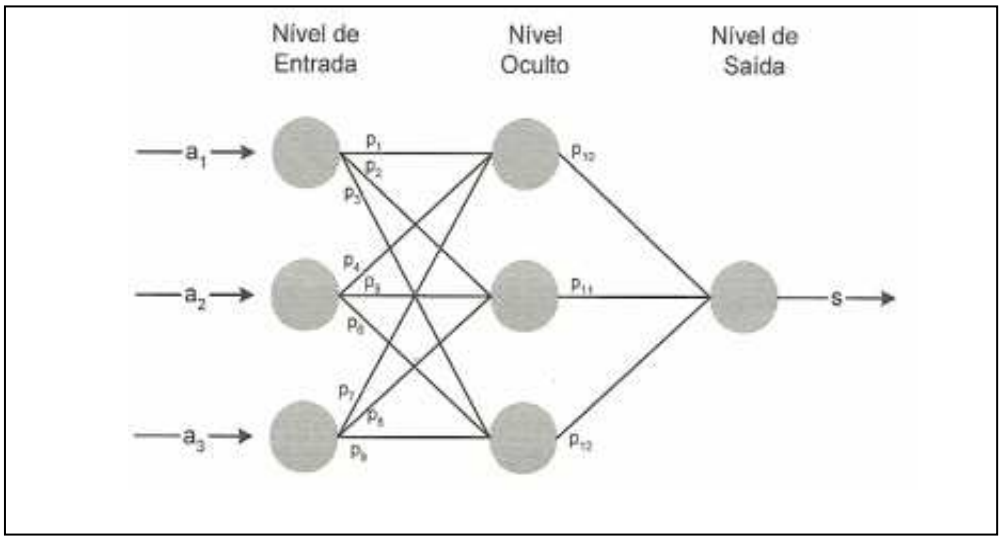


Figura 8: REDE NEURAL ARTIFICIAL
Fonte Ramos e Santos (2009)

Santos e Ramos (2009) destacam também que existem dois estágios na utilização de uma rede neural artificial. O primeiro diz respeito à aprendizagem, no qual a rede é treinada para a execução de determinada tarefa. No segundo acontece a previsão, na qual a rede é utilizada para classificar registros desconhecidos. Estes conceitos também são aplicados em árvores de decisão.

Algoritmos Genéticos: Segundo Santos e Ramos (2009) a técnica de algoritmo genético utiliza princípios da biologia e da ciência da computação. Esta técnica inicia com um conjunto de regras, as quais são submetidas a operadores de seleção e reprodução, de forma a desenvolverem regras mais apuradas, e então são etiquetadas com determinado valor de utilidade, que facilita a seleção das mesmas. A Figura 9 apresenta um modelo de algoritmo genético.

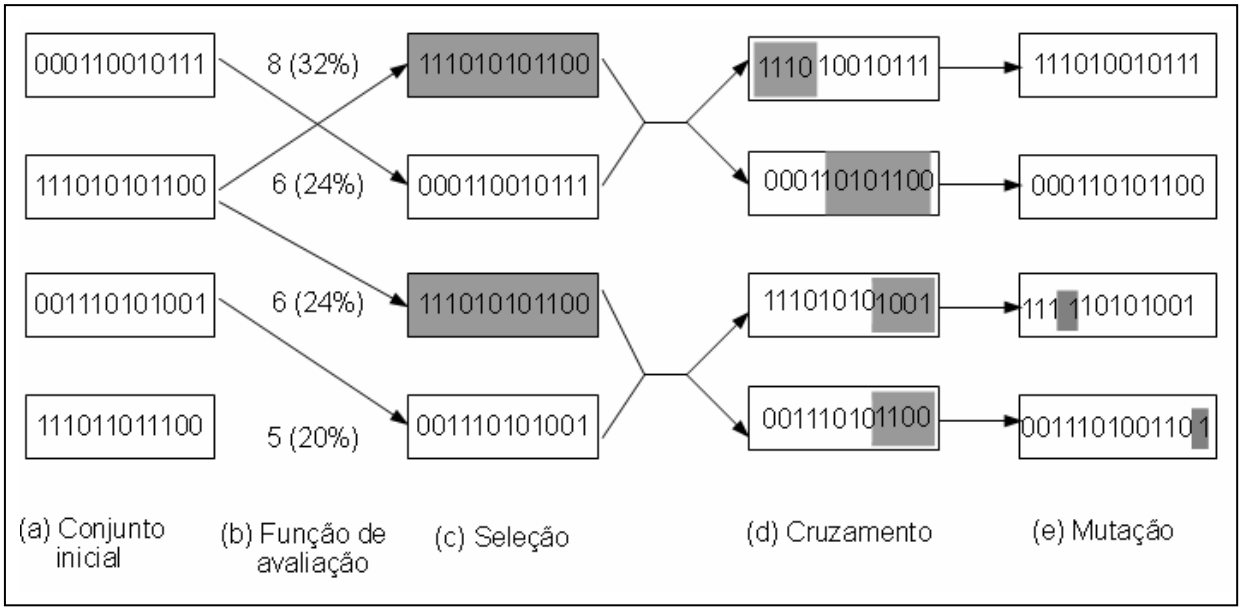


Figura 9: MODO DE OPERAÇÃO DOS ALGORITMOS GENÉTICOS
Fonte: Santos e Ramos (2009)

Santos e Ramos (2009, p. 142) fazem a seguinte descrição para o funcionamento dessa técnica.

> A Figura 9 (a),representa a amostra inicial dos dados, um conjunto com quatros regras. Para cada uma das regras é identificado o valor de utilidade, através da função de avaliação (Figura 9 (b)). Por exemplo, a primeira regra foi classificada com 8, o que significa que apresenta uma probabilidade de 32% de ser selecionada. Na Figura 9 (c), é apresentada a identificação dos pares e a definição do ponto de cruzamento, o que conduz à construção de novas regras, apresentadas na Figura 9 (d). O último passo está associado à mutação (Figura 9 (e)), no qual é verificada a mutação aleatória de dois caracteres.

Vizinhos mais Próximos: Santos e Ramos (2009, p. 143) definem esta técnica da seguinte maneira.

> A segmentação de dados através dos vizinhos mais próximos integra um conjunto de técnicas que se baseiam no princípio de que registros semelhantes estão próximos uns dos outros, quando analisados numa perspectiva espacial. A verificação da localização dos registros, interpretados como pontos no espaço, permitem a identificação de regiões denominadas classes (ou segmentos), que apresentam características comuns para os registros que representam.

A Figura 10 exemplifica a funcionalidade dessa técnica.

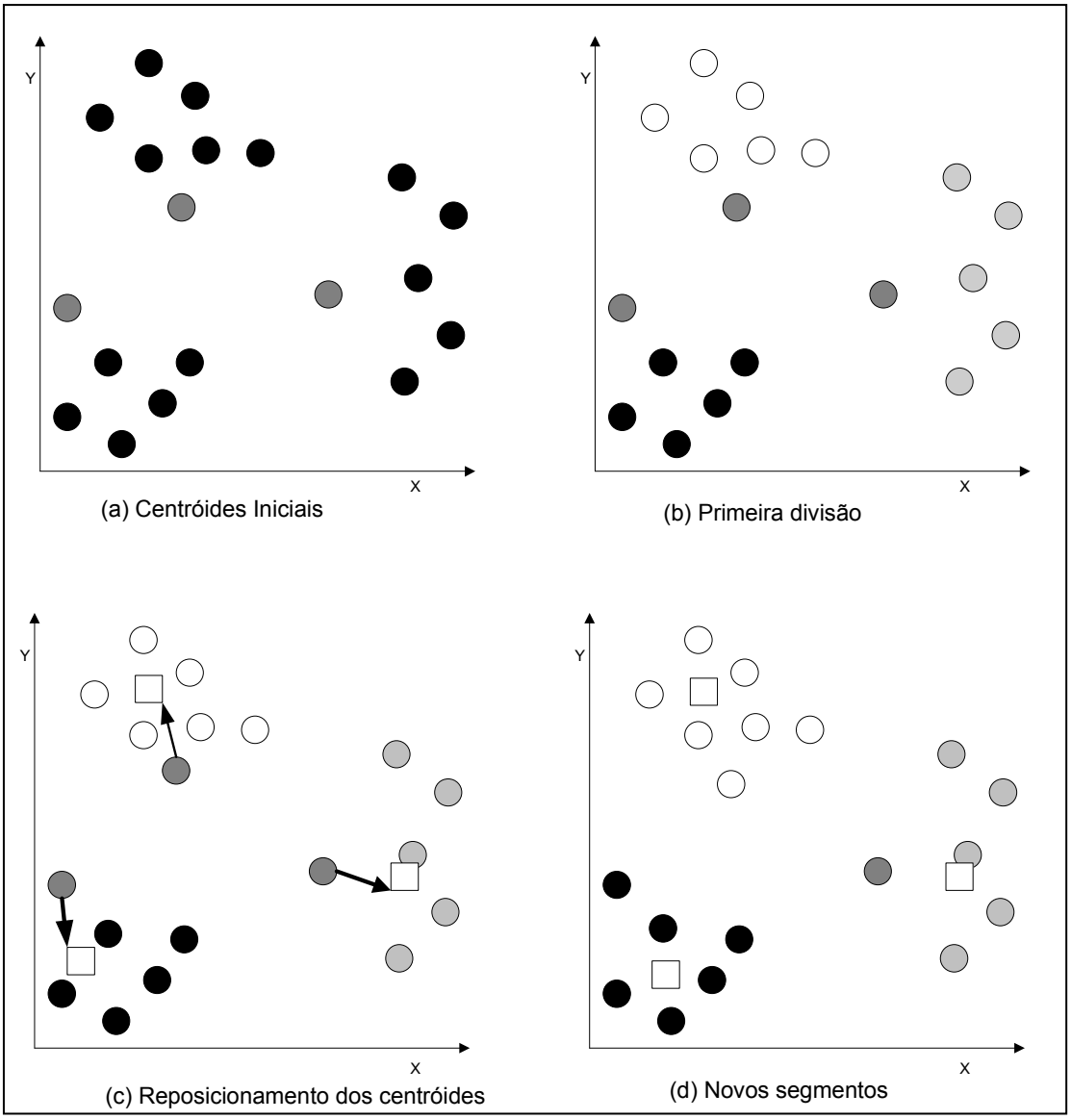


Figura 10: PROCESSO DE IDENTIFICAÇÃO DOS SEGMENTOS
Fonte: Santos e Ramos (2009)

Santos e Ramos (2009, p. 144) fazem a seguinte descrição para o funcionamento da Figura 10.

> Na Figura 10 (a) é apresentado o conjunto inicial dos dados, onde são selecionados três pontos aleatórios, o cinzento, como sendo o centróide do segmento. Para estes pontos, e na Figura 10 (b), verificam-se quais os registros (pontos) que integram cada um dos segmentos. No passo seguinte (Figura 10 (c)), são ajustados os centróides, através da verificação do ponto médio dos elementos de um dado segmento. Para as novas posições dos centróides é verificado a que segmento cada registro pertence, atendendo às alterações observadas na posição dos centróides (Figura 10 (d)). Este processo é sucessivamente repetido até que não sejam verificadas quaisquer mudanças na posição dos centróides e, como tal, nos registros que integram cada segmento.

2.2.4 Algoritmos para mineração de dados

Conforme as tarefas de mineração de dados (apresentado na seção 2.2.2) e as técnicas (apresentado na seção 2.2.3) os algoritmos são construídos e então implementados em uma ferramenta de mineração de dados. Nos próximos parágrafos são apresentados alguns algoritmos de mineração de dados, e também é feita uma breve descrição de um algoritmo para cada tarefa de mineração de dados apresentada.

Segundo Santos e Ramos (2009) as técnicas de mineração são concretizadas através de diferentes algoritmos.

Na seção 2.4 são apresentadas algumas ferramentas de mineração de dados, onde as mesmas são desenvolvidas para contemplar um algoritmo ou um conjunto de algoritmo de mineração de dados de acordo com o propósito da ferramenta. O Quadro 3 apresenta alguns algoritmos que são utilizados na ferramenta de mineração de dados WEKA.

Quadro 3: ALGUNS ALGORITMOS PARA MINERAÇÃO DE DADOS, WEKA[4] versão 3.7.1 (2010)

Tarefa de Mineração de Dados	Algortimo
Agrupamento	OPTICS
	SimpleKMeans
	K – Means

[4] http://www.cs.waikato.ac.nz/ml/weka. Acesso 30/12/2010

Classificação	J48
	RAndomTree
	UserClassifier
Associação	Apriori
	HotSpot
	GeneralizedSequentialPatterns

A tarefa de classificação é exemplifica através do algoritmo J48. Esse algoritmo gera uma árvore de decisão. Segundo Tan et al. (2009) este algoritmo consiste na entrada de registros de treinamento *E* e o conjunto de atributos *F*. Então o algoritmo funciona selecionando recursivamente o melhor atributo para dividir os dados e expandir os nodos da folha da árvore até que o critério de parada seja satisfeito.

A Tarefa de Associação é exemplificada através do algoritmo Apriori. Segundo Tan et al. (2009) este algoritmo faz uso de poda baseada em suporte para controlar de forma sistemática o crescimento exponencial dos conjuntos de itens candidatos. Inicialmente, cada item é considerado como um conjunto candidato de item 1. Após a contagem de seus suportes, os conjuntos de itens candidatos com menos transações são descartados.

Para a tarefa de agrupamento o algoritmo K-means é exemplificado. Tan et al. (2009) abordam que esse algoritmo primeiro escolhe K centróides iniciais, onde K é um parâmetro especificado pelo usuário (o número de grupos desejado). Cada ponto é atribuído ao centróide mais próximo, e cada coleção de pontos atribuídos a um centróide é um grupo. O centróide de cada grupo é então atualizado baseado nos pontos atribuídos ao grupo. É feita a repetição desses passos até que nenhum ponto mude de grupo.

Nesta seção foram apresentados alguns algoritmos para concretizar uma tarefa de mineração de dados, as quais podem utilizar uma ou mais técnicas para a mineração de dados. Os algoritmos: J48, Apriori e K-Means foram abordados devido ao fato de muitos algoritmos de mineração de dados serem baseados nesses algoritmos.

2.3 Metodologias para Mineração de Dados

A partir do momento que a mineração de dados passou a ser utilizada por empresas e a academia conseguiu comprovar sua eficiência, a mesma foi alvo de uma nova necessidade, que é a criação de metodologias para definir o ciclo de vida de um processo de mineração de dados.

Este trabalho aborda duas metodologias para a construção de um projeto de mineração de dados a CRISP-DM e a D^3M.

2.3.1 CRISP-DM

Segundo Chapman et al. (2000) a metodologia CRISP-DM foi concebida no final dos anos de 1996 pelas empresas NCR (Estados Unidos da América e Dinamarca), Daimler-Chrysler AH (Alemanha), SPSS Inc. (Estados Unidos da América) e OHRA (Grupo Bancário Holandês). Essas empresas reuniram suas experiências para o desenvolvimento de um processo padrão para mineração de dados e o batizaram de CRISP-DM.

Segundo Chapman et al. (2000) CRISP-DM é uma metodologia para as diferentes fases na implantação de um projeto de MD. Esta metodologia é definida em termos de um processo hierárquico, que constitui em um conjunto de tarefas descritas em quatro níveis de abstração (do geral para o específico).

A Figura 11 apresenta os quatro níveis de abstração que são: fases, tarefas genéricas, tarefas específicas e instâncias do processo com seus respectivos conjuntos de tarefas.

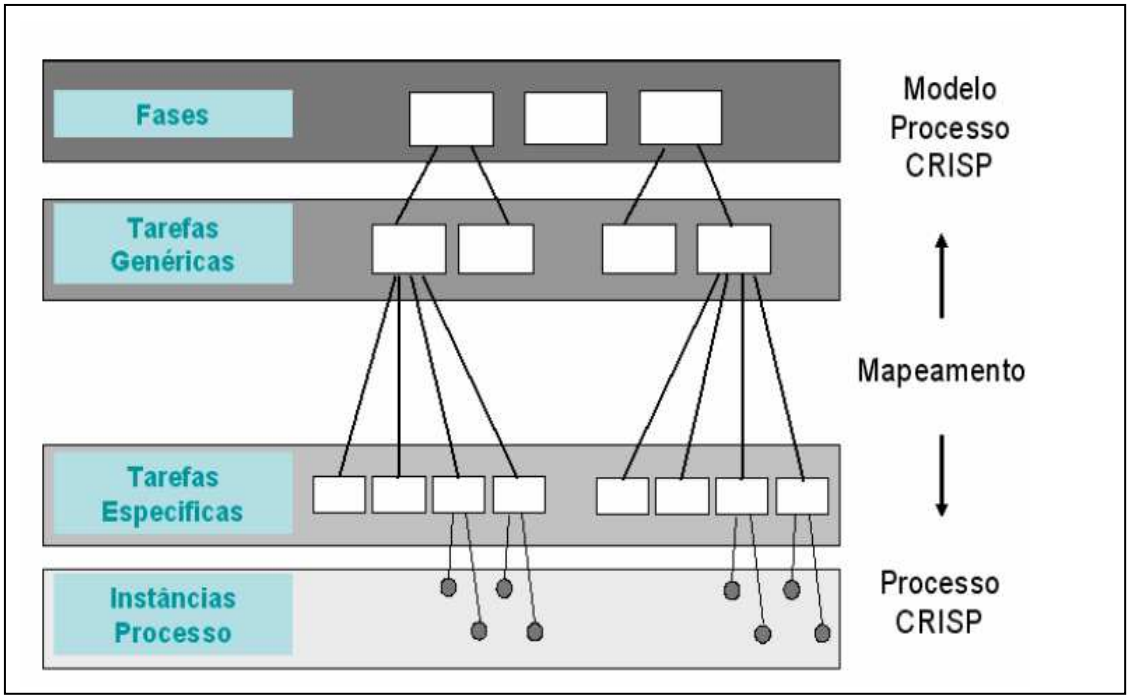


Figura 11: QUATROS NÍVEIS HIERÁRQUICOS DA METODOLOGIA CRISP-DM
Fonte: Chapman et al. (2000, pg. 9, tradução nossa)

A Figura 11 mostra que para cada uma das fases superiores há um conjunto de tarefas na próxima fase, havendo uma hierarquia entre as tarefas, onde as tarefas gerais estão em um nível superior e as mais específicas são uma ramificação das superiores.

O Quadro 4 apresenta uma descrição para cada uma dessas fases, segundo Chapman et al. (2000):

Quadro 4: DESCRIÇÃO DAS FASES DA METODOLOGIA CRISP-DM (CHAPMAN et al., 2000)

Nível	Tarefa
Fases	O processo de mineração de dados é organizado dentro de um número de fases; cada fase consiste de várias tarefas do segundo-nível. Exemplo: a tarefa “Compreensão do Negócio”.
Tarefas Genéricas	É chamado genérico, porque é suficiente para cobrir todas as possibilidades da situação da mineração de dados. A tarefa genérica tem como pretensão ser completa e estável. Exemplo: “Identificar Objetivos do Negócio”.
Tarefas Específicas	É o nível das tarefas especializadas, é o lugar para descrever como as ações nas tarefas deveriam ser executadas em certas situações específicas. Exemplo: “Informação do Negócio”.

Instâncias Processo	A instância do processo é um registro de ações, decisões e resultados de uma atual função da mineração de dados. Exemplo: Registrar os resultados da compreensão do negócio.

O Quadro 4 fez uma breve descrição das tarefas existentes na metodologia CRISP-DM sob uma visão hierárquica. Como exemplo, foi apresentada uma das fases dessa metodologia com algumas de suas tarefas, onde a tarefa de “Compreensão do Negócio” contém a tarefa de “Informação do Negócio”, e esta contém o “Registro dos Resultados da compreensão do Negócio”. A próxima seção descreve cada fase dessa metodologia e suas respectivas tarefas.

2.3.1.1 O Modelo de Referência CRISP-DM

Segundo Chapman et al. (2000), o ciclo de vida de um projeto de mineração de dados consiste em seis fases, que são: Entendimento do Negócio (*Business Understanding*), Entendimento dos Dados (*Data Understanding*), Preparação dos Dados (*Data Preparation*), Modelagem (*Modeling*), Avaliação (*Evaluation*) e Aplicação *(Deployment*).

A Figura 12 apresenta essas fases e o relacionamento entre elas. A sequência das fases não é rígida, pode acontecer das tarefas posteriores voltarem para as fases anteriores, isso depende dos resultados de cada fase. As setas indicam as mais importantes e frequentes dependências entre as fases.

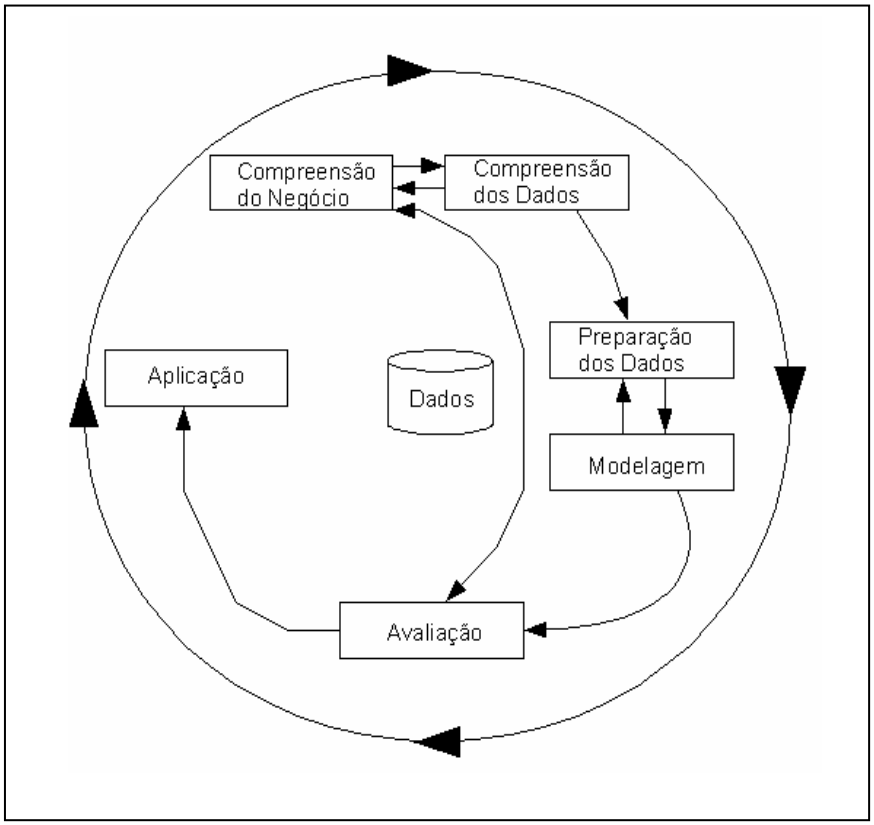


Figura 12: FASE DO MODELO DE REFERÊNCIA DO CRISP-DM
Fonte: Chapman et al. (2000, pg. 13, tradução nossa)

Chapman et al. (2000) faz a seguinte descrição para cada uma dessas fases:

- Entendimento do negócio: esta fase foca no entendimento dos objetivos e requisitos do projeto sob uma perspectiva de negócio;
- Entendimento dos dados: esta fase tem como objetivo um entendimento inicial dos dados, e logo em seguida possui atividades que permitem um entendimento dos dados, identificar problemas em alguns dados, e encontrar subconjuntos de dados interessantes para o projeto;
- Preparação dos dados: A fase de preparação dos dados cobre todas as atividades para preparar os dados para a realização das atividades;
- Modelagem: Nesta fase, várias técnicas de modelagem são selecionadas e aplicadas e seus parâmetros são calibrados para obter valores otimizados;

- Avaliação: Esta fase consiste em criar um modelo de alta qualidade a partir da análise dos dados, e certificar que os objetivos serão alcançados na primeira fase onde foram estabelecidos. Além disso, será levantado se alguma questão importante do negócio não foi suficientemente considerada. No final dessa fase, uma decisão sobre a utilização dos resultados da mineração deverá ser avaliada;
- Aplicação: Esta fase consiste em estruturar e apresentar os resultados obtidos com a aplicação da tarefa da mineração de dados de modo que o cliente possa utilizar.

Cada uma dessas fases possui um conjunto de atividades a serem realizadas. O Quadro 5 apresenta as fases com suas respectivas tarefas a serem executadas, segundo Chapman et al. (2000):

Quadro 5: TAREFAS EXECUTAS EM CADA ETAPA DO CRISP-DM, (CHAPMAN et al., 2000)

Fase	Tarefas
Entendimento do negócio	• Determinar os objetivos do negócio • Avaliar a situação • Determinar as metas de mineração de dados • Produzir o plano do projeto
Entendimento dos dados	• Coletar os dados iniciais • Descrever os dados • Explorar os dados • Verificar a qualidade dos dados
Preparação dos dados	• Selecionar os dados • Limpeza de dados • Construir dados • Integrar os dados • Formatar os dados
Modelagem	• Selecionar as técnicas de modelagem • Gerar o teste padrão • Construir o modelo • Avaliar o modelo
Avaliação	• Avaliar os resultados • Rever os processos • Determinar os passos seguintes

Aplicação	• Plano de Aplicação • Plano de modelagem & manutenção • Produzir relatório final • Rever o projeto

Além da Metodologia CRISP-DM há outras metodologias que podem ser aplicadas em projetos de mineração de dados, entre estas metodologias a D³M, que será abordada na próxima seção, onde algumas dos seus propósitos foram utilizados com a ontologia desenvolvida, para criar uma arquitetura orientada ao domínio para ferramentas de mineração de dados.

2.3.2 Mineração de Dados Orientada ao Domínio (D³M)

Segundo Cao e Zhang (2006) a mineração de dados realizada atualmente é baseada em metodologias orientadas aos dados, onde a visão dos resultados é feita de forma isolada, e tem como consequência a geração de muitos resultados que não interessam às necessidades dos negócios. Neste contexto, a metodologia D³M veio para propor que o processo de mineração de dados seja baseado em conhecimento do domínio do negócio e na cooperação entre humanos e máquinas durante a mineração.

Os fatores que justificam a criação e desenvolvimento da metodologia D³M, é que as metodologias atuais de mineração de dados apresentam as seguintes características, segundo Cao e Zhang (2006):

- Padrões que são interessantes para os negócios frequentemente estão ocultos em grande quantidade de dados com estruturas de dados complexas, dinâmicas, e origem distribuída;
- Distanciamento entre o meio acadêmico e o interesse do negócio;
- Padrões interessantes frequentemente não podem ser implementados na vida real, se eles não são integrados com regras do negócio, regulamentos e processos.

Para a metodologia D³M foi criado um *framework* chamado DDID-PD (*domain-driven in-depth pattern discovery*), que considera os seguintes componentes chaves, segundo Cao e Zhang (2006):

- Restrições baseadas em contexto: A descoberta de padrões envolve uma profunda compreensão do ambiente em torno do domínio do problema, dos seus dados e da análise de seus objetivos.
- Integração do domínio de conhecimento: A integração de conhecimentos de domínio está sujeito à forma como ele pode ser representado e preenchido para o processo de descoberta de conhecimento. Ontologia baseada em representação do conhecimento de domínio é uma das abordagens adequadas para o modelo de conhecimento do domínio.
- Cooperação entre homem e máquina: consiste em haver uma mineração de dados cooperativa entre especialistas e sistemas de mineração em todo o processo.
- Mineração em profundidade: consiste em avaliar e refinar conhecimento acionável, na tentativa de buscar resultados mais interessantes para o objetivo do negócio.
- Descoberta de regras ou padrões mais interessantes para o negócio (o que é chamado de conhecimento acionável): consiste em disparar regras quando forem descobertos dados que satisfaçam um determinado conhecimento acionável.
- Processo do refinamento iterativo dos resultados: consiste em criar sub-cenários a partir do cenário principal e então refinar os padrões gerados a partir do sub-cenário criado.
- Suporte de mineração paralela e interativa: consiste em obter pedidos dos usuários, gerenciar informações e usar algoritmos para processá-los em máquinas distintas.

A Figura 13 apresenta o *framework* DDID-PD, onde as tarefas da metodologia D³M apresentada nos parágrafos anteriores são combinadas com as tarefas tradicionais da mineração de dados.

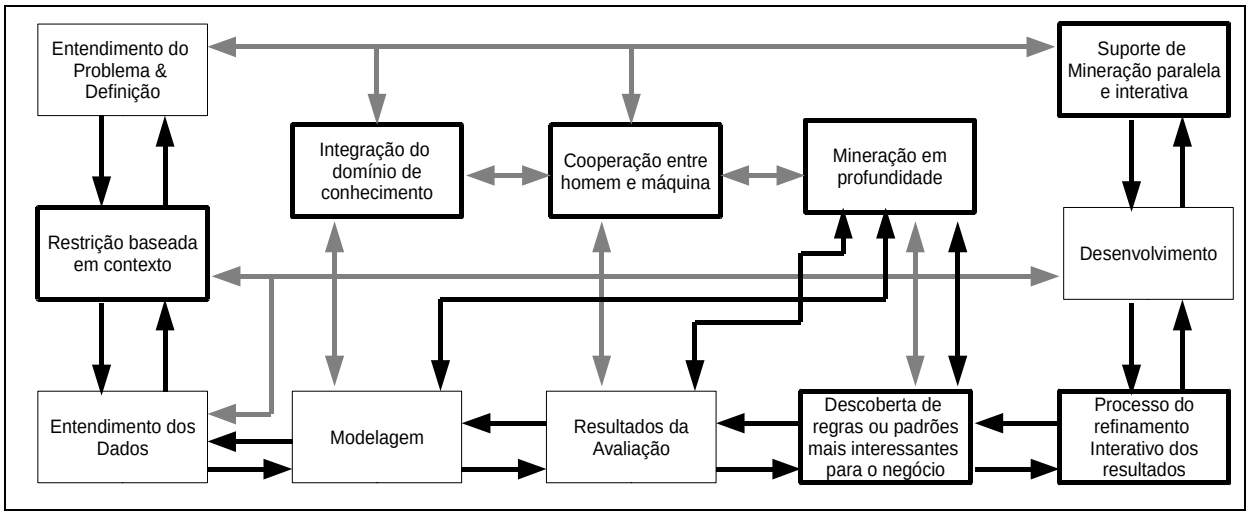


Figura 13: MODELO DO PROCESSO DDID-PD
Fonte Cao e Zhang (2006, pg 53, tradução nossa)

Segundo Cao e Zhang (2006), a sequência apresentada na Figura 13 não é rígida; algumas fases podem ser descartadas ou deslocadas para uma fase à frente ou atrás em uma aplicação real. Cada passo do processo DDID-PD pode envolver conhecimento de domínio e auxilio de especialistas do domínio. Esse *framework* tem os componentes chaves de suporte para mineração de dados orientada ao domínio, que são críticas para o sucesso de um processo de mineração de dados no mundo real.

2.4 Ferramentas para Mineração

Atualmente há várias ferramentas de mineração de dados que executam um ou vários processos na descoberta de padrões a partir dos dados. Algumas dessas ferramentas foram desenvolvidas em trabalhos de conclusão de mestrado e doutorado nas faculdades onde há cursos nesses níveis; outras são de propriedades de empresas já consagradas no mercado. Dentre algumas dessas

ferramentas estão: *Bramining*[5], Pacote *Weka*[6], *DBminer*[7], *Oracle Data Mining*[8] e Kira (Mendes, 2009).

Cada uma dessas ferramentas tem uma ou mais funções dentro do campo de mineração de dados. As subseções a seguir apresentam uma breve descrição da ferramenta Weka, por sua grande popularidade, e da ferramenta Kira, ter sido desenvolvida com o diferencial de ser amigável para o usuário.

2.4.1 A Ferramenta WEKA

Weka é uma abreviatura para *Waikato Environment for Knowledge Analysis*. Segundo Witten & Frank (2006) essa ferramenta contém um pacote de algoritmos para mineração de dados e ferramentas de pré e pós-processamento.

Weka foi desenvolvida na Universidade de Waikato na Nova Zelândia. Ela foi desenvolvida em Java e distribuída sobre os termos de licença GNU. Pode ser executada em várias plataformas, como é o caso do *Windows, Linux* e *Macintosh*. Fornece uma interface uniforme para a aplicação de vários algoritmos de aprendizado, juntamente com os métodos de pré e pós-processamento. A Figura 14 apresenta a interface inicial da ferramenta Weka.

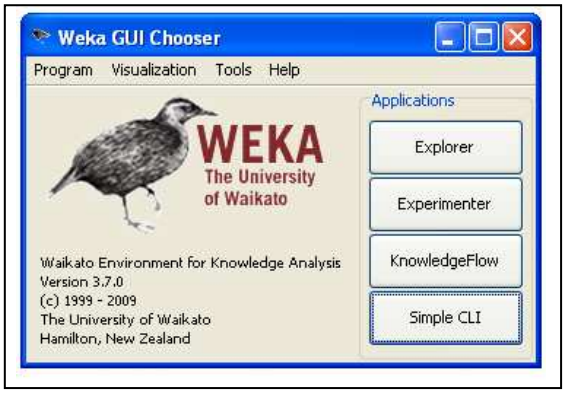


Figura 14: TELA INICIAL DA FERRAMENTA WEKA
Fonte: Ambiente de Desenvolvimento do WEKA

[5] www.graal-corp.com.br. Acesso: 16 jun. 2010.
[6] www.cs.waikato.ac.nz. Acesso: 16 jun. 2010.
[7] http://db.cs.sfu.ca/DBMiner. Acesso: 16 jun. 2010.
[8] www.oracle.com. Acesso: 16 jun. 2010.

A partir da interface da Figura 14, é possível acessar as funcionalidades da ferramenta. Segundo Witten & Frank (2006), Weka fornece implementações de algoritmos de aprendizagem que podem ser aplicadas no conjunto de dados; isto inclui uma variedade de ferramentas para transformação de conjuntos de dados, tais como algoritmos para discretização. Pode-se fazer o pré-processamento de um conjunto de dados e analisar os resultados de classificação e sua performance, isso sem escrever qualquer código de programa. A Figura 15 mostra um dentre os ambientes de trabalho da Weka.

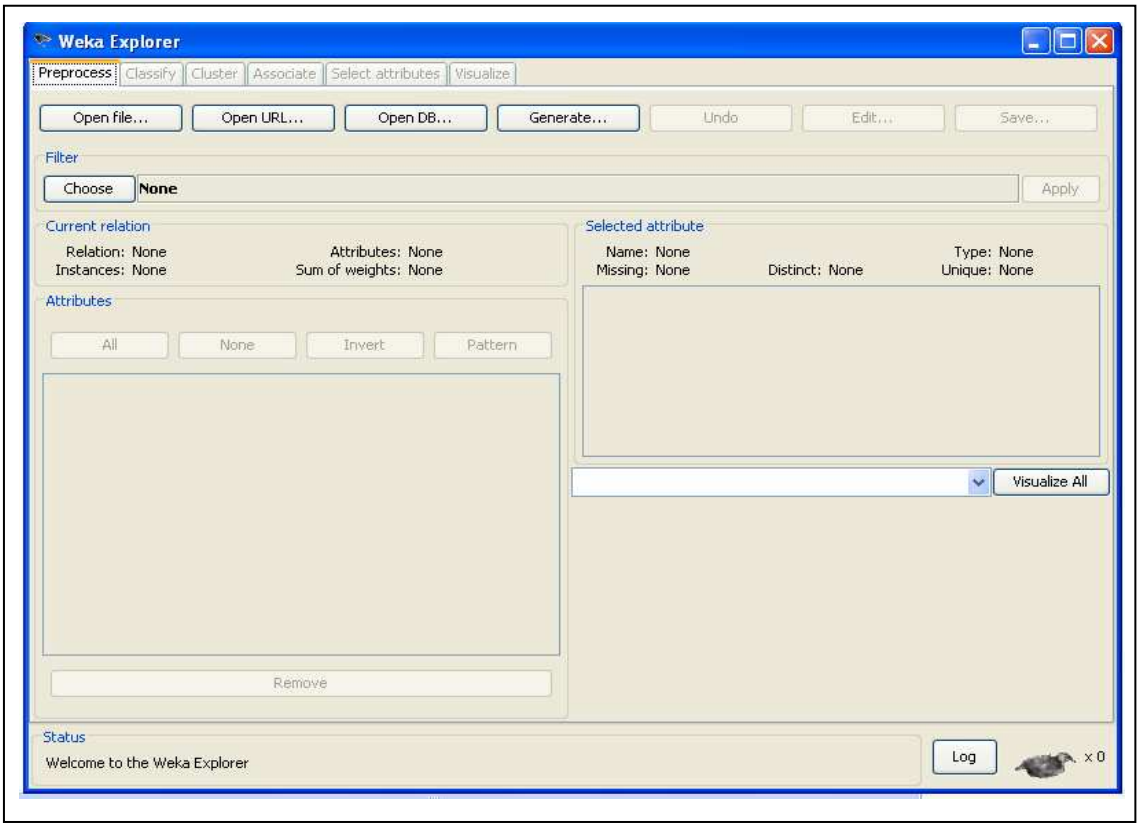


Figura 15: AMBIENTE DE DESENVOLVIMENTO WEKA
Fonte: Ambiente de Desenvolvimento do WEKA

A utilização dessa ferramenta é por meio de abas e seleção do que o usuário deseja, conforme é apresentado na Figura 15. Entre as vantagens encontradas ao utilizar esta ferramenta está o fato de dar suporte a vários algoritmos de mineração de dados consagrados tanto no meio acadêmico como em aplicações de negócio; o ambiente de trabalho é totalmente gráfico, há livros e outras publicações que servem como manuais de referência para realizar o processo de descoberta de informações a partir de uma base de dados, como, por exemplo: livro

de Witten e Frank (2005) e no sítio www.cs.waikato.ac.nz/ml/weka/ onde podem ser obtidos vários artigos relacionados à aplicação da ferramenta Weka em projetos de mineração de dados.

Algumas dificuldades foram levantadas ao utilizar esta ferramenta: dificuldades na fase de preparar os dados de maneira que a ferramenta possa trabalhar; o usuário deve ter um bom conhecimento de mineração de dados para definir o que se deseja fazer e interpretar os padrões gerados. Estas dificuldades foram constatadas ao realizar projetos fictícios de mineração de dados.

2.4.2 Ferramenta de Mineração de Dados Kira

A ferramenta de mineração de dados Kira foi desenvolvida por Mendes (2009), como parte de um trabalho de mestrado.

A intenção dessa ferramenta é abstrair grande parte do conhecimento exigido do analista de dados para executar a tarefa de mineração de dados. É possível realizar um projeto de mineração de dados a partir das orientações que são descritas em cada uma das telas dessa ferramenta.

A Figura 16 apresenta a arquitetura da ferramenta KIRA, que possui três camadas. Estas camadas são descritas nos próximos parágrafos.

A Camada 1 está relacionada com a interface do usuário, é onde o usuário pode ver e inserir informações para realizar um projeto de mineração de dados, entre estas operações estão definir o problema e objetivo da mineração de dados e ser orientado pela a ferramenta nas diversas operações a serem realizadas durante o projeto de mineração de dados, como preparar os dados, definir a tarefa de mineração de dados e visualizar os resultados produzidos.

Na Camada 2 ocorre as operações lógicas da ferramenta, onde os dados são preparados, é feito a análise dos dados, e é realizado a execução da mineração de dados, além disso nesta camada foi implementado um sistema especialista desenvolvido por Silva et al. (2009) para definir qual tarefa de mineração é a mais adequada para o projeto de mineração de dados que está em execução.

A Camada 3 está relacionada com o repositório de informações, onde estão os diferentes tipos de dados, como, por exemplo: a base de dados a ser

minerada, dados prontos para ser minerados, repositório de informações entre outros.

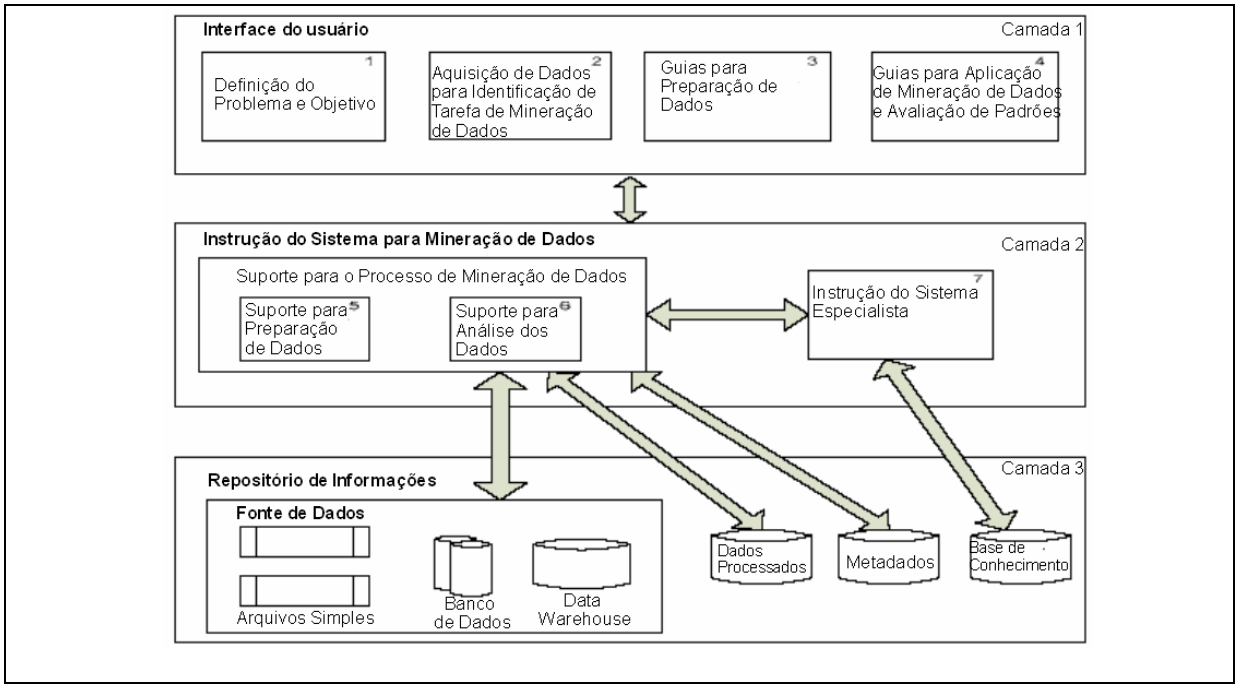


Figura 16: ARQUITETURA DA FERRAMENTA KIRA
Fonte: Silva et al. (2009, pg 13)

Segundo Mendes (2009), cada módulo da Camada 1 (Interface do Usuário) têm como objetivo oferecer facilidades para o usuário preparar os dados, executar o algoritmo de mineração e avaliar os padrões obtidos. Inicialmente, o usuário tem acesso às funções para ajustar os parâmetros das fontes de dados a serem usadas. Depois o usuário é orientado a oferecer informações específicas, escolher e estruturar os dados destinados, escolher e executar um algoritmo de mineração de dados e analisar os resultados gerados. No final de cada fase, o usuário é informado sobre o próximo passo do processo. Estes processos guias acompanham o usuário na execução de cada etapa não tendo a necessidade de conhecimentos detalhados sobre o processo de mineração de dados. A Figura 17 apresenta uma das interfaces da ferramenta Kira, onde mostra uma visão geral dos passos a serem seguidos para a realização da mineração de dados.

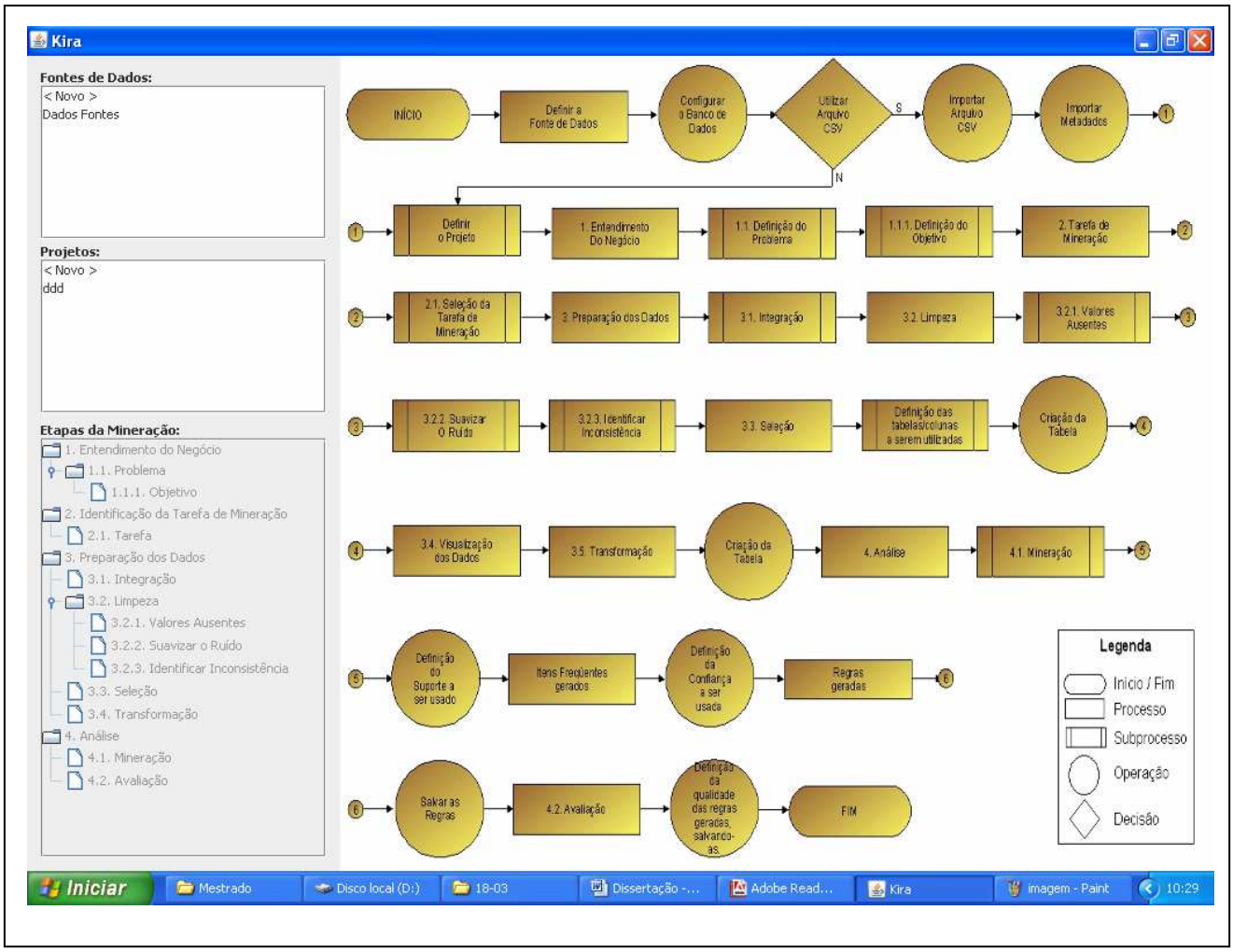


Figura 17: TELA DA FERRAMENTA KIRA
Fonte: Ambiente de desenvolvimento da Ferramenta Kira

Entre os componentes da interface desta ferramenta estão:

- O diagrama de blocos: demonstra cada passo que a ferramenta irá executar;
- No canto direito inferior existe uma legenda que identifica a representação de cada um dos itens que compõem o diagrama de blocos;
- A paleta à esquerda está dividida da seguinte forma:
    - Fontes de dados, onde é indicada a fonte de dados utilizada;
    - Projetos, nome atribuído ao projeto de mineração de dados;
    - Etapas da mineração indicam cada uma das etapas a ser realizadas.

A Figura 18 apresenta uma das intefaces da ferramenta Kira para executar a atividade de definir o problema e objetivo de um projeto de mineração de dados.

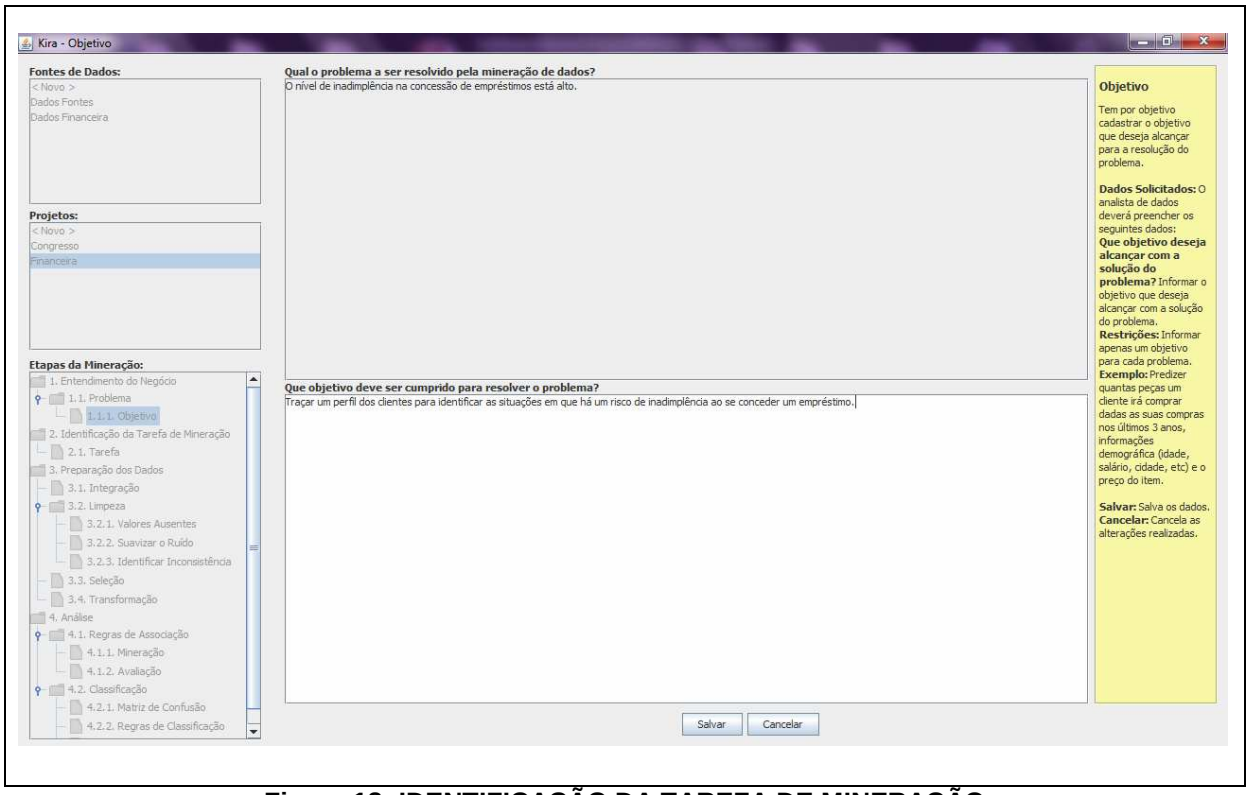


Figura 18: IDENTIFICAÇÃO DA TAREFA DE MINERAÇÃO
Fonte: Ambiente de Desenvolvimento da Ferramenta

A partir da Figura 18, o minerador de dados faz uma análise do propósito do projeto de mineração de dados e auxliado pela a guia de descrição do que deve ser feito no lado esquerdo da ferramenta defini o objetivo e o problema do projeto de mineração de dados a ser realizado.

O funcionamento desta ferramenta consiste em chamar a tela com o diagrama de blocos apresentada na Figura 17, e então é indicado o próximo processo a ser executado e logo, em seguida, uma outra tela é exibida com campos a serem preenchidos auxiliados por mensagens, como apresentado na Figura 18.

O funcionamento desta ferramenta resumidamente consiste em executar as seguintes etapas:

- Escolher uma base de dados,
- Atribuir um nome ao projeto,
- Descrever o problema a ser resolvido e objetivo a ser cumprido,

- Fazer a preparação dos dados,
- Selecionar a tarefa de mineração de dados.
- Avaliar os resultados da mineração de dados.

Na etapa final a ferramenta mostra o resultado dos padrões gerados. Em todas as etapas, no lado direito há uma tag que indica ao usuário o que fazer.

2.5 Considerações Finais

Um projeto de mineração de dados envolve várias fases. A metodologia CRISP-DM foi desenvolvida para elaboração de projetos de mineração de dados orientados a dados onde disciplina e descreve o que o minerador deve fazer no conjunto de etapas da metodologia.

A metodologia D³M foi desenvolvida para tornar a mineração de dados mais interativa, cujo propósito é uma mineração feita a partir do conhecimento do domínio, onde suas tarefas são combinadas com as tarefas de uma metodologia orientada a dados na tentativa de buscar resultados melhores.

As fases da CRISP-DM e a necessidade do conhecimento humano no processo de mineração de dados são a base para o desenvolvimento da ontologia META-DM. Porém para realizar esta tarefa foi necessário fazer um levantamento bibliográfico sobre ontologias, que é abordado no próximo capítulo.

3 ONTOLOGIAS

3.1 Considerações Iniciais

Ontologias vêm sendo aplicadas em vários domínios da computação (Web Semântica, Segurança da Informação, Mineração de Dados entre outros) como forma de representar determinado domínio por meio de conceitos e relacionamentos entre eles.

Neste capítulo é descrito alguns aspectos sobre ontologias, considerados importantes, que são necessários para o desenvolvimento da ontologia de domínio proposta. Entre estes aspectos estão: conceito e classificação de ontologias, metodologias para criar ontologias, linguagens para implementação de ontologias e ferramentas para desenvolver ontologias.

As subseções abaixo detalham cada um desses itens com o objetivo de levantar informações bibliográficas suficientes para o desenvolvimento da ontologia de domínio proposta.

3.2 Definições

Um projeto, independentemente de sua natureza (seja computacional, de engenharia, médica e outros), tem várias pessoas envolvidas. Dessa forma, há necessidade que os envolvidos em um determinado projeto entendam o que os outros fizeram na fase anterior, para que possam começar uma função da melhor forma possível. Diante da necessidade de um constante entendimento de uma situação surgiu o termo ontologia.

Há diversas definições para o termo ontologia, como a utilizada por Falbo et al. (1998, p. 2) a seguir:

> Uma ontologia é uma especificação de uma conceitualização que é uma descrição de conceitos e relações que podem existir para um agente de software ou um agente da comunidade. Basicamente, uma ontologia consiste de conceitos e relações, e suas definições, propriedades e restrições são expressas como axiomas. Uma Ontologia não é somente uma hierarquia de termos, mas uma estrutura falando sobre um domínio. (tradução nossa)

A maioria das definições para o termo ontologia, inclusive a utilizada por Falbo et al. (1998), são extensões da definição de Gruber (1993), que é amplamente aceita na comunidade científica.

Segundo Gruber (1993), uma ontologia é "*uma especificação explícita de uma conceituação*", ou seja, uma representação de um determinado conhecimento de maneira formal.

Neste trabalho, será adotada a definição feita por Falbo et al. (1998), pois ela se mostra completa e deixa explícito o uso de ontologias para expressar conhecimento sobre determinado domínio, o que vai ao encontro, ao objetivo deste trabalho.

Segundo Noy & McGuinness (2001) a primeira aplicação de ontologias, na área de Ciência da Computação, foi no campo da inteligência artificial em meados dos anos 90. Desde então, as ontologias têm sido aplicadas em vários ramos da Ciência da Computação, como, por exemplo: web semântica, engenharia de software, arquitetura da informação, dentre outros, como uma forma de representar conhecimento sobre o mundo ou parte desse mundo.

3.3 Classificação das Ontologias

De acordo com Linhalis (2007), ontologias podem ser classificadas sob diversos aspectos, como, por exemplo, o grau de formalismo do vocabulário, a estrutura, o conteúdo da conceituação, dentre outros. Uma das classificações mais citadas na literatura é a feita por Guarino (1998), que classifica as ontologias segundo sua função e as divide em três níveis. A Figura 19 apresenta essa divisão e a inter-relação dos tipos de ontologias.

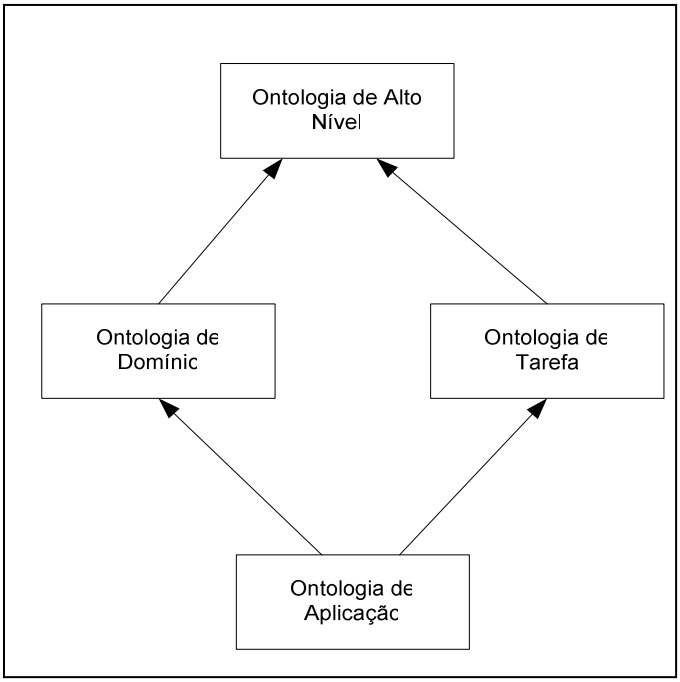


Figura 19: CLASSIFICAÇÃO DE ONTOLOGIAS
Fonte: Guarino (1998, pág: 9, tradução nossa)

O Quadro 6 faz uma breve descrição dos níveis que compõem a classificação de ontologias de acordo com Guarino (1998).

Quadro 6: DESCRIÇÃO DOS TIPOS DE ONTOLOGIAS, GUARINO (1998)

Tipo de Ontologia	Descrição
Alto Nível	Descreve conceitos gerais, como espaço, tempo, ação e outros; que são independentes de um problema ou domínio específico. Exemplo: Cyc[9], WordNet[10] e SUMO[11]
Domínio e Tarefa	Refere-se, respectivamente, a um vocabulário relacionado a um domínio (como medicina ou automóveis) ou a uma tarefa ou atividade (como diagnóstico e venda), que especializa os termos introduzidos na ontologia de alto nível. Exemplo: ontologia de veículos, documentos e computadores.
Aplicação	Descreve conceitos que dependem de domínios e tarefas particulares, que são frequentemente especializações das ontologias relacionadas. Exemplo: Uma ontologia que trabalhe com carros de luxos, que especializará conceito da ontologia de veículos (que é uma ontologia de domínio).

[9] http://www.cyc.com/. Acesso em: 03 set. 2010.
[10] http://wordnet.princeton.edu/. Acesso em: 03 set. 2010.
[11] http://ontology.teknowledge.com/. Acesso em: 03 set. 2010.

3.4 Metodologias para Desenvolvimento de Ontologias

Vários autores desenvolveram etapas a serem seguidas para construir ontologias, organizadas de modo que forneça suporte para que os envolvidos em um determinado projeto saibam que técnicas são mais apropriadas e o que cada um vai produzir. Ao conjunto dessas etapas deu-se o nome de metodologias. Este trabalho utiliza duas metodologias para o desenvolvimento de ontologias: a metodologia de Noy e McGuiness (2001) e a Metodologia METHONTOLOGY.

Na metodologia desenvolvida por Noy e McGuiness (2001) propõem-se as seguintes fases:

1) Determinar o domínio e o escopo da ontologia: consiste em verificar o que a ontologia irá cobrir e assim limitar o escopo do modelo. Ao longo do desenvolvimento esta fase pode ser alterada de acordo com o amadurecimento dos reais propósitos da criação da ontologia;

2) Considerar a reutilização de outras ontologias: consiste em verificar o que já foi criado e refinar ou estender para o domínio ou tarefa no qual se deseja trabalhar;

3) Enumerar termos importantes para a ontologia: consiste em encontrar os termos mais comuns no domínio e as propriedades que eles possuem;

4) Definir as classes e a hierarquia entre elas: consiste em observar a clareza e a consistência da hierarquia ao serem criadas subclasses. Isto é, deve-se observar se uma classe tem subclasses a mais ou a menos;

5) Definir as propriedades das classes: consiste em criar alguns conceitos na hierarquia, e, logo em seguida suas propriedades;

6) Definir as facetas das propriedades: consiste em descrever os valores de tipos, valores permitidos, número máximo e mínimo (cardinalidades) para os valores das propriedades, e outros;

7) Criar instâncias: consiste em escolher a classe para a qual se deseja criar as instâncias, criar uma instância e preencher os valores das propriedades para cada instância.

Linhalis (2007) ressalta que a metodologia de *Noy e McGuinness* se concentra principalmente na fase de conceituação, que é a mais crítica no desenvolvimento de uma ontologia, pois é a mais ligada à definição do conhecimento.

A metodologia de Noy e McGuiness (2001) foi adotada com o objetivo de definir conceitos, propriedades e relacionamentos para o domínio de mineração de dados. A metodologia dá ênfase à fase de conceituação, e por isso se mostra bem adequada para este propósito.

A proposta da metodologia *METHONTOLOGY,* segundo Fernández et al. (1997), baseia-se na construção de ontologias a partir do ponto zero, e podem ser utilizadas outras ontologias ou não. Os autores fazem uma comparação do ciclo de vida de uma ontologia com o processo de desenvolvimento de um software tradicional e ressaltam que é bem complicado levantar todos os requisitos necessários antes de começar o processo de desenvolvimento. As fases do ciclo de vida de uma ontologia para esta metodologia são: especificação, aquisição do conhecimento, conceituação, integração, implementação, avaliação e documentação.

Para documentar o ciclo de vida do desenvolvimento da ontologia a metodologia METHONTOLOGY foi adotada. O Quadro 7 faz uma descrição de cada uma dessas fases:

Quadro 7: ETAPAS DA METODOLOGIA METHONTOLOGY, Fernández et al. (1997)

Fases	Descrição
Especificação	A meta dessa fase é produzir um documento de especificação da ontologia, escrito em uma linguagem natural, onde é usado um conjunto de representação intermediária ou questões de competência. Nesta fase é proposto que no mínimo as seguintes informações devem ser incluídas: • O propósito da ontologia: incluindo usuário, cenários de uso, usuários finais, etc. • Nível de formalidade da ontologia implementada: depende da formalidade que irá ser usada para codificar os termos e seus significados. O grau de formalidade pode ser altamente formal, semi-

	formal ou rigorosamente formal. • Escopo que a ontologia irá cobrir: inclui um conjunto de termos a ser representado, suas características e granulosidade.
Aquisição de conhecimento	Esta fase é realizada simultaneamente com a fase de especificação, e está relacionada a adquirir conhecimentos necessários para começar o processo de criação da ontologia, é um processo independente do desenvolvimento da ontologia, no entanto coincide com outras atividades. Para adquirir o conhecimento necessário são utilizadas consultas a especialistas, livros, manuais, figuras, tabelas e mesmo outras ontologias como fonte de conhecimento. Esses elementos podem ser usados em conjunto com: brainstorming, entrevistas, análise de texto formal e informal e ferramentas de aquisição de conhecimento.
Conceituação	Nesta fase será estruturado o conhecimento de domínio em um modelo conceitual que irá descrever o problema e suas soluções em termos do vocabulário de domínio identificado na atividade de especificação da ontologia. A primeira atividade a ser realizada é construir um completo Glossário de Termos (conceitos, instâncias, verbos e propriedades), que irá resumir tudo o que é útil e potencialmente utilizável no conhecimento de domínio e seu significado. Uma vez completado o glossário de termos, deve-se agrupar os termos em conceitos (dicionário de dados que descreve e reúne tudo o que é útil e potencialmente usado no conceito de domínio, seus significados, atributos e instâncias) e verbos (ações no domínio). No final dessa fase será produzido um modelo conceitual expresso como um conjunto de conceitos bem definidos, que permitirá ao usuário final: verificar se a ontologia será ou não útil e utilizável para uma aplicação sem inspecionar seu código fonte; e comparar o escopo e plenitude de várias ontologias, sua reusabilidade e compatibilidade pela análise do conhecimento.
Integração	Como meta de acelerar o processo de desenvolvimento de uma ontologia, pode-se considerar o reuso de definições já desenvolvidas, dentro de outras ontologias, ao invés de começar a construção do seu início.
Implementação	Consiste em implementar a ontologia em uma linguagem formal, tais como: CLASSIC, OWL, LOOM, Ontolingua, ou uma outra linguagem de programação. Nesta fase é requerido um ambiente de desenvolvimento de ontologias e que deve pelo menos incluir: uma analise léxica e sintática, um tradutor, um editor, um navegador, realizar pesquisa de termos e apresentação dos resultados produzidos.
Avaliação	Realizar um julgamento técnico da ontologia, seu ambiente de software e documentação. A avaliação inclui os termos de verificação e validação. Verificação refere-se ao processo técnico que garante a correção de uma

	ontologia e validação garante que a ontologia desenvolvida representa o domínio do conhecimento definido na fase de especificação.
Documentação	Para cada uma das fases anteriores é feito um documento descrevendo o que foi realizado. A documentação é parte integrante do desenvolvimento da ontologia. Assim esta etapa está presente em todas as anteriores.

O conjunto dessas etapas forma o ciclo de vida de desenvolvimento de uma ontologia. Esta metodologia foi adotada neste trabalho para registrar cada um dos passos do desenvolvimento da ontologia.

3.5 Linguagens para Representação de ontologias

Segundo Linhalis (2007), no início dos anos 90 um conjunto de linguagens para implementação de ontologias foram desenvolvidas a fim de formalizar as informações nas ontologias. Essas linguagens utilizam um ou mais formalismos para representar conhecimento, tais como regras de produção, lógica de primeira ordem, lógica de segunda ordem, *frames*, redes semânticas e lógica de descrições (Russel & Norving, 2003).

O Quadro 8 mostra algumas das linguagens para implementação de ontologias (Linhalis, 2007).

Quadro 8: LINGUAGENS PARA REPRESENTAR ONTOLOGIAS (LINHALIS, 2007)

Linguagem	Característica
Ontolingua[12]	• Foi desenvolvida em 1992; • Combina *frames* e lógica de primeira ordem para representar conhecimento; • Apresenta taxonomia de conceitos, relações n-árias, funções, axiomas, instâncias e procedimentos; • O grande poder de expressividade dessa linguagem dificulta o desenvolvimento de inferência.
LOOM[13]	• Foi desenvolvida no mesmo período da Ontolingua; • Inicialmente, não foi criada para representar ontologias, mas bases de conhecimentos genéricas; • É baseada em lógica de descrições e regras de produção;

[12] http://ontolingua.stanford.edu/. Acesso: 16 jun. 2010
[13] http://www.isi.edu/isd/LOOM/. Acesso: 16 jun. 2010.

	• Fornece a classificação automática de conceitos e pode representar taxonomias de conceitos, relações n-áreas, funções, axiomas e regras de produção.
OCML[14]	• Foi desenvolvida em 1993; • É considerada uma Ontolingua operacional; • A maioria das definições que são representadas em OCML. São similares a Ontolingua e alguns componentes adicionais podem ser definidos, como regras dedutíveis e de produção.
F-Logic[15]	• Combina frames e lógica de primeira ordem, permite a representação de taxonomias de conceitos, relações binárias, funções, instâncias, axiomas e regras dedutíveis; • Sua máquina de inferência, Ontobroker, pode ser utilizada para a verificação de restrições e para deduzir novas informações.
KIF[16]	• Foi uma das primeiras linguagens criadas especificamente para representar informações a serem transmitidas de um programa a outro; • Utiliza lógica de primeira ordem e representa objetos, funções e relações.
OIL[17]	• Herda primitivas baseadas em frames RDF(S)[18], mas sua semântica formal é baseada em lógica de descrições.
DAML+OIL[19]	• Incorpora aspectos tanto da linguagem DAML (DARPA *Agente Markup Language*) quanto da linguagem OIL; • Tanto OIL quanto DAML+OIL permitem a representação de taxonomias, relações binárias, funções e instâncias.
OWL[20]	• Esta linguagem se consolidou como padrão da web semântica; • Foi criada a partir da DAML+OIL; • Mais detalhes dessa linguagem são abordados a seguir.

Para a implementação deste projeto foi utilizada a linguagem OWL, que utiliza recursos das linguagens RDF e RDFS. A escolha da linguagem OWL se justifica pelo fato de ser um padrão amplamente recomendado para a codificação de ontologias não apenas para a Web Semântica. A seguir é feita uma breve descrição de cada uma delas.

Segundo Manola e Miller (2004) RDF é uma linguagem para representar informação sobre recursos na *World Wide Web*. Destina-se

[14] http://technologies.kmi.open.ac.uk/ocml. Acesso: 16 jun. 2010.
[15] http://semanticweb.org/wiki/KWTR:_f-logic. Acesso: 16 jun. 2010.
[16] http://www-ksl.stanford.edu/knowledge-sharing/kif/ Acesso: 16 jun. 2010.
[17] http://xml.coverpages.org/oil.html. Acesso 16 jun. 2010.
[18] Combinação de RDF e RDF esquema.
[19] http://www.daml.org/2001/03/daml+oil-index.html. Acesso: 16 jun. 2010.
[20] http://semantic-web.indelv.com/web-ontology-language-owl.html. Acesso: 16 jun 2010.

particularmente a representar metadados sobre recursos da Web, como o título, autor e data de modificação de uma página Web, *copyright* e licença de informações sobre um documento da Web, ou o cronograma de disponibilidade de algum recurso compartilhado.

Segundo Breitman (2005) o RDF possui três elementos fundamentais que são: sujeito, predicado e objeto. Sujeito é um elemento que expressa algo, predicado é o que se fala de um elemento (são os relacionamentos) e objeto (pode ser uma URI ou uma string) é uma descrição expressa a partir do sujeito e predicado. Ao conjunto desses três elementos é dado o nome de tripla. A Figura 20 apresenta a representação de uma tripla sob forma de um grafo dirigido.

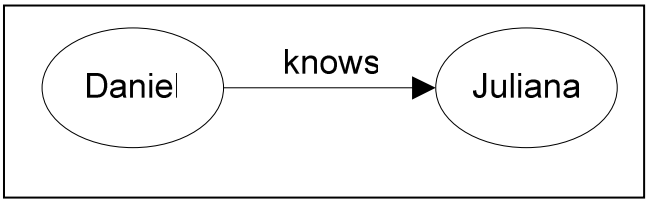


Figura 20: GRAFO REPRESENTANDO UMA TRIPLA
Fonte: Linhalis (2010)

A Figura 20 poderia ser interpretada da seguinte forma: Daniel conhece Juliana. Segundo Breitman (2005) esta representação da Figura 20 pode ser representada por sintaxe XML, onde um documento RDF é representado utilizando-se um elemento XML com a etiqueta rdf:RDF. O conteúdo desse elemento é um conjunto de descrições que utilizam a etiqueta rdf:description. Entre as descrições existentes estão: about (faz referência a um recurso existente), ID (cria um novo recurso) e se não atribuir nenhuma descrição cria-se um atributo anônimo.

Assim a Figura 20 poderia ser representada da seguinte forma:

```
<rdf:RDF>
    <rdf:Description about:” Daniel>
        <f:knows>
            Juliana
        </f:knows>
    </rdf:Description>
</rdf:RDF>
```

Mais detalhes da linguagem RDF podem ser encontrados em Manola e Miller (2004).

Segundo Linhalis (2010) esta linguagem, se usada sozinha, não representa a semântica por trás das descrições. Essa capacidade de descrever semântica é fornecida pelas Linguagens RDF Schema e OWL.

Segundo Brickley e Guha (2003) a linguagem RDF Schema é uma extensão semântica do RDF, que fornece mecanismos para descrever grupos de recursos e as relações entre esses recursos.

De acordo com Linhalis (2010) com a linguagem RDFS é possível organizar as classes e propriedades em hierarquias de generalização/especialização, definir domínios e conjuntos de valores esperados para propriedades, definir membros de classes, especializar e interpretar tipos de dados.

A Figura 21 apresenta um exemplo de hierarquias de generalização/especialização.

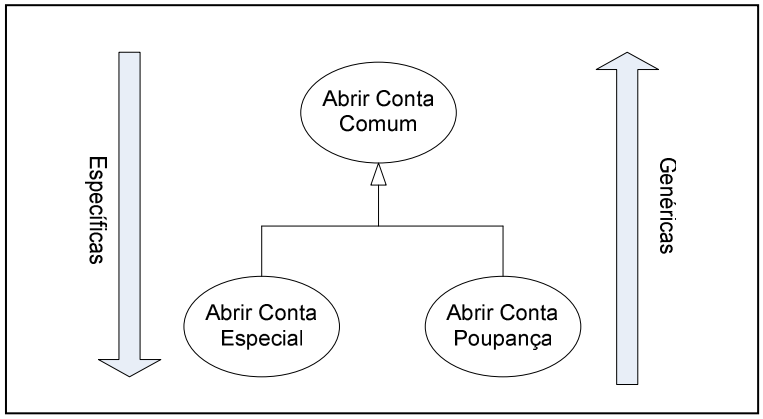


Figura 21: HIERARQUIAS DE ESPECIALIZAÇÃO/GENERALIZAÇÃO
Fonte: Guedes (2009)

Na Figura 21 há a relação de hierarquia onde as duas classes mais abaixo herdam características da classe mais acima, e especializam-se em alguma coisa que as difere das demais, enquanto que a classe acima tem suas características próprias e está compartilhando estas características com as que estão abaixo dela. Dessa forma na relação de hierarquia as classes mais embaixo são classes mais específicas e a mais acima são mais genéricas.

Nos Quadros 9 e 10 Breitman (2005) resume as classes essenciais do RDF Schema e também os relacionamentos.

Quadro 9: CLASSES ESSENCIAIS DO RDF-S, (BREITMAN 2005)

Classe	Descrição
rdfs: Resource	Classe de todos os recursos.
rdfs: Class	Classe de todas as classes.
rdfs: Literal	Classe de todos os literais.
rdfs: Property	Classe de todas as propriedades

QUADRO 10: RELACIONAMENTOS, (BREITMAN, 2005)

rdfs: subClassof	Define um relacionamento de herança entre duas classes.
rdfs: subPropertyof	Define um relacionamento de herança entre duas propriedades.

Um exemplo da utilização da linguagem RDF-S é apresentado a seguir.

```
1  :Abrir_Conta_Comum rdf:type owl:Class .
2  :Abrir_Conta_Especial rdf:type owl:Class ;
3       rdfs:subClassOf :Abrir_Conta_Comum .
4  :Abrir_Conta_Poupanca rdf:type owl:Class ;
5       rdfs:subClassOf :Abrir_Conta_Comum .
```

Na linha 1 do exemplo apresentado é criada uma classe geral chamada Abrir Conta Comum e nas linhas 2 a 5 são criadas e especificadas classes que possuem características da primeira classe, mas que possuem alguma característica que justifica sua criação.

Segundo Breitman (2005) utiliza-se o RDF-S em conjunção com o RDF, onde o RDF-S pode ser considerado um tipo de dicionário que pode ser lido por máquinas. Mais detalhes da linguagem RDF-S são apresentados pelos autores Brickley e Guha (2003).

McGuinness e Harmelen (2004) fazem a seguinte definição para a linguagem OWL: é uma linguagem que foi projetada para facilitar o processamento de informações por máquinas, pois fornece vocabulário adicional com uma semântica formal.

De acordo McGuinness e Harmelen (2004) a OWL pode ser utilizada para representar explicitamente o significado dos termos em vocabulários e as relações entre estes termos. Esta representação dos termos e suas inter-relações são chamadas de ontologia.

Segundo Linhalis (2010) a linguagem OWL estende o vocabulário de RDFS com recursos adicionais que podem ser usados para construir ontologias mais expressivas. Entre os elementos presentes nesta linguagem estão: cabeçalho da ontologia, anotações, classes e indivíduos, definição e uso de propriedades, descrição das propriedades, afirmação negativa da propriedade, restrições de propriedades, descrições avançadas de classes e equivalência. Mais detalhes da linguagem OWL é apresentado pelos autores McGuinness e Harmelen (2004).

O código apresentado a seguir mostra um exemplo da linguagem OWL associada com a linguagem RDF e RDF-S.

```
:Value rdf:type owl:Class ;
       rdfs:subClassOf :Data .
```

A interpretação dessa codificação seria que Value é uma classe e que esta classe é uma sub-classe de Data. Mais detalhes sobre codificação de ontologias são apresentadas no Capítulo 5 e nos Apêndices 1 e 2.

Para tornar a visualização da codificação produzida a partir da OWL mais intuitiva é possível utilizar serializações, como, por exemplo, a serialização[21] RDF chamada Turtle. Segundo Beckett e Berners-Lee (2008) Turtle permite que grafos RDF possam ser totalmente escritos em um texto de forma natural e compacta, com abreviações para os padrões de uso comum e tipos de dados.

O formato de serialização Turtle foi adotado por ser bem intuitivo para leitores humanos. Para exemplificar o quanto a serialização Turtle torna mais intuitiva a codificação é apresentada abaixo a definição de uma classe serializada em RDF/XML.

```
1  <owl:Class rdf:about="&mineracao;Source">
2          <rdfs:subClassOf rdf:resource="&metadm;Data"/>
3  </owl:Class>
```

Agora é apresentado o mesmo código na serialização Turtle.

```
1  :Source rdf:type owl:Class ;
2         rdfs:subClassOf :Data .
```

[21] Segundo Linhalis (2009) A serialização fornece uma maneira de converter o modelo abstrato para um formato concreto, tal como um arquivo ou stream de bytes.

Estes códigos apresentados têm um mesmo propósito que é definir uma classe e defini-la como sub-classe de outra. Porém o segundo exemplo torna bem mais fácil a leitura e entendimento do código. Mais detalhes de Turtle podem ser obtidos em Beckett e Berners-Lee (2008).

3.6 Ferramentas para Desenvolvimento e Visualização de Ontologias

Ferramentas para o desenvolvimento de ontologias são bem convenientes, pois agilizam a formalização de determinados domínios e dão suporte para a sua codificação em diversas linguagens. Segundo estudos realizados por Linhalis (2007), os fatores que influenciam na escolha de uma ferramenta de desenvolvimento de ontologias são: facilidade de uso, entendimento intuitivo da interface, visibilidade da ontologia, interfaces gráficas, conexão a repositórios, portabilidade, organização dos arquivos gerados, documentação de alterações, suporte a trabalho cooperativo, extensibilidade e ferramentas de apoio que facilitam o desenvolvimento.

Dentre as ferramentas para o desenvolvimento de ontologias que merecem destaque, são citadas as seguintes: o Servidor Ontolingua[22], o Ambiente WebODE[23], o OntoEdit[24], o Pacote de Ferramentas KAON[25], o Protégé[26] e o Swoop[27].

O Quadro 11 apresenta as principais características de cada uma das ferramentas citadas no parágrafo anterior.

[22] http://www.ksl.stanford.edu/software/ontolingua/. Acesso em: 03 jun. 2010
[23] http://webode.dia.fi.upm.es/WebODEWeb/index.html. Acesso em: 03 jun. 2010
[24] http://www.ontoknowledge.org/tools/ontoedit.shtml. Acesso em: 03 jun. 2010
[25] http://kaon.semanticweb.org/. Acesso em: 03 jun. 2010
[26] http://protege.stanford.edu/. Acesso em: 03 jun. 2010
[27] http://code.google.com/p/swoop/. Acesso em: 03 jun. 2010

Quadro 11: CARACTERÍSTICAS DAS FERRAMENTAS PARA A CRIAÇÃO DE ONTOLOGIAS, LINHALIS (2007)

Ferramentas	Características
Servidor Ontolingua	• Foi a primeira ferramenta criada para trabalhar especificamente com ontologias; • No início, o principal módulo era o editor de ontologias, depois foram acrescentados outros módulos como o OKBC (*Open Knowledge Based Connectivity*) e Chimaera[28]; • Foi criado para trabalhar com a linguagem Ontolingua[29].
Ambiente WebODE	• É o sucessor do ODE (*Ontology Design Environment*); • A base do ambiente WebODE é o serviço de acesso à ontologia, o qual é utilizado por todas as aplicações cliente conectadas ao servidor, especialmente pelo editor de ontologias; • Há serviços para importar e exportar linguagens (XML, RDF(S), OIL, DAML+OIL, F-Logic, JESS, Prolog), editar axiomas, documentar, avaliar e fazer *merging* de ontologias; • O ambiente apóia a maioria das atividades do processo de desenvolvimento de ontologias proposto por METHONTOLOGY; mas pode também ser usado com outras metodologias ou com nenhuma metodologia.
OntoEdit	• É similar ao WebODE; • Além dos serviços fornecidos pela ferramenta WebODE também suporta: edição, navegação, exportação e importação de diferentes formatos; • Há duas versões de OntoEdit: OntoEdit Free e OntoEdit Professional; • Recentemente, o pacote KAON incorporou OntoEdit.
Pacote de Ferramentas KAON[30]	• Foi desenvolvido pela Universidade de Karlsruhe e pela empresa Ontoprise; • Além de OntoEdit como editor de ontologias, o pacote possui diversas ferramentas comerciais para a aplicação de ontologias em comércio eletrônico, gestão de conhecimento e web semântica; • O projeto KAON trabalha com ontologias em OWL-DL, SWRL e F-Logic.
Protégé	• É uma ferramenta de código aberto com uma arquitetura extensível; • O núcleo de seu ambiente é o editor de ontologias, mas é possível acrescentar mais funcionalidade por meio de uma grande gama de *plugins*, dentre eles importação e exportação de linguagens, criação e

[28] Uma ferramenta para fazer o merging de ontologias
[29] Uma linguagem para implementar ontologias
[30] The KArsruhe ONtology and semantic web tool suite

	execução de restrições; • Pode utilizar as linguagens: OWL, JESS, F-Logic, Prolog, RDF, OIL, XML com a adição de *plugins* apropriados; • Na busca de aumento de usuários passou por várias reengenharias, provendo ferramentas simples e configuráveis; • Pode ser adaptada a diversos usos; • Como consequência de possuir código aberto surgiu uma arquitetura integrável a diversas aplicações, via componentes que podem ser conectados ao sistema.
Swoop	• Possui um navegador de ontologias inspirado em hipermídia e um editor baseado em OWL, que permite extensibilidade por meio de adição de plugins; • É a única ferramenta a trabalhar especificamente com OWL e seu projeto reflete isso; • Contém duas máquinas de inferência adicionais RDF-like e Pellet.

A seguir é apresentado com mais detalhes a ferramenta Protégé, onde a mesma possui uma série de características que permite um rápdido desenvolvimento de ontologias. Além disso, há uma série de materiais que podem ser usados para consultas, plugins que podem ser instalados dentro da ferramenta e suporte para fazer a serialização dos códigos gerados por esta ferrramenta. Dessa forma é bem conveniente o desenvolvimento de ontologias a partir dessa ferramenta.

A versão que será feita uma breve descrição é a 4.1.0. A Figura 22 apresenta uma visão geral dessa ferramenta.

Os principais componentes neste ambiente segundo HORRIDGE et al (2009) são: *Individual, Properties* e *Classes.* No Quadro 12 é feita uma breve descrição de cada um desses itens segundo os autores.

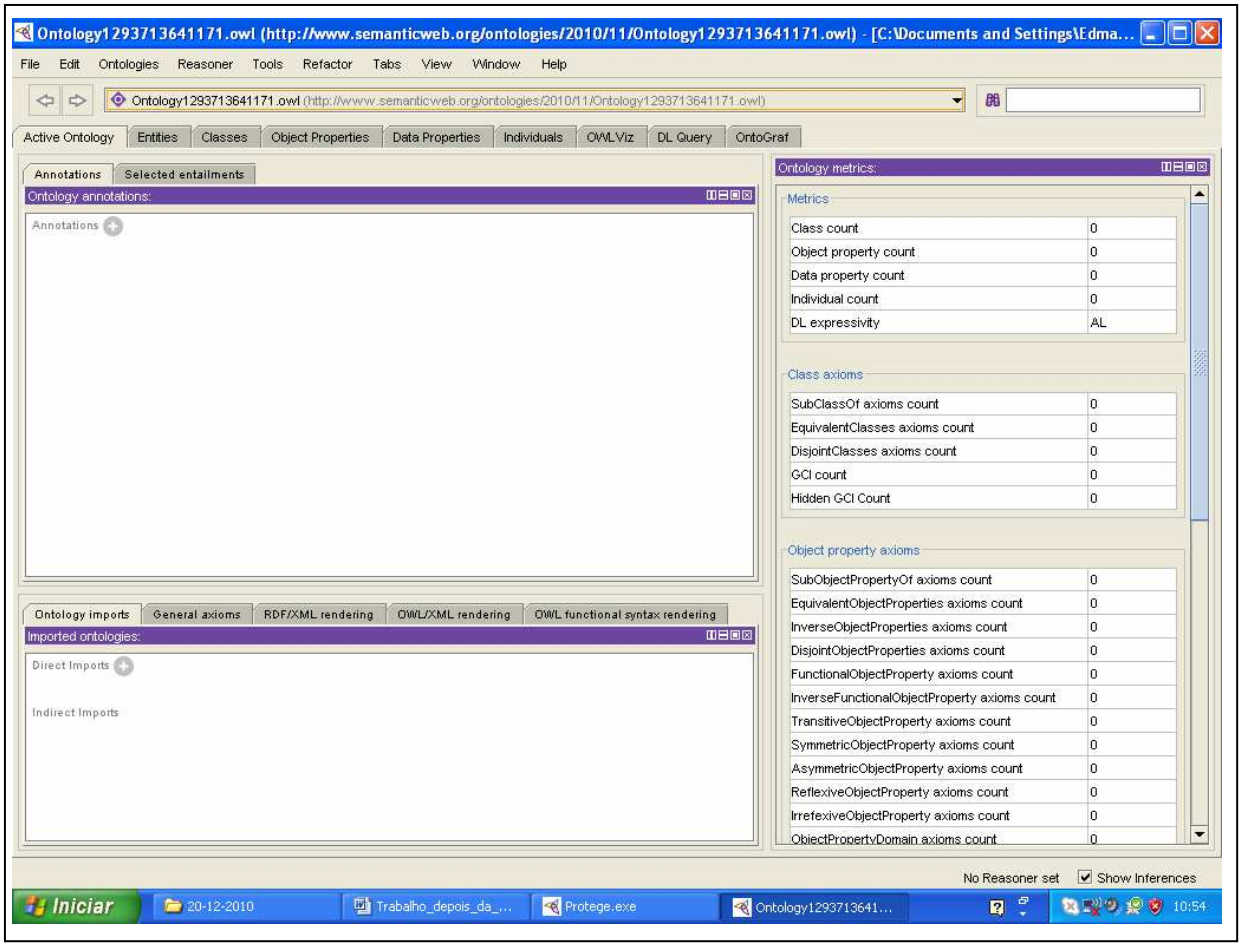


Figura 22: AMBIENTE DE DESENVOLVIMENTO DO PROTÉGÉ 4.1.0
Fonte: Software Protégé

Quadro 12: DESCRIÇÃO DOS COMPONENTES DA ONTOLOGIA OWL, HORRIDGE ET AL. (2009)

Componentes	Descrição
Classes	São conjuntos que contêm indivíduos. Exemplo: A classe gato poderia conter todos os indivíduos que são gatos em seu domínio de interesse.
Properties	São relações binárias entre indivíduos. Exemplo: A ligação entre dois indivíduos.
Individuals	Representa objetos no domínio, também conhecidos como indivíduos ou instâncias.

Estes componentes são implementados no Protégé através das várias abas que compõe a ferramenta, conforme mostra a Figura 22. Entre os items a serem implementados para a construção de uma ontologia estão:

- *Classes*: Para definir as classes da ontologia proposta;
- *Object Properties:* Para definir as relações entre as classes;

- *Data Properties:* Para definir os atributos pertencentes a determinadas classes;
- *Individuals:* Para definir a instancias das classes; e
- *OWLViz:* Para a vizualização da ontologia gerada.

Para a visualização gráfica de uma ontolgia pode ser utilizado o visualizador OwlViz[31], onde é possível ver graficamente todas as classes e relacionamentos criado durante o processo de desenvolvimento.

Outro item a ser destacado são raciocinadores e máquinas de inferência[32]. Segundo Martimiano (2006) máquinas de inferência normalmente oferecem diferentes serviços de consultas que podem ser utilizados para questionar uma ontologia.

Entre as máquinas de inferência existentes há a Pellet[33], onde a mesma pode ser utilizada para verificar se uma ontologia possui inconsistência ou não.

A máquina de inferência Pellet pode ser instalada dentro da ferramenta Protégé por meio de um plugin.

3.7 Considerações Finais

Este capítulo fez uma introdução sobre alguns conceitos relacionados a ontologias. As subseções 3.2 e 3.3 tiveram como objetivo fazer uma conceituação sobre este tema. A subseção 3.4, apresentou metodologias para criar ontologias, em especial a metodologia METHONTOLOGY onde, a partir de seus conceitos, foi documentado o ciclo de vida da ontologia de domínio proposto neste trabalho e a metodologia de Noy e McGuiness onde, a partir de seus passos foram definidos os conceitos, propriedades e relacionamentos que devem constar na ontologia proposta. Na subseção 3.5 foram abordadas algumas linguagens para implementação de ontologias, em especial a linguagem OWL que foi utilizada para a

[31] http://www.co-ode.org/downloads/owlviz/
[32] Segundo Martiniano (2006) Máquina de Inferência é um programa que possibilita a geração de hipóteses a partir das informações na base de conhecimento
[33] Segundo Martiniano (2006) Pellet é uma máquina de inferência que possui uma implementação completa para linguagem OWL/DL, incluindo análise das instâncias da ontologia, dos tipos de dados e das restrições.

implementação da ontologia proposta. E por fim na seção 3.6 são abordadas ferramentas para o desenvolvimento de ontologias, onde no caso deste projeto foi utilizada a ferramenta Protégé.

O próximo capítulo apresenta o ciclo de vida da ontologia proposta, feito com base no levantamento bibliográfico desse Capítulo, sobre o domínio de conhecimento levantado no Capítulo 2.

4 DESENVOLVIMENTO DE UMA ONTOLOGIA PARA O DOMÍNIO DA MINERAÇÃO DE DADOS

4.1 Considerações Iniciais

Conforme apresentado nas seções anteriores uma ontologia é uma forma de compartilhar informações e reusar conhecimento sobre um domínio específico. Uma boa maneira de permitir que terceiros entendam a proposta da ontologia criada é ter documentos que registram todo o processo de desenvolvimento da ontologia.

Fernández et. al. (1997) ressaltam que muitos desenvolvedores de ontologias não documentam o que foi feito, provocando dificuldades de terceiros no entendimento e reuso da ontologia. Assim, quanto mais detalhada for a documentação de desenvolvimento de uma ontologia melhor será para seu entendimento, manutenção e crescimento.

Conforme Martimiano (2006) uma ontologia evolui constantemente, principalmente para atender a mudanças ocorridas com o domínio de conhecimento que ela representa.

O objetivo deste capítulo é documentar a ontologia desenvolvida, onde é expresso a sua motivação e seu processo de desenvolvimento. Assim é possível, por meios dos documentos criados, ter uma melhor compreensão do que foi feito, permitindo um melhor suporte para uma possível manutenção e crescimento da ontologia Meta-DM.

4.2 Documentação do Ciclo de Vida da Ontologia Meta-DM

A ontologia Meta-DM, além de ter o propósito de guiar o processo de mineração de dados (onde é identificada a necessidade de conhecimento humano), têm como propósito adicional poder ser re-utilizada por outros desenvolvedores de ontologias e sofrer constantes alterações de acordo com novos propostos que possam ser adicionados. Assim foi necessário utilizar alguma metodologia de

desenvolvimento de ontologias, que pudesse registrar cada uma das fases de desenvolvimento da ontologia para então ter documentos suficientes, que facilitasse a realização de possíveis operações sobre a ontologia.

Nesta seção é apresentada a documentação do ciclo de vida da ontologia, onde foi utilizada a metodologia de Fernández et. al. (1997), também conhecida como METHONTOLOGY, para documentar cada uma das etapas. A metodologia de Noy e McGuiness (2001) foi utilizada na fase de formalização.

Nas próximas seções são apresentadas as fases da metodologia METHONTOLOGY e sua aplicação no desenvolvimento da ontologia Meta-DM.

4.2.1 Especificação

Nessa etapa são apresentados os levantamentos iniciais para o desenvolvimento da ontologia, onde foram abordadas questões sobre o domínio da ontologia; seus objetivos, usuários, tarefas a serem realizadas e os recursos necessários.

- Definição do domínio: Processo de descoberta de conhecimento em bases de dados (Mineração de Dados);
- Definição do objetivo principal: criar uma ontologia para guiar o processo de mineração de dados e para servir de terminologia comum para ferramentas de mineração de dados em geral. Também procurou-se identificar a necessidade de intervenção humana em algumas etapas da mineração de dados;
- Definição dos usuários: Os potenciais usuários da ontologia são os mineradores de dados, as ferramentas de mineração de dados e possíveis interessados no desenvolvimento de ontologias para o domínio da mineração de dados;
- Definição das tarefas: As principais tarefas realizadas para o desenvolvimento da ontologia consistem em seguir as metodologias de Noy e McGuinnes e METHONTOLOGY, conforme apresentado no início da seção 4.2.

- Definição dos recursos: Os recursos necessários para o desenvolvimento da ontologia são: ferramenta computacional para modelar a ontologia, linguagem para formalizá-la, e recursos humanos para desenvolvê-la, o que inclui um especialista no domínio de mineração de dados.

4.2.2 Aquisição de conhecimento

Após ter definido o domínio e os objetivos da ontologia, foi necessário fazer a aquisição de conhecimento necessários para o desenvolvimento da ontologia proposta.

Para adquirir conhecimento a respeito do domínio de mineração de dados, foram consultados livros, artigos, tutoriais, ferramentas de mineração de dados e especialistas da área.

Esta etapa foi realizada constantemente durante a elaboração do ciclo de vida da ontologia, pois conforme ocorre o amadurecimento da ontologia é verificada a necessidade de buscar outras definições.

4.2.3 Conceituação

Após as fases de Especificação e Aquisição de Conhecimento foi iniciada a conceituação da ontologia, onde foi adotada a metodologia de Noy e McGuiness (2001) para definir as classes e relacionamentos que iriam ser representados na ontologia de domínio. Vale ressaltar que esta etapa passou por constantes mudanças conforme surgiam novas necessidades e outros conceitos necessitavam ser abordados. Como mencionado antes, o processo de desenvolvimento de uma ontologia é dinâmico.

A ideia base utilizada para definir os conceitos da ontologia constitui em seguir as etapas para o processo de mineração de dados conforme definido na metodologia CRISP-DM:

- Definir o objetivo e o problema do projeto de mineração de dados;

- Fazer um entendimento inicial dos dados;
- Preparar os dados;
- Definir uma tarefa de mineração de dados;
- Gerar um padrão;
- E por último interpretar os padrões gerados.

Na Figura 23 são apresentados os conceitos identificados na forma de um diagrama que representa a ontologia.

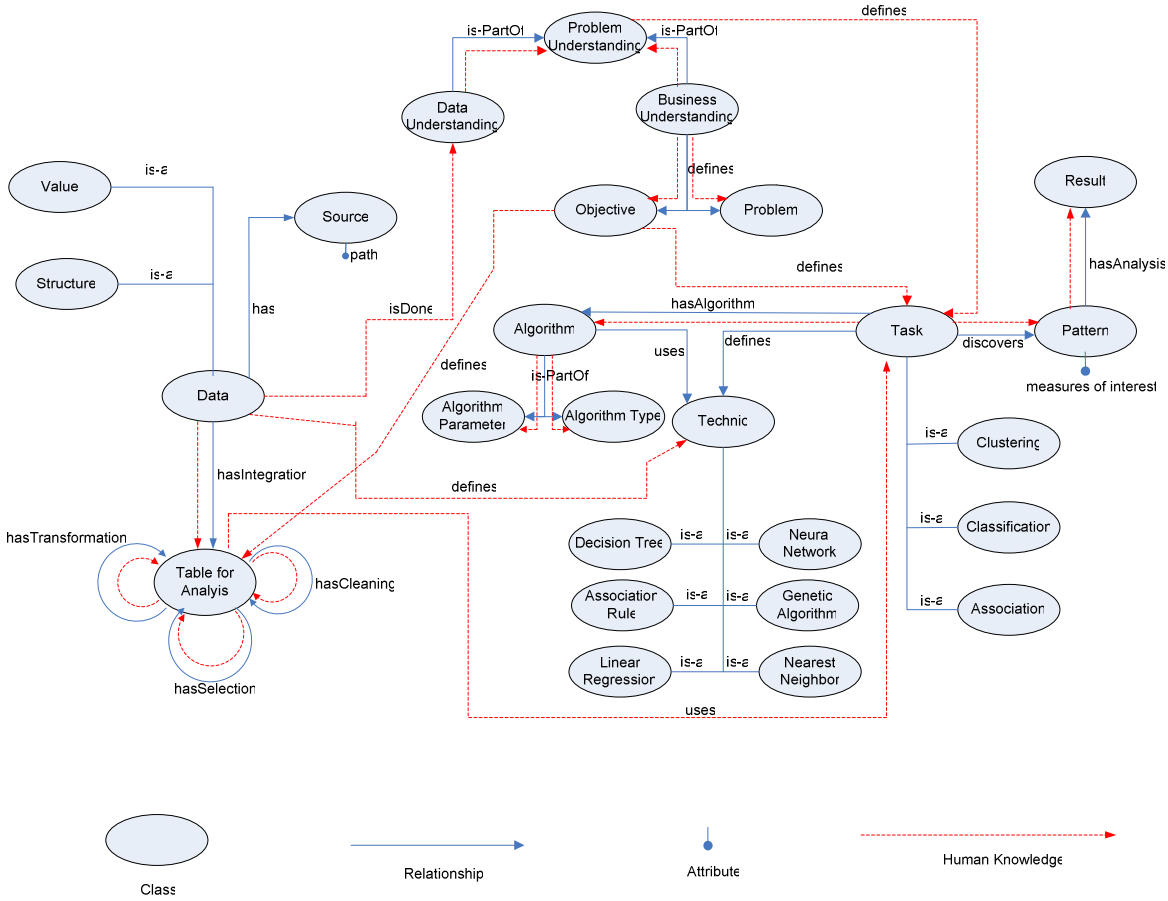


Figura 23: DIAGRAMA DA ONTOLOGIA NO SEU MAIS ALTO NÍVEL

Este diagrama é constituído por quatro elementos que são: as classes, os relacionamentos entre as classes, os atributos e os relacionamentos entre classes onde é necessário conhecimento humano. Há também na ontologia de domínio partes que estão representados ao mesmo tempo por setas azuis e vermelhas tracejadas, onde há tanto a necessidade de representar a ligação entre as classes como também a necessidade de representar conhecimento humano.

Os Quadros 13, 14 e 15 fazem uma breve descrição de cada um desses elementos identificados na elaboração da ontologia.

Quadro 13: DICIONÁRIO DE DADOS DAS CLASSES

N°	**Classe**	**Significado**
01	Data	Classe responsável por organizar as classes referentes aos dados a serem minerados.
02	Value	Representa os dados a serem minerados.
03	Source	Refere-se ao caminho da fonte da base de dados.
04	Structure	Contém a estrutura da base de dados a ser minerada.
05	Table for Analyis	Responsável por abrigar os dados que sofrerão alteração para realizar a mineração de dados. Isto é, são os dados que serão minerados.
06	Problem Understanding	Classe responsável por organizar as classes “Data Understanding e Business Understanding”.
07	Data Understanding	Consiste no entendimento inicial da base de dados a ser minerada.
08	Business Understanding	Consiste no entendimento do que a mineração de dados se propõe a resolver sob uma perspectiva de negócio.
09	Objetive	Define o objetivo a ser alcançado com o processo de mineração de dados.
10	Problem	Define o problema que a mineração de dados deve solucionar.
11	Task	Define a tarefa de mineração de dados.
12	Association	É uma tarefa de mineração de dados. Associa campos de uma base de dados.
13	Classification	É uma tarefa de mineração de dados. Classifica campos de uma base de dados, conforme rótulos pré-estabelecidos.
14	Clustering	É uma tarefa de mineração de dados. Agrupa campos de uma base de dados, conforme um critério pré-estabelecido pelo usuário.
15	Technic	Define a técnica de mineração de dados a ser adotada.
16	Decision Tree	É uma técnica de mineração de dados que utiliza árvore de decisão.
17	Association Rule	É uma técnica de mineração de dados que utiliza relacionamentos entre os elementos de uma base de dados.
18	Linear Regression	É uma técnica de mineração de dados que utiliza valores contínuos.

19	Neural Network	É uma técnica de mineração de dados que utiliza redes neurais.
20	Genetic Algorithm	É uma técnica de mineração de dados que utiliza algoritmos genéticos.
21	Nearest Neighbor	É uma técnica de mineração de dados que utiliza os valores próximos de um dado para definir seu valor.
22	Algorithm	Algoritmo selecionado para a tarefa de mineração de dados.
23	Algorithm Parameter	Parâmetros que devem ser definidos conforme o algoritmo selecionado.
24	Algorithm Type	Identificação do tipo de algoritmo, que pode pertencer a uma das tarefas de mineração de dados (associação, classificação e agrupamento).
25	Pattern	Gera um determinado padrão conforme as técnicas aplicadas.
26	Result	Contém uma descrição da análise feita sobre os padrões encontrados com a mineração de dados.

Quadro 14: DICIONÁRIO DE DADOS DOS RELACIONAMENTOS

Nº	**Relacionamento**	**Significado**
01	is-a	Indica que uma classe é subclasse de uma outra.
02	Defines	Indica que uma classe define um determinado conceito de uma outra classe.
03	is-PartOf	Indica que uma classe é parte ou extensão de uma outra.
04	hasAlgorithm	Indica que uma classe obtém um determinado tipo de algoritmo de uma outra classe.
05	Discovers	Indica que uma classe descobre um determinado padrão a partir de uma outra classe.
06	isDone	Indica que uma determinada operação será realizada sobre os dados.
07	hasValue	Indica que uma classe obtém valores de uma outra classe.
08	hasColumn	Indica que uma classe contém colunas de uma outra classe.
09	hasIntegration	Indica que uma tabela ou mais tabelas de dados são integradas a uma determinada tabela.
10	hasTransformation	Indica que um conjunto de dados é transformado para uma melhor adequação à realização da mineração de dados.
11	hasSelection	Indica que um conjunto de dados é selecionado para a realização da mineração de dados.
12	hasCleaning	Indica que a partir de um conjunto de dados analisados é feito tratamentos, conforme a necessidade de adequação dos dados para a mineração de dados.

13	Uses	Indica que uma classe utiliza os elementos de uma outra classe.
14	hasAnalysis	Indica que é feita uma análise do que foi gerado em uma determinada classe.

Quadro 15: DICIONÁRIO DE DADOS DOS ATRIBUTOS

N°	**Atributo**	**Significado**
01	Path	Contém o caminho de uma base de dados.
02	Measures of interest	Medidas de interesses de acordo com os resultados geradas com a aplicação da mineração de dados.

Além dos elementos descritos nos quadros, há um outro elemento que se encontra na cor vermelha e tracejada que indica os pontos em que há a necessidade de inserção de conhecimento humano para realizar o processo de mineração de dados. O Capítulo 5 aborda esse assunto com mais detalhes.

Para uma melhor descrição da ontologia desenvolvida, apresentado na Figura 23, à mesma pode ser particionada. Dessa forma os próximos parágrafos apresentam a ontologia dividida em partes, conforme as principais atividades realizadas.

A Figura 24 apresenta a parte da ontologia referente aos dados.

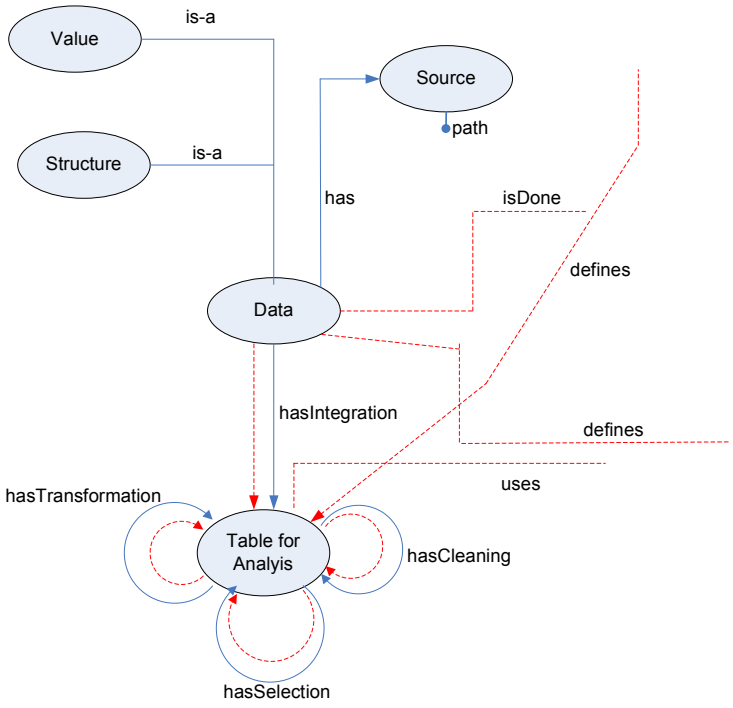


Figura 24: CLASSE DATA

Esta parte refere-se à obtenção e ao tratamento dos dados a serem minerados, onde são considerados: a estrutura da base de dados constituída pela classe Structure; o caminho da base de dados representado pela classe Source; os dados armazenados na base de dados representados pela classe Value. A classe Table for Analyis é responsável por representar a tabela que contém os dados prontos para serem minerados, porém para que esta tabela seja gerada é necessário que ocorra às ações: hasIntegration, hasTransformation, hasCleaning e hasSelection, que são os relacionamento de classes que representam a fase de pré-processamento dos dados. Todas estas informações estão organizadas pela classe Data que serve como uma classe abstrata para as demais.

Há ainda os relacionamento isDone, uses e defines com seta em vermelho pontilhado que aparece sem relacionamento que indica que existe uma ligação com outras classes da ontologia de domínio. O primeiro relacionamento indica que a partir da classe Data é possível fazer o entendimento dos dados, o segundo relacionamento indica que os dados prontos para ser minerados são utilizados pela a classe Task e o relacionamento defines indica que a partir da classe Data é possível definir qual é a técnica de mineração de dados mais adequadada.

A Figura 25 apresenta parte da ontologia responsável pela definição do entendimento do problema da mineração de dados.

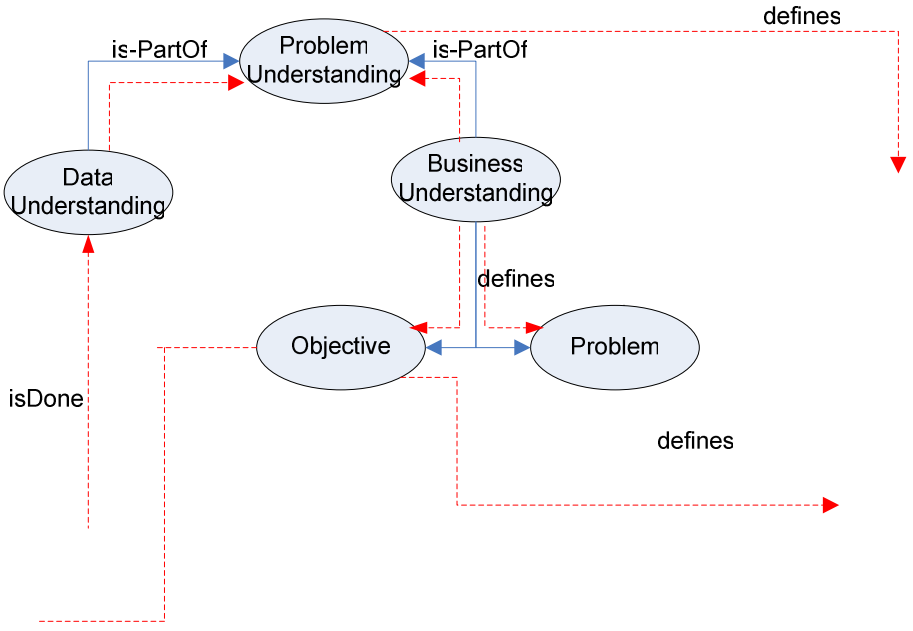


Figura 25: ENTENDIMENTO DO PROBLEMA

Esta parte é constituída pela classe Data Understanding que é responsável por representar o entendimento dos dados a serem minerados, e pela classe Business Understanding que é responsável por representar o entendimento do negócio. A partir dessa última classe são definidos o objetivo e o problema do projeto da mineração de dados. Por fim, a classe Problem Understanding serve como uma classe abstrata para definir o entendimento do problema a partir das classes Data Understaning e Business Understaning.

Nesta seção vale ressaltar que o relacionamento is-PartOf indica que as classes Data Understanding e Business Understanding fazem parte da classe Problem Understanding. A relação defines indica que com o entendimento do negócio é possível definir o objetivo e o problema de um projeto de mineração de dados.

Nesta parte da ontologia há dois relacionamentos defines que saem das classes Problem Understanding e objective e fazem o relacionamento dessas classes com a classe Task. No primeiro relacionamento é indicado que a partir do ententimento do negócio é possível definir uma tarefa de mineração de dados e no segundo relacionamento é indicado que com a definição do objetivo é possível definir a tarefa de mineração de dados mais adequada.

Por fim há ainda um relacionamento defines que sai da classe objective que indica que a partir da definição do objetivo da mineração de dados é possível preparar os dados para a mineração de dados.

Na Figura 26 é apresentada a fase de processamento dos dados para a obtenção de um determinado padrão.

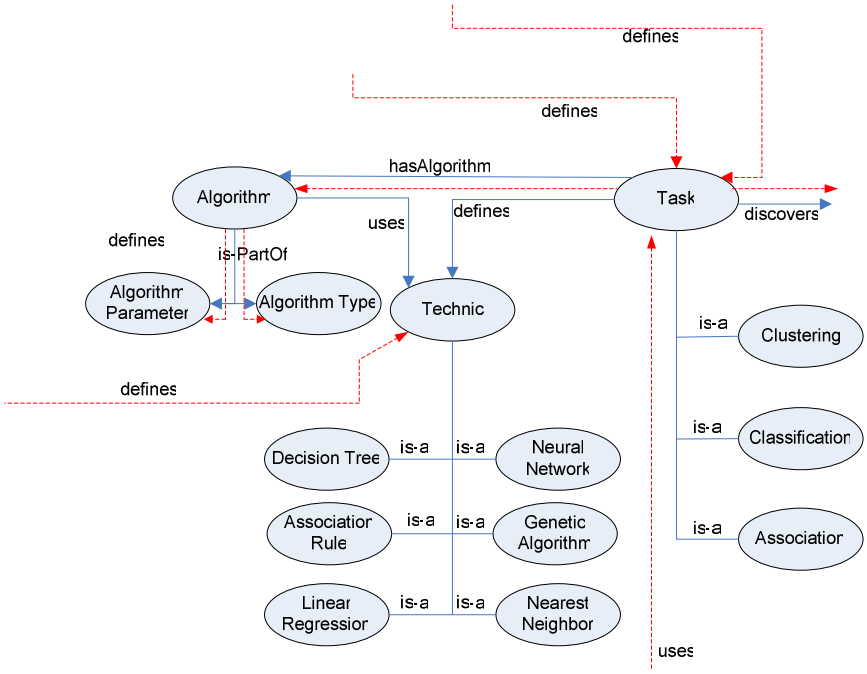


Figura 26: PROCESSAMENTO DOS DADOS

Esta parte da ontologia tem três classes essenciais para que ocorra o processo de mineração de dados. A primeira delas é a classe Algorithm, que representa o algoritmo de mineração de dados a ser utilizado, nesta classe há ainda duas subclasses onde são definidos os parâmetros do algoritmo e o tipo de algoritmo a ser utilizado. A classe Technic representa a técnica de mineração de dados a ser utilizada, que pode ser: árvore de decisão, regras de associação, regressão linear, redes neurais, algoritmos genéticos ou o vizinho mais próximo. Vale ressaltar que foram definidas apenas as técnicas base da mineração de dados. Por fim, na classe Task é representado as tarefas de mineração de dados que podem ser: agrupamento, classificação ou associação.

O relacionamento hasAlgorithm define que uma tarefa de mineração de dados terá um algoritmo. O relacionamento defines indica que a técnica a ser utilizada será definida de acordo com a tarefa de mineração de dados. O relacionamento uses indica que um algoritmo usa uma determinada técnica de mineração. O relacionamento is-PartOf indica que as classes Algorithm Parameter e Algorithm Type são parte da classe Algorithm. E por fim o relacionamento is-a indica que árvore de decisão, regras de associação, regressão linear, redes neurais, algoritmo genético e vizinho mais próximo são técnicas e

também que agrupamento, classificação e associação são tarefas de mineração de dados.

Há também três linhas vermelhas tracejadas, que chegam à classe Task. Isso indica que com a definição do entendimento do problema é possível definir a tarefa de mineração de dados mais adequada, com a elaboração do objetivo do projeto de mineração de dados é definida uma tarefa de mineração de dados e os dados prontos para ser minerados são utilizados pela a classe Task.

Um quarto relacionamento chega a classe Technic, que representa que a partir dos dados é possível identificar qual seria a técnica de mineração de dados mais adequada.

Por fim há o relacionamento discovers que sai da classe Task e que indica que a partir da execução dos procedimentos de uma tarefa de mineração de dados são gerados alguns padrões.

Na Figura 27 a fase de pós-processamento é apresentada.

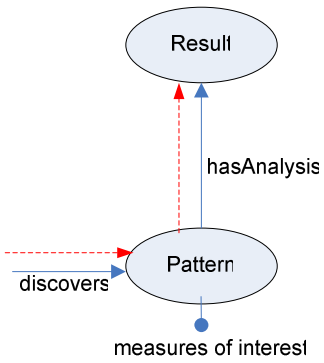


Figura 27: PÓS-PROCESSAMENTO

A fase de pós-processamento é constituída de duas classes: a primeira é Pattern, responsável por representar os padrões gerados a partir da execução do processo de mineração de dados. Nesta classe há o atributo chamado de medidas de interesses que é a porcentagem mínima de valores que deverão ser gerados em uma determinada aplicação de uma tarefa de mineração de dados. A outra classe, Result, representa os resultados definidos pelo o minerador de dados.

Nesta parte da ontologia encontram-se dois relacionamentos: o primeiro, discovers, indica que a partir da classe Task é descoberto um ou vários padrões durante o processo de mineração de dados. O segundo relacionamento, hasAnalysis, define que a partir dos padrões encontrados o minerador de dados pode realizar uma análise dos mesmos.

4.2.4 Integração

O desenvolvimento da ontologia Meta-DM utilizou alguns conceitos de uma ontologia desenvolvida por Pinto e Santos (2009), entre estes conceitos utilizados estão a classe "Data" e as suas subclasses "Source" e "Structure" e também a classe "Algorithm" e as suas subclasses "Algorithm Parameter" e "Algorithm Type".

A reutilização da ontologia desenvolvida por Pinto e Santos (2009) foi feita a partir da disposição das classes mencionadas no parágrafo anterior, onde estas classes foram readaptadas conforme os requisitos levantados para a construção da ontologia. Os relacionamentos foram modificados por não apresentarem o mesmo significado requerido para a ontologia.

Foram feitas análises em outras ontologias desenvolvidas para o domínio da mineração de dados, conforme apresentado na seção 1.5, porém foi constatado que as mesmas não se enquadravam no propósito de desenvolvimento da ontologia proposta neste trabalho.

4.2.5 Implementação

A ontologia foi implementada utilizando a linguagem OWL e a ferramenta Protégé 4.1.

A metodologia de Noy e McGuiness (2001) foi utilizada nesta fase, pois mostra como definir uma ontologia na ferramenta Protégé, utilizando a linguagem OWL.

A implementação foi feita a partir da ontologia apresentada na Figura 23, onde as classes, relacionamentos e atributos foram definidos, graficamente, na Protégé 4.1.

Entre os itens que foram gerados a partir da ferramenta Protégé 4.1 estão:

- Namespaces: Indicação de um conjunto de vocabulários específicos que precisam ser utilizados;
- Classes: Representações das classes da ontologia;

- Object Properties: Relacionamentos entre as classes;
- Data properties: Atributos de uma determinada classe;
- Individuals: Descrição de seus membros;
- General axioms: Representação das classes disjuntas.

A seguir são apresentados alguns exemplos do código que foi gerado para implementar alguns conceitos da ontologia:

O código a seguir mostra o cabeçalho da ontologia:

```
1  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2  @prefix mdm: <http://www.semanticweb.org/ontologies/meta-dm.owl#> .
3  @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
4  @prefix xml: <http://www.w3.org/XML/1998/namespace> .
5  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
6  @prefix owl: <http://www.w3.org/2002/07/owl#> .
```

A linha 1 define o namespace da linguagem RDF para que seja possível utilizar essa linguagem nas declarações da ontologia. Na linha 2 é definido o namespace da ontologia Meta-DM; caso terceiros queiram fazer o reuso da ontologia basta chamar incluir o namespace e usar o prefixo mdm. As linhas 3 a 6 também indicam namespaces para que seja possível utilizar as definições de XML Esquema, XML, RDFS e OWL, respectivamente.

O código a seguir mostra a formalização da classe Column da ontologia.

```
1  :Column rdf:type owl:Class ;
2          rdfs:subClassOf :Structure ;
3          owl:disjointWith :Table .
```

A linha 1 indica que o elemento Column é uma classe, na linha 2 que é uma sub-classe de Structure e na linha 3 que é uma classe disjunta de Table.

O código a seguir mostra a formalização do relacionamento defines, entre as classes Business Understanding, Objective e Problem. Em OWL, esses relacionamentos são implementados com o elemento Object Property.

```
1  :defines rdf:type owl:ObjectProperty ;
2           rdfs:domain :Business_Understanding ;
3           rdfs:range :Objective ,
4                      :Problem ;
5           rdfs:subPropertyOf owl:topObjectProperty .
```

A linha 1 indica que defines é um tipo de Object Property (relacionamento), na linha 2 indica que a classe Business Understanding é o domínio desse relacionamento, nas linhas 3 e 4 indica que as classes Objective e Problem são o range (alcance) dessa relação. Por fim, a linha 5 indica que esse relacionamento é uma sub-propriedade da propriedade objeto, ou seja, um Object Property pai que abriga os demais.

O código a seguir exemplifica o uso de um Data Property (atributos) da ontologia Meta-DM.

```
1  : measures_of_interest rdf:type owl:DatatypeProperty ;
2          rdfs:domain :Pattern ;
3          rdfs:range xsd:string ;
4          rdfs:subPropertyOf owl:topDataProperty .
```

A linha 1 indica que measures_of_interest é um tipo de Datatype Property (atributo), nas linhas 2 e 3 indica que este atributo tem como domínio a classe Pattern, na linha 3 indica o range (alcance) desse atributo é um valor do tipo string. A linha 4 indica que o atributo é uma sub-propriedade de topDataProperty, ou seja, um Datatype Property pai que abriga os demais.

O código a seguir exemplifica a implementação de um Individual (valor) atribuído para a classe Value da ontologia. Ao atribuir valores para a ontologia é possível instanciar a mesma e verificar como as informações se comportam na ontologia.

```
1  :val034 rdf:type :Value ,
2           owl:NamedIndividual ;
```

```
3          :data "Televisao"^^xsd:string .
```

A linha 1 indica que o elemento val034 (um elemento criado para apoiar o processo de instanciação) foi criado para a classe Value, a linha 2 indica que este elemento é um indivíduo e a linha 3 que o valor Televisão, do tipo string, foi atribuído ao indivíduo, que no caso deste exemplo é o val034.

O código a seguir exemplifica a implementação de um General axioms (classes disjuntas, neste caso), atribuído à ontologia:

```
1   [ rdf:type owl:AllDisjointClasses ;
2     owl:members ( :Association
3                   :Classification
4                   :Clustering ) ] .
```

A linha 1 indica que todos os elementos são classes disjuntas, as linhas 2, 3 e 4 indicam que os membros dessa classe disjunta são: Association, Classification e Clustering.

O Apêndice 1 apresenta a ontologia completa formalizada em OWL e serializada em Turtle.

No Anexo 1 deste trabalho é referenciado uma base de dados, onde a mesma possui informações sobre um congresso de tecnologia que aconteceu na cidade de Mococa-SP no ano de 2007. Parte das informações dessa base de dados foi utilizada para fazer a instanciação da ontologia, que é abordada com mais detalhes nos próximos parágrafos.

Durante o processo de implementação da ontologia foi feita também a sua instanciação. No Protégé tem uma aba chamada de Individuals onde são atribuídos valores aos elementos da ontologia. O objetivo de instanciar a ontologia é verificar se os dados ou informação de um domínio estão de acordo com os conceitos, relacionamentos e atributos da ontologia.

O Apêndice 2 apresenta esta instanciação da ontologia para o problema do congresso de acordo com o Anexo 1, também formalizada em OWL e serializada em Turtle, onde foram utilizados alguns registros para instanciar a ontologia.

4.2.6 Avaliação

A avaliação da Meta-DM foi realizada em duas etapas (verificação e validação) conforme a metodologia METHONTOLOGY, onde foi utilizado o Pellet na fase de verificação com o intuito de verificar se a ontologia tinha ou não alguma inconsistência e a validação foi realizada através da instanciação da ontologia. Os parágrafos abaixo fazem uma breve descrição do que foi realizado nestas duas atividades.

O Pellet foi inicialmente instalado no Protégé, através de um plugin disponível pela própria ferramenta. Dessa forma, a verificação da ontologia foi mais fácil, pois ao inserir algum elemento na ontologia basta ir ao menu Reasoner e escolher Pellet para verificar inconsistências. A Figura 28 apresenta esse menu.

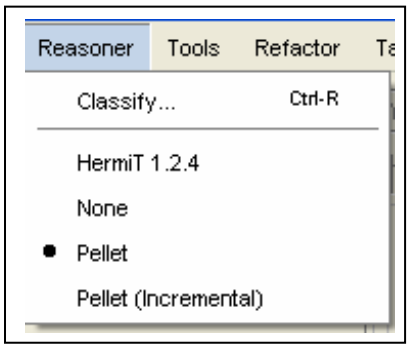


Figura 28: UTILIZAÇÃO DO PELLET DENTRO DO PROTÉGÉ
Fonte: Ferramenta Protégé

O desenvolvimento da ontologia Meta-DM foi realizado em partes conforme apresentado na seção 4.2.3, ao termino de uma das partes então era feita a verificação da ontologia. A Figura 29 apresenta dois conjuntos de elementos, um na cor preta representa os elementos já verificados e outro na cor vermelha que indicada que ainda não foi submetido a verificação.

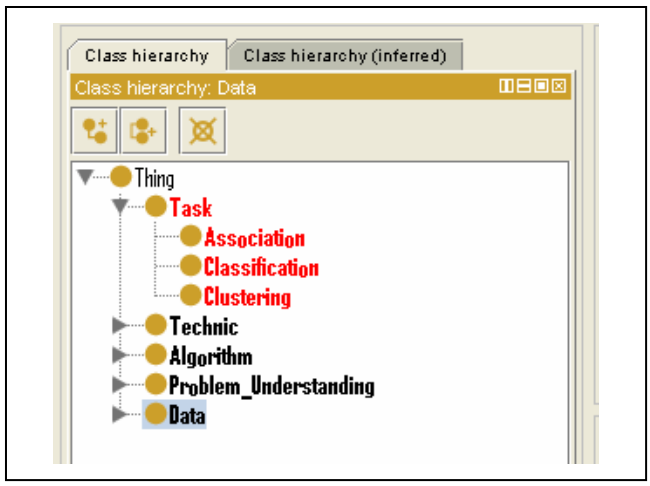


Figura 29: ELEMENTOS VERIFICADOS E NÃO VERIFICADOS
Fonte: Ferramenta Protégé

Após a aplicação da máquina de inferência nas classes inseridas, caso não haja inconsistências as partes em vermelho geradas a partir da inserção de novas classes é mudada para preto, porém caso haja alguma inconsistência é apresentado uma mensagem de erro.

Após o desenvolvimento da ontologia e a verificação de todos os elementos dispostos na ontologia, é possível comprovar que todos os elementos estão declarados de forma correta conforme apresentado na Figura 30.

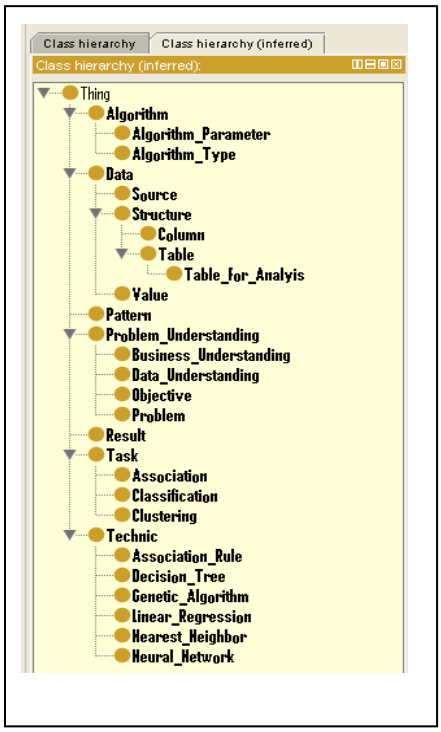


Figura 30: RESULTADO DA VERIFICAÇÃO DO PELLET
Fonte: Ferramenta Protégé

Para uma melhor avaliação da ontologia foi necessário verificar como os dados se comportariam nela durante o processo de mineração de dados, por isso foi realizada a instanciação da ontologia. A ferramenta Protégé permite fazer a instanciação a partir da aba Individuals onde é então possível representar as instâncias de cada classe e os relacionamentos entre elas.

A Figura 31 apresenta um exemplo da instanciação da classe Column da ontologia.

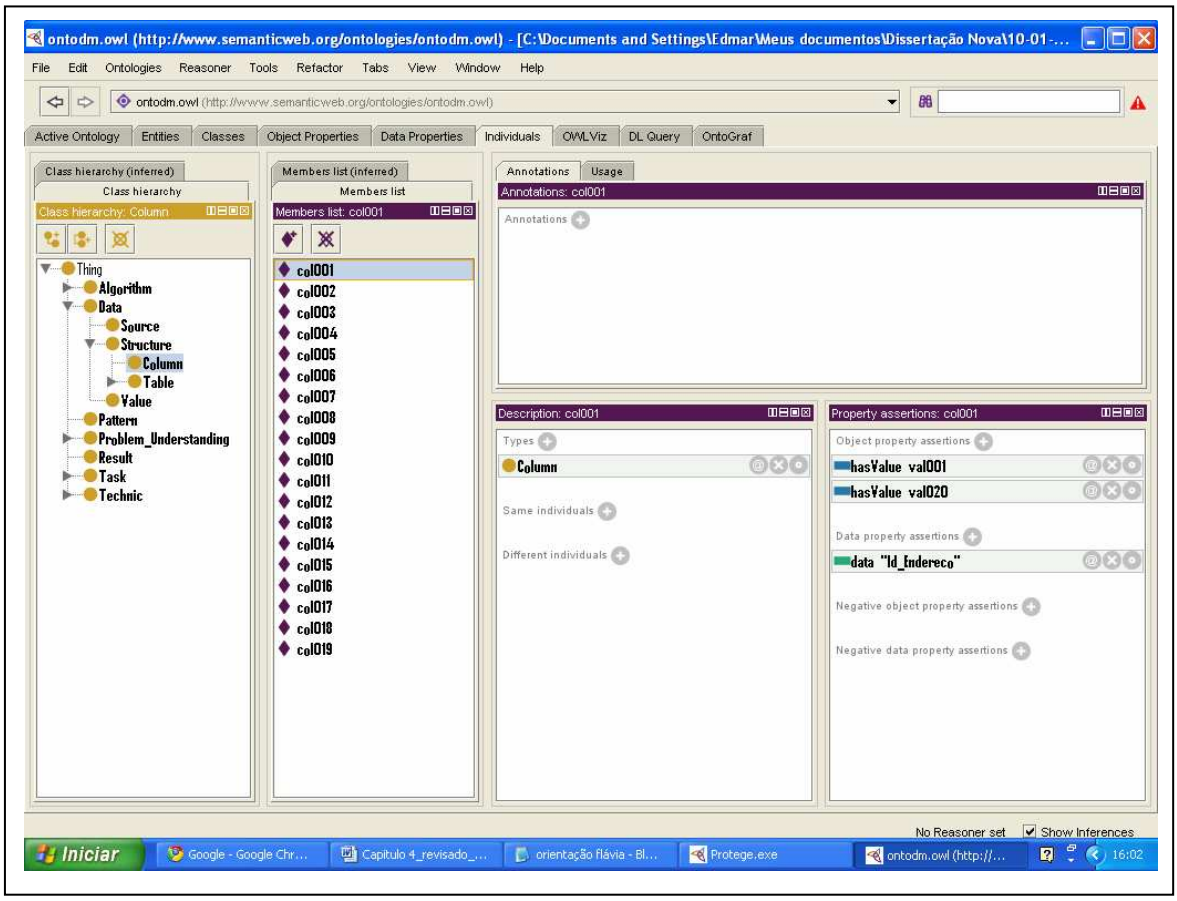


Figura 31: EXEMPLO DE INSTANCIAÇÃO
Fonte: Ferramenta Protégé

No Apêndice 2 é apresentada toda a instanciação da ontologia realizada.

4.2.7 Documentação

Na metodologia METHONTOLOGY (Fernández et al, 1997), a documentação é um processo dinâmico, ou seja, conforme algo mude na construção da ontologia é necessário fazer as adequações na documentação. A fase de documentação deve estar presente em todas as fases anteriores. O processo de documentação da ontologia Meta-DM gerou os seguintes documentos, de acordo com as sub-seções 4.2.1, 4.2.2, 4.2.3, 4.2.4, 4.2.5 e 4.2.6:

- Diagrama da ontologia;
- Dicionário das classes, relacionamentos e atributos;
- Descrição do que foi reutilizado na ontologia;
- A codificação gerada com a implementação da ontologia;
- Instanciação da ontologia
- Documento de avaliação.

4.3 Considerações Finais

Este capítulo descreveu e documentou o processo de criação da ontologia Meta-DM. Para definir as etapas a serem seguidas foi utilizada a metodologia METHONTOLOGY, e para ajudar na fase de formalização foi utilizada a metodologia de Noy e McGuinness. Para fazer a formalização foi utilizada a ferramenta Protégé 4.1, que gera automaticamente o código OWL.

O processo de desenvolvimento da ontologia foi feito através de pesquisas em manuais, artigos, livros, ferramentas de mineração de dados, ontologias já desenvolvidas para este domínio e consulta a especialistas da área, onde eram definidos conceitos e como esses conceitos se encaixavam (ou não) na proposta da ontologia.

O próximo capítulo apresenta tarefas da metodologia D^3M que podem ser inseridas na ontologia, com o intuito de gerar resultados mais eficientes em um projeto de mineração de dados. Também é apresentada uma arquitetura para ferramentas de mineração de dados que utilizam esta ontologia.

5 DEFINIÇÃO DE UMA ARQUITETURA PARA FERRAMENTAS DE MINERAÇÃO DE DADOS COM BASE NA ONTOLOGIA META-DM E NA METODOLOGIA D^{3}M

5.1 Considerações Iniciais

Na ontologia desenvolvida foram detectados alguns pontos onde é essencial o conhecimento humano para que possa ocorrer o processo de mineração de dados. Entre os componentes estão classes e relacionamentos, conforme ilustrado por linhas pontilhadas na Figura 23.

A metodologia D^{3}M foi desenvolvida para melhorar a eficiência no processo de mineração de dados por meio de informação de contexto e interação humana durante a mineração de dados. Neste capítulo é definida uma arquitetura para ferramentas de mineração de dados, onde são inseridas tarefas da metodologia D^{3}M nos pontos onde há necessidade do conhecimento humano, identificados na ontologia Meta-DM.

5.2 Identificação das tarefas da metodologia D^{3}M na ontologia Meta-DM

Esta seção tem como objetivo apresentar as tarefas da metodologia D^{3}M que podem ser inseridas na ontologia, conforme os pontos em que foi identificada a necessidade de conhecimento humano. Para isso foi inserido um retângulo, além dos elementos que pertencem à ontologia (classe, relacionamentos e atributos), para identificar a tarefa da metodologia D^{3}M que pode ser inserida.

A ontologia é constituída por cinco partes principais, que são: dados, entendimento do problema, tratamento dos dados, identificação da tarefa da mineração de dados e padrões gerados. Com base nessas partes é apresentada nas subseções seguintes a ontologia particionada, com as tarefas da metodologia D^{3}M inseridas.

5.2.1 Dados

A parte de Dados da ontologia está relacionada com a base de dados e sua estrutura, que será utilizada em um projeto de mineração de dados. Nessa etapa praticamente não há participação humana, uma vez que, a ontologia representa apenas uma estrutura típica de uma base de dados. Esta parte da ontologia fornece informações e dados para as outras partes, que por sua vez necessitam da intervenção humana, conforme mostram as linhas pontilhadas em vermelho; há também linhas pontilhadas de preto que fazem a separação entre as partes da ontologia, onde as classes de outras partes da ontologia estão diferenciadas por cores distintas.

A Figura 32 apresenta as partes da ontologia onde são identificados pontos onde o conhecimento humano está presente.

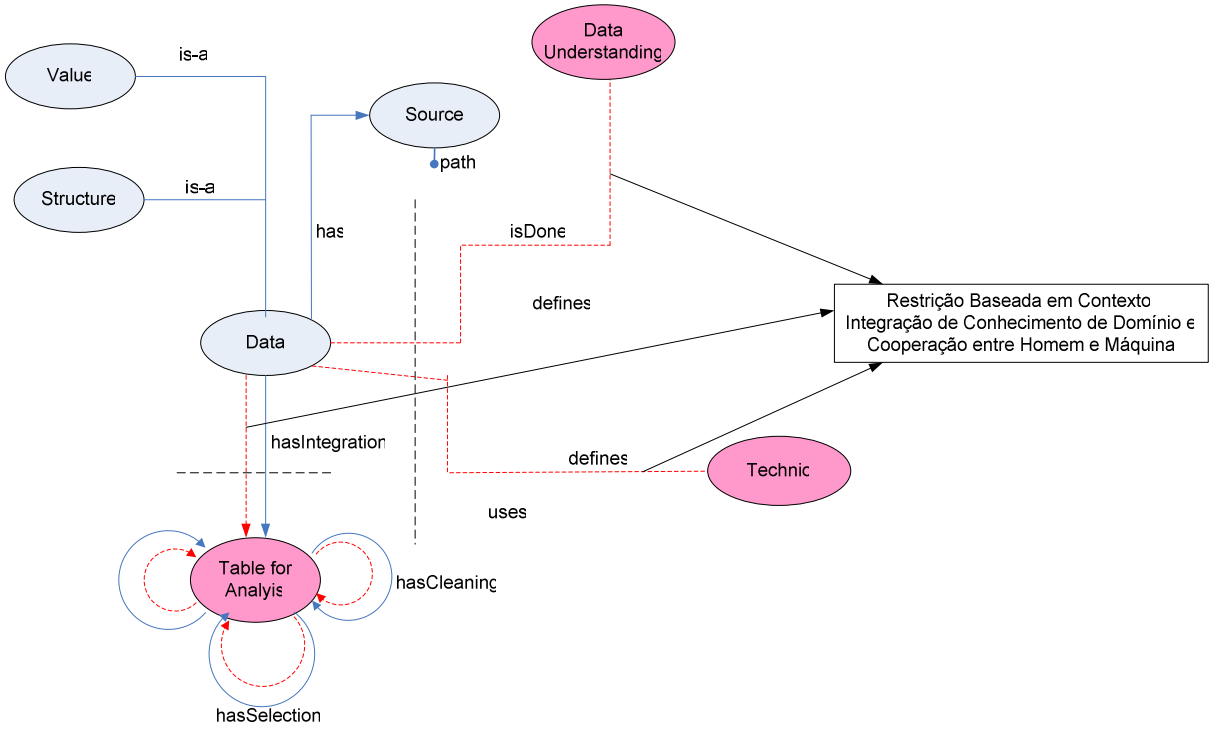


Figura 32: DADOS DA ONTOLOGIA META-DM COM A METODOLOGIA D^{3}M

De acordo com a metodologia D^{3}M a tarefa Restrição Baseada em Contexto do framework DDID-PD pode ser aplicada nos três pontos onde o conhecimento humano se faz necessário. O minerador de dados deve-se restringir ao contexto de domínio do projeto de mineração de dados para então fazer: o entendimento dos dados, pré-definir quais dados são mais convenientes para

trabalhar a mineração de dados e pré-definir qual é a técnica de mineração mais adequada segundo as características dos dados.

Uma outra tarefa da metodologia D^3M indentificada nesta parte da ontologia é Cooperação entre Homem e Máquina, onde o humano poderia através de seu conhecimento de domínio influenciar a execução das tarefas: entendimento dos dados, definição dos dados que serão trabalhados, e definição de qual técnica de mineração de dados é a mais adequada.

Por fim, nesta etapa ainda foi identificado à presença da tarefa Integração de Conhecimento de Domínio, onde para ser feito o entendimento dos dados, definir quais dados são os mais adequados ao projeto de mineração de dados e pré-definir uma técnica de mineração de dados, o minerador de dados poderia ser apoiado por diferentes conhecimentos de domínios levantados por ele e também por ontologias de apoio a este processo.

5.2.2 Entendimento do Problema

Nessa parte da ontologia estão as seguintes classes: Problem Understanding, Data Understanding, Business Understanding, Objetive e Problem, onde estas classes são responsáveis respectivamente por definir o entendimento do problema, o entendimento dos dados e o entendimento do negócio, e também o objetivo e o problema de um projeto de mineração de dados.

Nessa parte da ontologia o minerador tem uma grande influência no processo de mineração de dados, pois cabe a ele fazer um estudo do propósito do projeto de mineração de dados e então caracterizar cada um desses itens. A Figura 33 apresenta a etapa do entendimento do problema da ontologia e as tarefas da metodologia D^3M que podem ser inseridas.

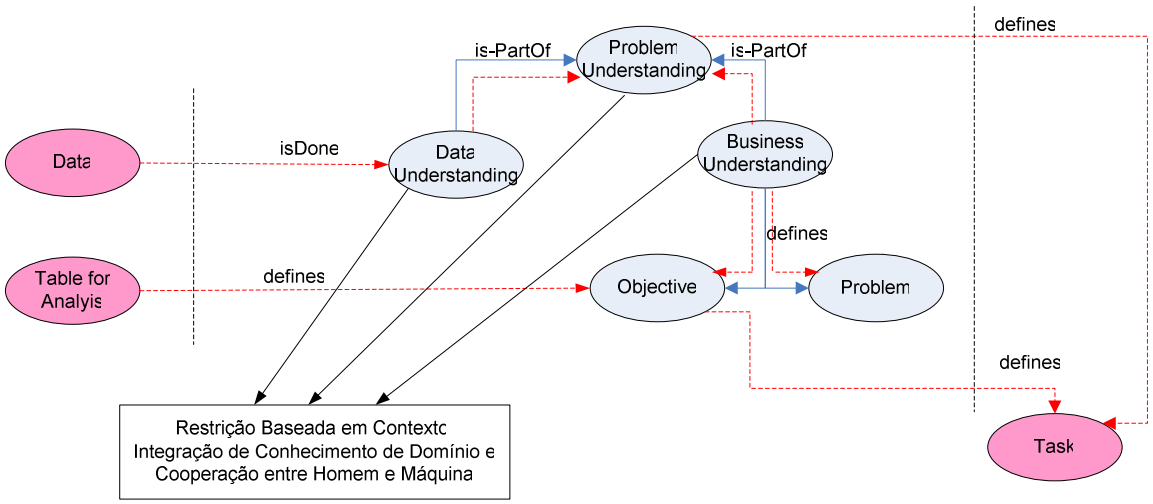


Figura 33: ENTENDIMENTO DO PROBLEMA COM A METODOLOGIA D^{3}M

Nesta parte da ontologia, ao invés de classificar o conhecimento humano de acordo com os relacionamentos, foram classificados de acordo com as classes, uma vez que as classes irão influenciar diretamente a definição da tarefa da metodologia D^{3}M.

Umas das tarefas da metodologia D^{3}M identifcada nesta parte da ontologia é a Restrição Baseada em Contexto, onde para se definir o entendimento do negócio o minerador de dados poderia utilizar uma ontologia específica para este fim. Um exemplo de ontologia que poderia ser utilizada é a de Sharma e Bryson (2008), que define o entendimento do negócio em um projeto de mineração de dados. Na etapa de entendimento dos dados o minerador de dados também deve se restringir ao contexto de acordo com os objetivos e problemas do projeto de mineração de dados e então a partir das tarefas pré-estabelecidas na metodologia CRISP-DM (Chapman et al., 2000) realizar o entendimento dos dados.

Uma outra tarefa da metodologia D^{3}M identificada nesta parte da ontologia foi Cooperação entre Homem e Máquina, onde o minerador de dados através de seu conhecimento de domínio poderia responder a questões essenciais do projeto de mineração de dados, nas etapas de entendimento dos dados e entendimento do negócio, e assim cooperar para a definição do objetivo e problema do projeto de mineração de dados e por fim com base nessas informações levantadas fazer o entendimento do problema.

E na tarefa Integração de Conhecimento de Domínio seria realizado a integração do conhecimento do minerador de dados sobre o projeto, possivelmente por meio de ontologias de domínio.

Nesta subseção não foram descritos dois relacionamentos: isDone, e um relacionamento defines que foi abordado na Seção 5.2.1, e dois relacionamentos defines que são abordado na Seção 5.2.4.

5.2.3 Preparação dos Dados Para a Mineração de Dados

Esta parte da ontologia está relacionada ao tratamento necessário da base de dados em um projeto de mineração de dados. A Figura 34 apresenta a classe responsável por representar os dados para a mineração de dados, os relacionamentos, hasIntegration, hasCleaning, hasSelection e hasTransformation, representam os tratamentos que podem ser feitos, e o retângulo representa as tarefas da metodologia D^{3}M que podem ser inseridas na etapa.

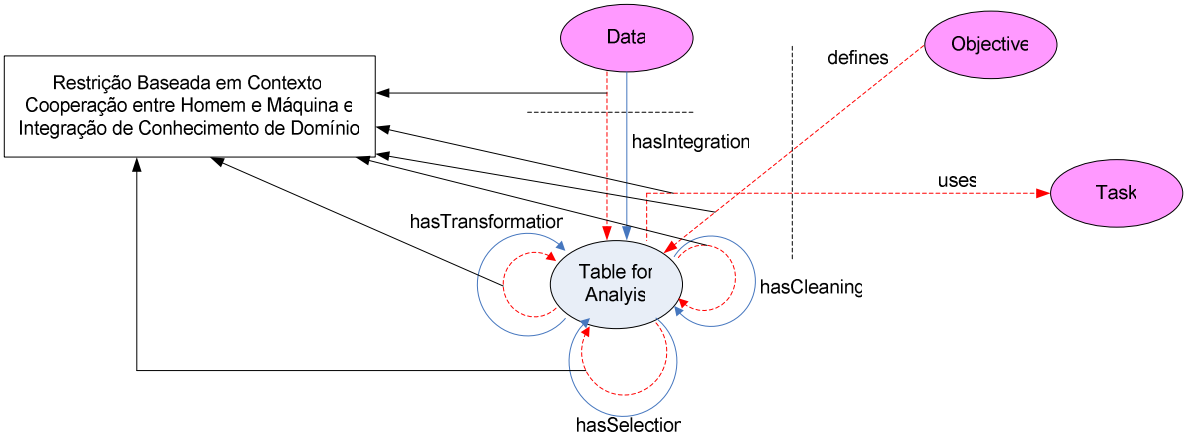


Figura 34: PREPARAÇÃO DOS DADOS COM A METODOLOGIA D^{3}M

Conforme apresentado na Figura 34 foram identificadas três tarefas da metodologia D^{3}M nessa fase. A primeira é Restrição Baseada em Contexto, onde para o minerador de dados integrar a base de dados, selecionar os dados, transformar os dados, limpar ruídos e inconsistências na base de dados deve se restringir ao contexto do problema e então realizar estas operações. A segunda é Cooperação entre Homem e Máquina, onde o minerador de dados por meio do seu conhecimento de domínio do projeto poderia auxiliar o processo de tratamento da base de dados, como, por exemplo, definir regras para realizar tratamentos dos dados. Por fim nesta etapa a tarefa Integração de conhecimento de domínio, poderia

integrar o conhecimento do minerador de dados sobre o projeto com ontologias de apoio a esta etapa.

Além dos tratamentos de dados, nesta parte da ontologia há o relacionamento uses que indica que os dados prontos para a mineração de dados são usados na classe Task. Dessa forma as tarefas da metodologia D^{3}M identificadas nesta etapa teriam as seguintes funções: na tarefa de Restrição de Conhecimento de Domínio, o minerador a partir das restrições das características dos dados seria capaz de definir quais dados seriam utilizados no projeto de mineração de dados, na tarefa Cooperação entre Homem – Máquina o minerador de dados iria definir quais dados seriam utilizado no projeto de mineração de dados, apoiado pelo sistema, e na tarefa Integração de Conhecimento de Domínio haveria a integração do conhecimento do minerador de dados com outros conhecimentos de domínio definido nesta etapa.

O relacionamento defines indica que o objetivo de um projeto de mineração de dados pode ser definido a partir dos dados prontos para a mineração de dados. Dessa forma as tarefas da metodologia D^{3}M identificadas nesta etapa teriam as seguintes funções: a partir da tarefa Restrição de Conhecimento de Domínio o minerador de dados poderia definir o objetivo do projeto de mineração de dados restringindo-se ao contexto de domínio. A partir da tarefa Cooperação entre Homem-Máquina, o sistema de mineração de dados apresentaria as características dos dados prontos para a mineração de dados, onde seria permitido fazer uma análise desses dados. E a Integração de Conhecimento de Domínio o minerador de dados poderia integrar diferentes conhecimento de domínio para definir o objetivo.

5.2.4 Tarefa de Mineração de dados

Esta etapa envolve a escolha da tarefa de mineração de dados que será utilizada, e o algoritmo e a técnica que serão utilizados para concretizar a tarefa de mineração de dados, as classes Task, Algorithm e Technic representam respectivamente cada uma dessas etapas. A Figura 35 apresenta essas classes e as tarefas da metodologia D^{3}M que foram detectadas nesta etapa.

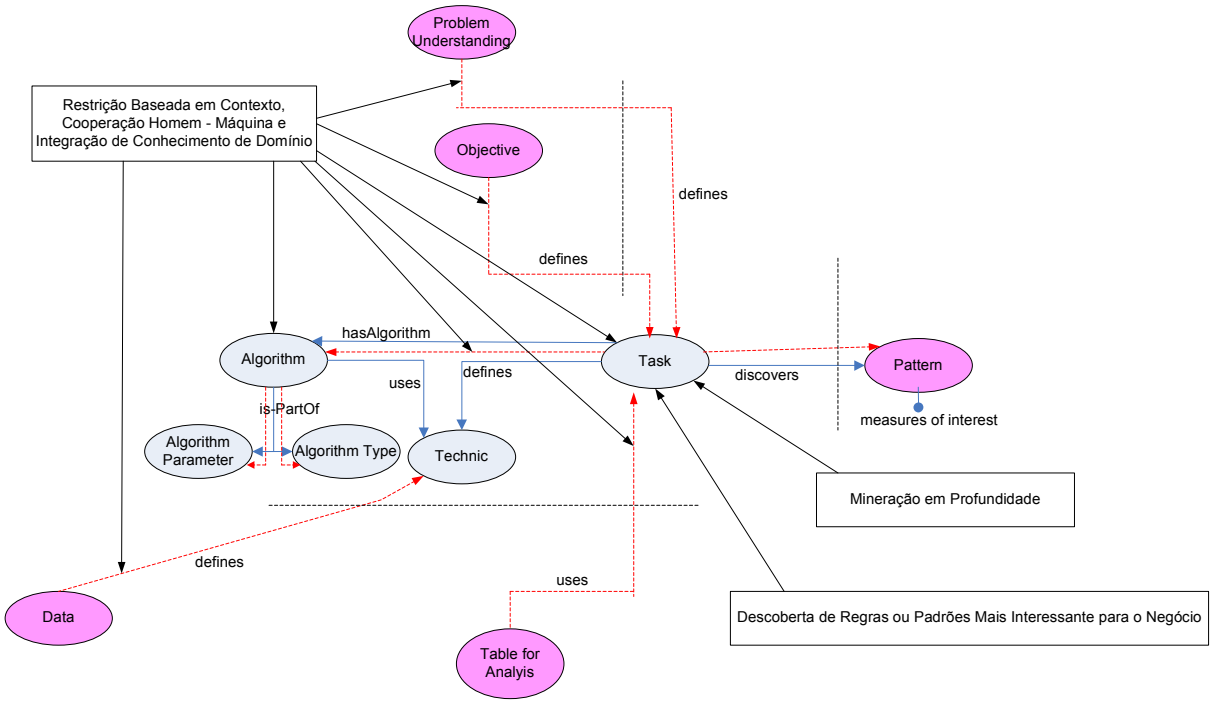


Figura 35: DEFINIÇÃO DA TAREFA DA MD COM A METODOLOGIA D^{3}M

Nesta etapa da ontologia foram detectadas cinco tarefas da metodologia D^{3}M que são: Restrição Baseada em Contexto, Integração de Conhecimento de Domínio, Cooperação entre Homem e Máquina, Mineração em Profundidade e Descoberta de Regras ou Padrões mais Interessante para o Negócio. A seguir é descrita a influência que essas cinco tarefas têm nesta fase da ontologia.

Com relação à Restrição Baseada em Contexto esta tarefa da metodologia D^{3}M tem um papel fundamental para definir a tarefa de mineração de dados, pois o minerador a partir do entendimento dos dados, com os dados prontos para serem minerados e com o entendimento do problema e objetivo, com base na restrição de contexto poderia definir qual seria a tarefa de mineração de dados mais adequada.

Com relação à Cooperação entre Homem e Máquina o minerador de dados juntamente com o sistema de mineração de dados poderia definir a tarefa de mineração de dados, técnica e algoritmo para um determinado projeto de mineração de dados, onde o sistema por meio de interfaces mostraria o que precisa ser realizado e o minerador com seu conhecimento de domínio poderia inserir informações sucintas, como, por exemplo, o minerador de dados poderia definir campos chaves para a descoberta de padrões.

Com relação à Integração de Conhecimento de Domínio o minerador de dados poderia utilizar seu conhecimento de domínio a cerca do projeto de mineração de dados e ainda utilizar ontologias de apoio para esta etapa e então definir qual é a tarefa de mineração de dados mais apropriada para um determinado projeto de mineração de dados, além de poder definir um algorimo e seus parâmetros e a técncica de mineração de dados mais apropriada.

Com relação à Descoberta de Regras ou Padrões mais Interessantes para o Negócio, o minerador de dados poderia avaliar regras acionáveis que seriam disparadas quando dados que as satisfaçam fossem identificados.

Com relação à Mineração em Profundidade o minerador de dados poderia avaliar uma regra acionável e refiná-la para atender aos objetivos do negócio.

O relacionamento discovers não foi mencionado nesta etapa por pertencer à etapa seguinte da ontologia e os outros relacionamentos foram descritos nas seções anteriores.

5.2.5 Padrões

Esta etapa representa os padrões que foram gerados a partir da execução do algoritmo de mineração de dados e que resultados foram obtidos com sua análise. Esta etapa é representada por duas classes: Pattern e Result. A Figura 36 apresenta essas classes, seus relacionamentos e as tarefas da metodologia D^{3}M identificadas.

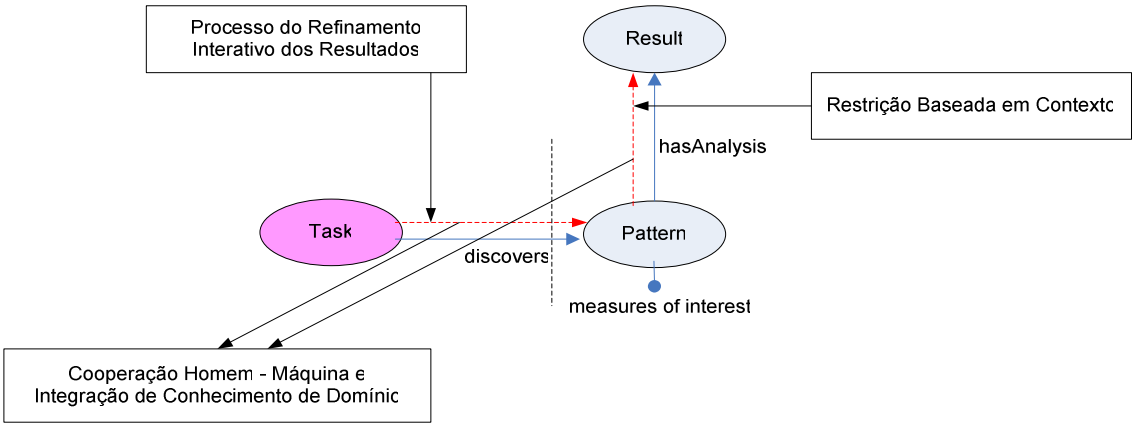


Figura 36: RESULTADOS DA MD COM A METODOLOGIA D3M

Nesta etapa ocorrem quatros tarefas da metodologia D^3M, que são: Processo do Refinamento Interativo dos Resultados, Restrição Baseada em Contexto, Cooperação entre Homem e Máquina e Integração de Conhecimento de Domínio. A seguir é feito uma breve descrição da presença dessas quatro tarefas na ontologia.

Com relação à tarefa Processo do Refinamento Interativo dos Resultados o minerador de dados a partir dos padrões gerados com a aplicação do algoritmo de mineração de dados poderia refinar estes padrões na tentativa de buscar resultados mais interessantes.

Para realizar a operação descrita no parágrafo anterior poderia haver uma cooperação entre o minerador de dados e o sistema de mineração de dados e ainda o minerador de dados utilizar seu conhecimento para refinar os resultados produzidos com a aplicação do algoritmo de mineração de dados.

Com relação à tarefa Restrição Baseada em Contexto o minerador de dados poderia classificar os resultados gerados e concluir se estes padrões gerados de alguma forma atendem aos objetivos dos negócios e se serão utilizados ou não para o objetivo que foi estabelecido.

Para realizar a operação descrita no parágrafo anterior o minerador de dados poderia fazer uso do sistema de mineração de dados e utilizar o conhecimento de domínio juntamente com outros conhecimentos para então definir se os resultados produzidos serão utilizados ou não.

5.3 Uma Arquitetura para Ferramentas de Mineração de Dados Baseada na Metodologia D^3M e na ontologia Meta-DM

A partir da ontologia desenvolvida e da inserção das tarefas da metodologia D^3M apresentada na Seção 5.2 uma arquitetura para ferramentas de mineração de dados foi proposta e ilustrada na Figura 37.

Para o desenvolvimento da arquitetura, a ontologia desenvolvida teve o papel de definir as tarefas essenciais para haver a mineração de dados, além de identificar pontos onde é essencial o conhecimento humano. A Seção 5.2 teve como objetivo identificar quais tarefas da metodologia D^3M seriam inseridas em quais pontos na arquitetura.

A arquitetura possui três camadas, onde a primeira está relacionada à interação do minerador com o sistema. A segunda está relacionada à parte lógica, onde são executadas as operações do processo de KDD. E na terceira camada está o repositório de informações. As subseções a seguir fazem uma descrição dessas camadas e dos elementos que as compõem.

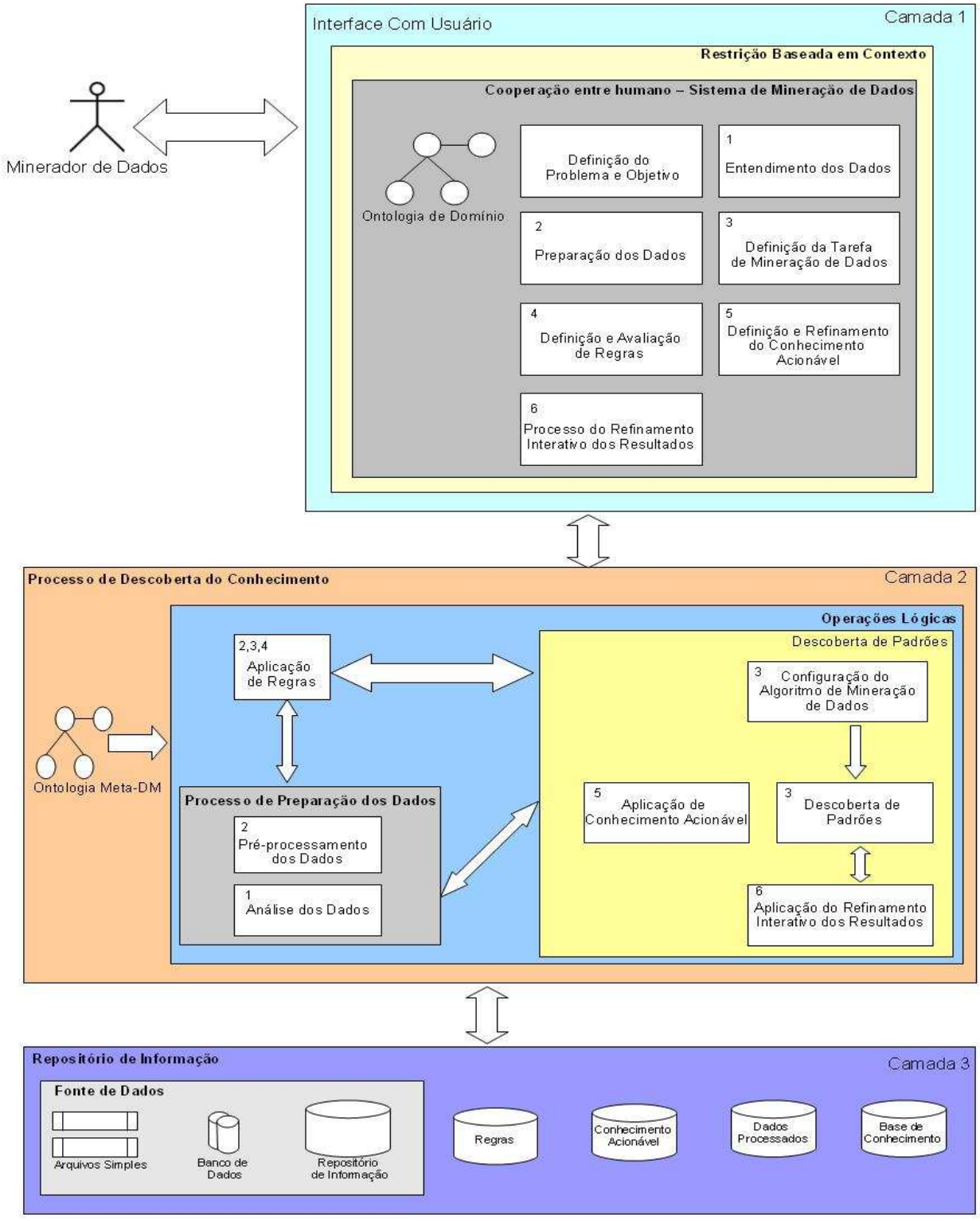


Figura 37: ARQUITETURA PARA FERRAMENTAS DE MINERAÇÃO DE DADOS

5.3.1 Interface com o usuário

A primeira camada está relacionada com a interação entre o minerador de dados e o processo de descoberta de conhecimento, através de interfaces gráficas, onde o minerador interage com o sistema visualizando e inserindo informações necessárias.

Basicamente nessa camada o minerador realiza as seguintes operações: define um objetivo e um problema para o projeto de mineração de dados, faz o entendimento dos dados, faz a preparação dos dados, define a tarefa mais adequada ao projeto, define e avalia regras, define e refina conhecimento acionável e faz refinamentos nos padrões gerados a partir da execução de um algoritmo de mineração de dados.

Foi constatado que durante o processo de mineração de dados a tarefa Restrição Baseada em Contexto estava presente em todas as fases. A arquitetura retratou esta situação colocando as demais tarefas dentro do seu sub-módulo como apresentado na Figura 37.

Conforme a Figura 37, a tarefa de Cooperação entre Homem e Máquina também está presente entre todos os módulos, dessa forma os módulos foram colocadas também em uma subcamada chamada de “Cooperação entre Humano-Sistema de Mineração de Dados”. Por exemplo, em Preparação de Dados o minerador pode definir regras ou outro conhecimento de domínio para tratar os dados, havendo cooperação entre o humano e o sistema de mineração de dados.

Uma ontologia do domínio do problema de mineração foi posicionada na camada de Restrição Baseada em Contexto, pois, além de auxiliar o minerador de dados na definição do objetivo e do problema de mineração, ela pode auxiliar o minerador de dados nas diversas operações que podem ser realizadas na arquitetura.

5.3.2 Processo de Descoberta de Conhecimento

Na segunda camada ocorrem todas as operações lógicas da arquitetura, onde basicamente são realizadas as seguintes tarefas: são aplicadas análises sobre os dados, dados são preparados, são executadas regras, são

executados conhecimentos de domínio, são aplicados parâmetros dos algoritmos, são descobertos padrões ao executar um algoritmo de mineração de dados e estes padrões podem ser refinados. O funcionamento dessa camada está resumido nos próximos parágrafos desta seção.

Nesta camada há um módulo chamado de Aplicação de Regras, onde sua função é executar regras definidas pelo o minerador de dados e que pode tanto ser aplicadas nas subcamadas Preparação dos Dados e Descoberta de Padrões.

Na subcamada Processo para Preparação dos Dados é obtida a base de dados que será trabalhada, esta base de dados é analisada e tratada de acordo com o projeto de mineração de dados. Para o tratamento dos dados o minerador de dados pode definir regras para tratá-los, com a utilização do módulo Aplicação de Regras, mas para realizar esta operação o minerador de dados irá antes realizar a operação de análise dos dados, a qual é realizada no módulo Análise dos Dados.

Na subcamada Descoberta de Padrões é onde será realizado o processo de descoberta de padrões com a aplicação da mineração de dados. Entre as operações que serão executadas nesta camada estão: execução de um sistema especilista para definir a tarefa de mineração de dados (Silva et al. 2009) e também regras que possam auxiliar a descoberta da tarefa, execução de conhecimento acionável, aplicaçãodos parâmetros do algoritmo definido, execução do algoritmo de mineração de dados que pode sofrer interferência de um conhecimento acionável, e por fim os padrões gerados com a aplicação do algoritmo poderão ser refinados.

Os módulos das camadas 1 e 2 estão relacionados, pois para cada tarefa da camada 2 existe a necessidade de interação com o minerador de dados. Para facilitar a visualização dessa relação entre as duas camadas, os módulos da camada 1 foram enumerados para indicar se possuem relacionamento com os módulos da camada 2. Os módulos das duas camadas estão relacionados da seguinte maneira:

- O módulo Análise dos Dados da camada 2 está relacionado com o módulo Entendimento dos Dados da camada 1, onde o minerador de dados fará o entendimento dos dados a serem minerados.

- O módulo Pré-processamento dos Dados da camada 2 está relacionado com o módulo Preparação dos Dados da camada 1, onde o minerador de dados fará todo o processo de preparação dos dados para aplicar a mineração de dados.
- Os módulos Configuração do Algoritmo de Mineração de Dados e Descoberta de Padrões da camada 2 estão relacionados com o módulo Definição da Tarefa de Mineração de Dados da camada 1, onde o minerador de dados irá executar o processo de descoberta de padrões a partir de uma base de dados.
- O módulo Aplicação de Regras da camada 2 relaciona-se com os módulos Preparação dos Dados, Definição da Tarefa de Mineração de Dados e Definição e Avaliação de Regras da camada 1, onde o minerador de dados define regras a serem executadas durante o processo KDD.
- O módulo Aplicação de Conhecimento Acionável da camada 2 está relacionado com o módulo Definição e Refinamento do Conhecimento Acionável da camada 1, onde o minerador de dados irá definir, refinar e aplicar conhecimento acionável durante o processo KDD.
- O módulo Aplicação do Refinamento Interativo dos Resultados da camada 2 está relacionada com o módulo Refinamento Interativo dos Resultados da camada 1, onde o minerador de dados poderá refinar os padrões gerados com a aplicação do algoritmo de mineração de dados.

5.3.3 Repositório de Informação

Esta camada está relacionada às informações e aos dados a serem utilizados ou modificados durante o processo de mineração de dados.

Nesta camada há uma subcamada chamada Fonte de Dados, constituída por arquivos simples, base de dados e repositório de informação. Esta

camada é responsável por abrigar os dados de sistemas de informação convencionais, que precisam ser adequados para o processo de mineração de dados.

Nesta camada há ainda a representação de Regras e Conhecimento Acionável definidos pelo minerador de dados. A camada ainda possui os dados prontos para passar pelo o processo de mineração de dados, representados por Dados Processados. A Base de Conhecimento também está nesta camada; ela tem como intuito armazenar as informações a serem utilizadas durante o processo de mineração de dados, as quais estão estruturadas de acordo com as ontologias. A Base de Conhecimento é formada, portanto, por instâncias das classes das ontologias.

5.4 Cenário de execução de um projeto de mineração de dados

O objetivo desta seção é apresentar um cenário de execução de um projeto de mineração, levando em consideração a arquitetura proposta, que está baseada em ontologias e interação humana, conforme a metodologia D^3M. Em resumo, o cenário mostra a necessidade do conhecimento do problema para definir os objetivos do projeto de mineração, o que pode ser auxiliado por uma ontologia de domínio. O cenário mostra também como o minerador pode fazer restrições baseadas em contexto e aplicar conhecimento acionável. O conhecimento acionável pode consistir em regras que podem ser definidas pelo minerador antes de começar a mineração dos dados, as quais são “acionadas“ se um padrão casar com a regra. Este conhecimento acionável pode, então, ser refinado para melhorar os resultados da mineração.

Este projeto de mineração utiliza na descrição do cenário uma base de dados de um congresso de tecnologia que ocorreu na cidade de Mococa – SP no ano de 2007, mais detalhe dessa base de dados é apresentado no Anexo 1.

Para apresentar as fases de execução do projeto de mineração de dados o cenário foi dividido de acordo com as fases da metodologia CRISP-DM Chapman et al. (2000), as subções a seguir apresentam com mais detalhes essas fases.

5.4.1 Execução do cenário no entendimento do negócio

A intenção da arquitetura proposta é tornar o processo de mineração de dados o mais automático possível. No processo de execução foi levado em consideração o fluxo corrente das tarefas, dessa forma, a primeira etapa a ser realizada é o entendimento do negócio, conforme a metodologia CRISP-DM de Chapman et al. (2000).

A proposta da arquitetura é que o entendimento do negócio seja feito a partir de uma ontologia de domínio desenvolvida especificamente para o problema do congresso. As definições dessa ontologia poderiam ser feitas a partir da ontologia proposta por Sharma e Bryson (2008), na qual eles tratam o entendimento do negócio.

Nesse contexto, a ontologia do domínio do problema deverá auxiliar o minerador a definir o objetivo do projeto que é: "*Definir a melhor forma de divulgação do congresso conforme o perfil do congressista*". A ontologia também deverá ajudar o minerador a definir o problema a ser abordado no projeto: "*A divulgação do congresso gera custo alto e nem sempre é eficiente*".

Na Figura 38 são destacados os pontos em que são executadas as tarefas de entendimento do negócio na arquitetura.

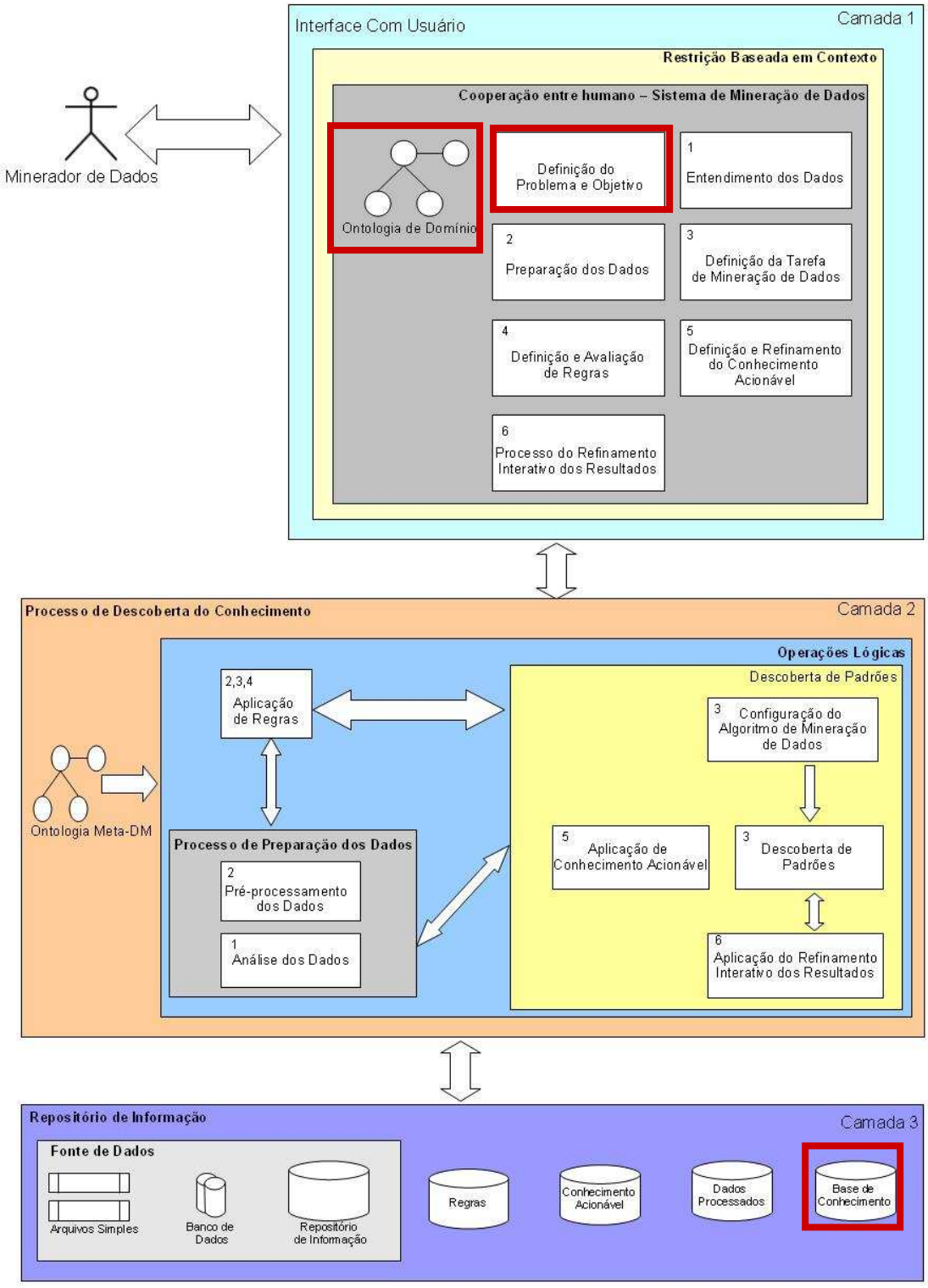


Figura 38: Execução do cenário - Entendimento do Negócio

5.4.2 Execução do cenário no entendimento dos dados

A próxima etapa é o entendimento dos dados, onde a partir de alguns resultados produzidos na fase anterior são definidass tarefas a serem trabalhados nesta fase.

Na etapa de entendimento dos dados conforme o Quadro 5 são definidas 4 tarefas, que são: recolher os dados, descrever os dados, explorar os dados e verificar a qualidade dos dados. O entendimento do negócio, realizado na fase anterior, deve trazer resultados importantes para o módulo Entendimento dos Dados da arquitetura. O objetivo desse módulo é obter informações sucintas sobre as características da base de dados a ser minerada.

Na fase entendimento dos dados, conforme proposto na arquitetura, as tarefas Restrição Baseada em Contexto e cooperação entre o humano-sistema de mineração de dados podem ser utilizadas. O minerador de dados, de acordo com o entendimento do negócio (definido com a ontologia de domínio) e com as tarefas da metodologia D^3M pode fazer uma análise da base de dados. Uns exemplos das informações que podem ser obtidas com esta base de dados são: A base de dados possui cinco tabelas (Alunos, Cidades, Empresas, Inscrições e Instituições), foi constato que a maioria das informações armazenadas são do tipo literal, a quantidade de registros é 2662. Como o objetivo é saber a melhor forma de divulgar um congresso a partir do perfil dos congressistas, a ontologia de domínio pode indicar que os campos: data de nascimento, sexo, e-mail, telefone, escolaridade, cidade e ficou_sabendo são os que melhor caracterizam o perfil dos congressistas e dizem como eles ficaram sabendo do congresso. Esses são, portanto, os campos que podem gerar padrões que podem vir ao encontro do objetivo estabelecido. Por fim, o minerador de dados verificaria a qualidade dos dados.

Na Figura 39 são destacadosos pontos em que são executadas as tarefas de entendimento dos dados na arquitetura.

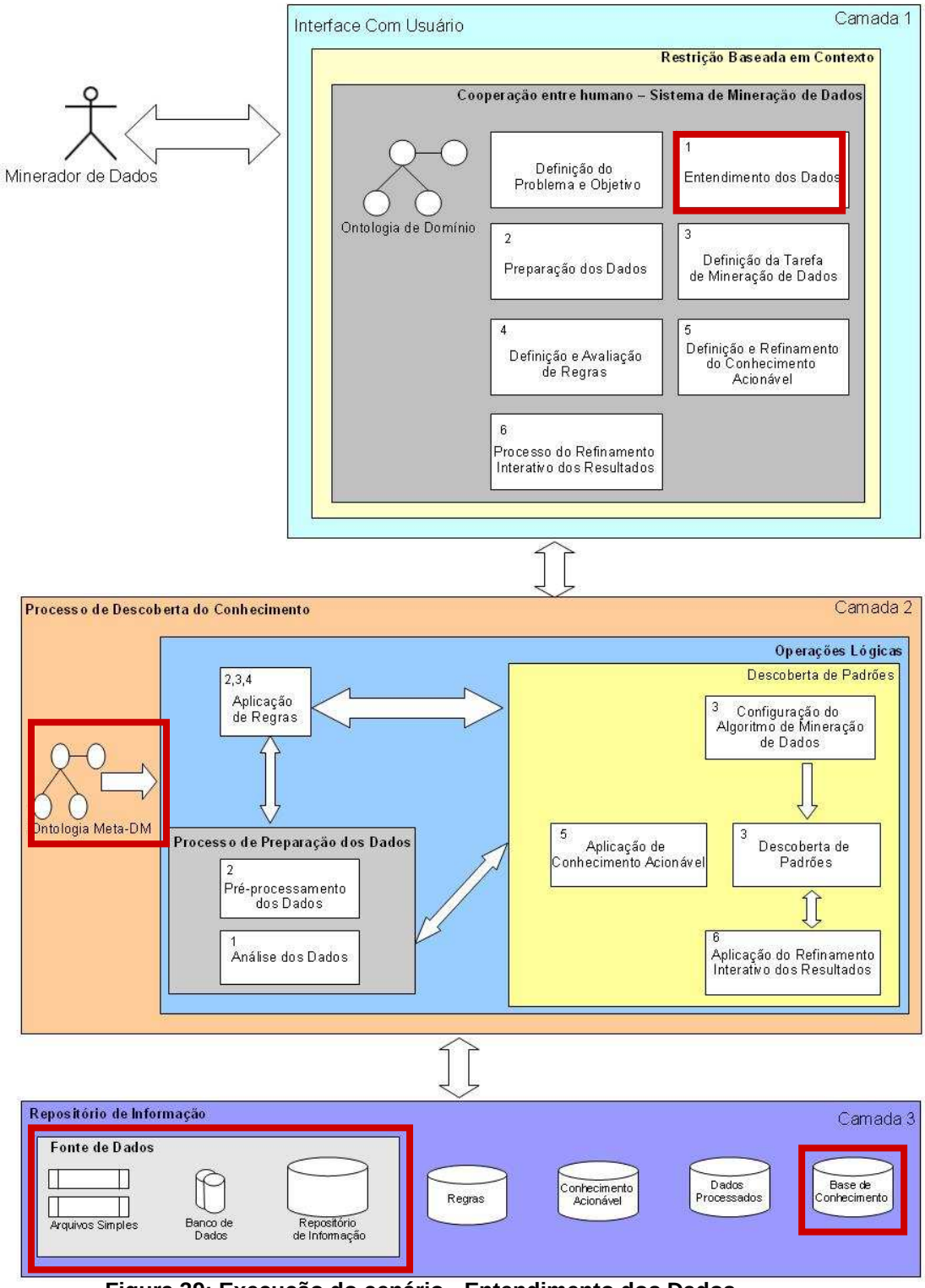


Figura 39: Execução do cenário - Entendimento dos Dados

5.4.3 Execução do cenário na preparação dos dados

A etapa seguinte é preparar os dados para a mineração de dados. Esta etapa é feita a partir do levantamento da qualidade dos dados obtidos na etapa de entendimento dos dados.

Esta etapa deve estar presente no módulo Preparação dos Dados e na subcamada Processo de Preparação dos dados da arquitetura proposta. A etapa é constituída das seguintes tarefas de acordo com a ontologia desenvolvida: integração e remoção de dados ruidosos e com inconsistências, transformação e seleção.

Como exemplo, o minerador pode definir as seguintes regras para transformação nos dados:

```
if campo = email then
   return string among (“@ “ and  “ . ”)

if campo = telefone then
   return string among ( “(“ and “)” )
```

Na primeira regra seria retornado o provedor de e-mail do congressista, por exemplo, no e-mail: edmar.yokome@gmail.com a regra apresentada retornaria gmail. Na segunda regra seria retornado o código postal do congressista, por exemplo, no telefone (64) 3614 – 2552 seria retornado o código 64.

Exemplos de regras para tratar dados ruidosos e com inconsistências seriam:

```
if campo = idade then
   idade válida >=15 e <=30

if campo = estado ≠ de {São Paulo e Minas Gerais}  then
   replace (outros estados)
```

A primeira regra estabelece uma idade estimada dos congressistas, onde valores diferentes do estabelecido seriam considerados uma anomalia. Na segunda foi verificado que apenas os estados de Minas Gerais e São Paulo são interessantes para gerar padrões, uma vez que são raros os participantes de outros estados.

Para a tarefa de seleção dos dados a ontologia Meta-DM poderia induzir o minerador de dados a definir que os campos nascimento, sexo, email, telefone, escolaridade da tabela Alunos; nome da tabela Cidades; ficou_sabendo da tabela Inscrições são interessantes para o projeto de mineração de dados, estabelecidos de acordo com a restrição de contexto e com o objetivo do negócio e apoiado com a tarefa cooperação entre humano-sistema de mineração de dados.

A tarefa de integração dos dados consiste em, a partir dos dados selecionados, fazer a integração desses dados em uma tabela de modo que as informações de cada campo das tabelas não se misturem, ou seja, o campo 1 de uma tabela X será integrado com o campo 1 da tabela Y e assim por diante. Esta tarefa é feita automaticamente pela ferramenta de mineração, porém cabe ao minerador de dados com base na restrição de contexto e cooperação entre humano e sistema de mineração de dados definir quais campos de tais tabelas ou base de dados externas deverão ser integradas.

Na Figura 40 são destacados os pontos em que são executadas as tarefas de preparação dos dados na arquitetura.

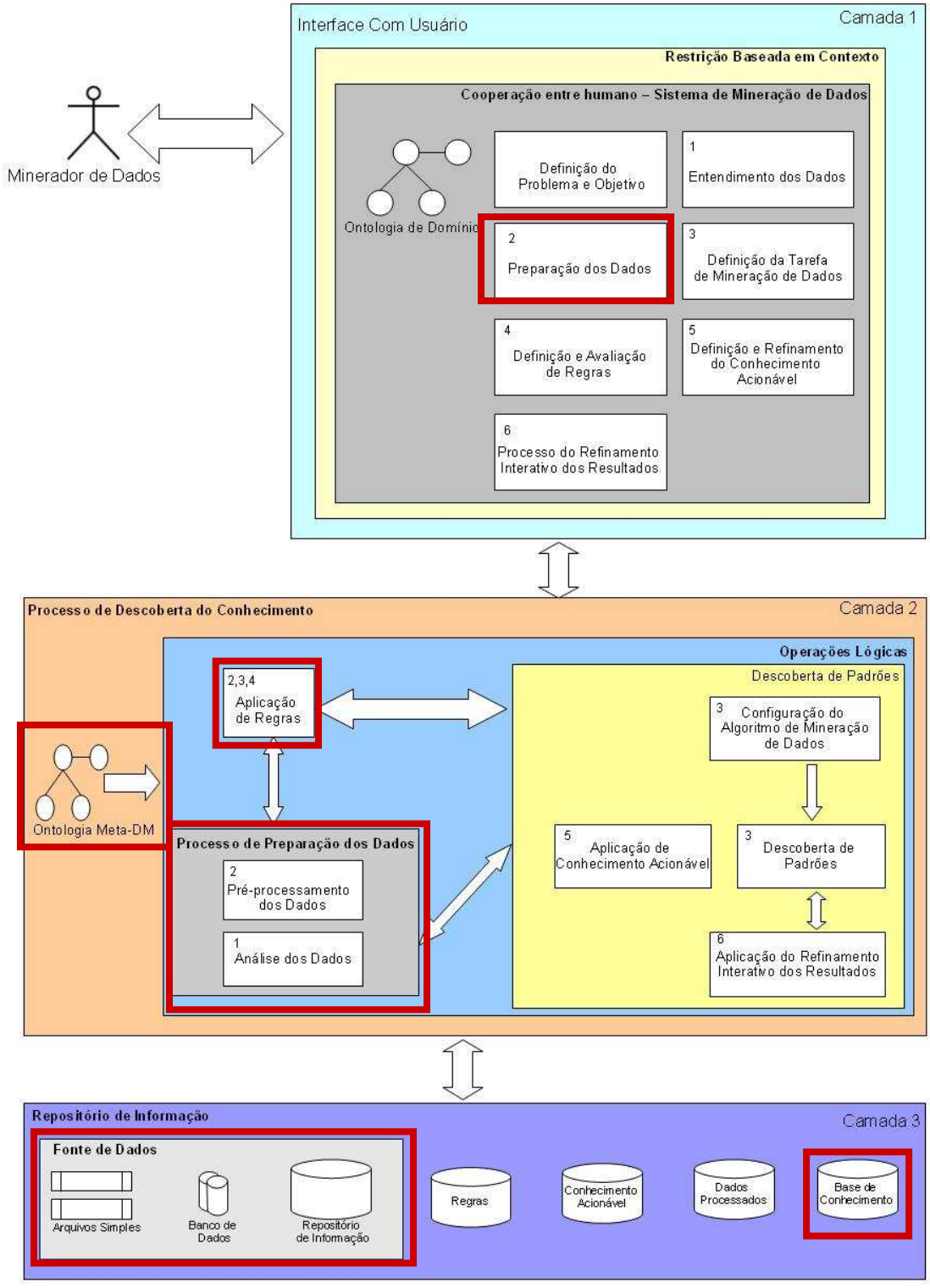


Figura 40: Execução do cenário - Preparação dos Dados

5.4.4 Execução do cenário na definição e aplicação da tarefa de mineração de dados

Feito o tratamento da base de dados, a próxima etapa é definir a tarefa de mineração de dados. Conforme mostra a Figura 34, nessa fase foram identificadas várias tarefas da metodologia D^3M que podem auxiliar o minerador de dados, entre estas tarefas estão: Restrição Baseada em Contexto, Integração de Conhecimento de Domínio, Cooperação entre Homem e Máquina, Mineração em Profundidade, Descoberta de Regras ou Padrões mais Interessante para o Negócio.

Por meio da cooperação entre homem e máquina e restringindo-se ao contexto do problema o minerador de dados poderia definir as seguintes regras para definir qual é a tarefa mais adequada ao projeto de mineração de dados.

se (há relação entre os campos) **e** (campos são mais literais) **e** (há um atributo que está diretamente relacionado com o problema) **então**

tarefa = = associação

Senão se (os atributos podem ser organizados em categorias pré-definidas) **e** (rotulados) **e** (campos são mais numéricos) **então**

tarefa = = classificação

senão se (pode-se dividir os dados em grupos de modo significativo) **e** (grupos compartilham características semelhantes) **então**

tarefa == agrupamento

senão

Não foi possível definir a tarefa de mineração com esta regra!

Estas regras devem estar no módulo Descoberta de Padrões e foram desenvolvidas com base em um sistema especialista descrito em Silva et al. (2009), cujo objetivo é definir uma tarefa de mineração de dados de acordo com a base de dados e os padrões que se deseja gerar.

Antes de ocorrer a aplicação do algoritmo de mineração de dados o minerador pode definir conhecimentos acionáveis, cujo objetivo é encontrar padrões

mais interessantes para o negócio. Por exemplo, ele concluiu que o campo ficou_sabendo é um campo chave para gerar padrões interessantes, dessa forma poderia gerar o seguinte conhecimento acionável.

se campo = ficousabendo então

cidade = “ ? ” and sexo = “ ? “ and estado = “ ? “ and nascimento = “ ? “ and escolaridade = “ ? “ and profissão = “ ? “

Onde com a aplicação do algoritmo de mineração de dados seria gerado apenas padrões que contemplassem os campos definidos com a elaboração do conhecimento acionável.

O conhecimento acionável deve ser elaborado no módulo Definição e Refinamento de Conhecimento Acionável e ficará armazenado no repositório Conhecimento Acionável. Caso sejam encontrados dados que casam com este conhecimento acionável, ele será “acionado” no módulo Descoberta de Padrões. O minerador de dados pode avaliar o retorno do conhecimento acionável conforme o objetivo do negócio, no módulo Definição e Refinamento de Conhecimento Acionável. Na tentativa de atender melhor os objetivos do negócio o minerador de dados ainda pode refinar esse conhecimento neste módulo.

Conforme a regra acionável exemplificada, a mesma será acionada quando o campo ficou_sabendo encontrar os campos (cidade, sexo, estado, nascimento, escolaridade e profissão) onde serão gerados padrões entre estes campos.

Caso o minerador de dados queira refinar esta regra, pode excluir ou inserir novos campos, bem como mudar o campo chave na tentativa de gerar outros padrões que possam vir ao encontro aos objetivos do negócio.

Depois de definida a tarefa de mineração de dados, é definido o algoritmo de mineração de dados, bem como seus parâmetros de execução. O minerador, de acordo com a restrição baseada em contexto e com o objetivo que espera obter, é induzido pela a ontologia Meta-DM a definir que algoritmo Apriori é o mais adequado para gerar os padrões desejados, onde serão geradas associações entre os campos escolhidos. Nesse exemplo, será gerado o relacionamento entre o campo ficou_sabendo com os demais, conforme definido pelo o minerador de dados.

As etapas referentes a definir a tarefa de mineração de dados e seus algoritmos devem estar presentes nos módulos Definição da Tarefa de Mineração de Dados da Camada 1 e ser aplicado nos módulos Configuração do Algoritmo de Mineração de Dados e Descoberta de Padrões da Camada 2.

Depois de definida a tarefa de mineração de dados, o algoritmo a ser utilizado, a aplicação ou não de um conhecimento acionável e/ou seu refinamento, a próxima etapa é a geração de padrões a partir dessas definições.

Com a aplicação do algoritmo de mineração de dados será gerado um conjunto de padrões, no módulo Descoberta de Padrões, conforme apresentado a seguir:

Cidade = Piracicaba => Ficou Sabendo = Televisão {suporte 25% confiança 40%}
Estado = São Paulo => Ficou_Sabendo = Televisão {suporte 35% confiança 84%}
Idade >20 e < 30 => Ficou_Sabendo = Televisão {suporte 40% confiança 86%}
Sexo = Masculino => Ficou_Sabendo = Jornal {suporte 30% confiança 76%}
Idade < 19 => Ficou_Sabendo = Televisão Jornal {suporte 40% confiança 86%}
Idade >= 30 => Sexo = Masculino {Suporte 45% confiança 89%}
Estado = São Paulo => Sexo = Masculino {Suporte 45% confiança 89%}
Estado = São Paulo => Escolaridade = Ensino Médio {Suporte 35% confiança 81%}
Escolaridade=Curso Superiro=>Ficou_Sabendo= Rádio {suporte 30% confiança 76%}

Na Figura 41 são destacados os pontos em que são executadas as tarefas de preparação dos dados na arquitetura.

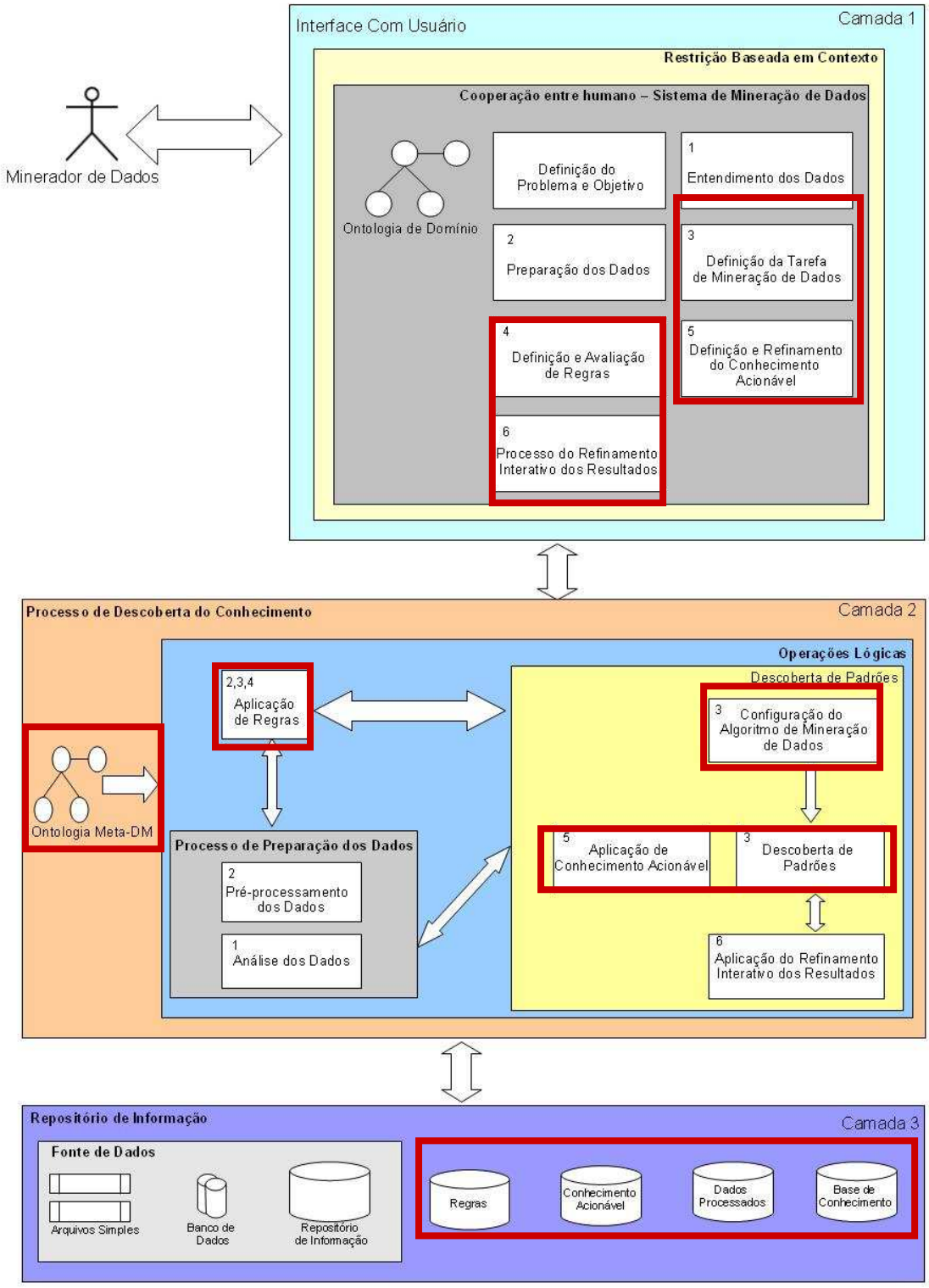


Figura 41: Execução do cenário - Definição e Aplicação da Tarefa de Mineração de Dados

5.4.5 Execução do cenário na avaliação dos padrões gerados

Após a gerar os padrões com a aplicação do algoritmo de mineração de dados ilustrado na seção 5.4.4, estas regras devem ser analisadas e avaliadas pelo minerador de dados, onde o minerador pode classificar as como ruim, regular ou boa, de acordo com o objetivo do projeto de mineração de dados.

Com o intuito de descobrir padrões mais interessantes do que o produzido pelo algoritmo de mineração de dados, o minerador pode refinar algumas regras classificadas como boas, na tentativa de buscar resultados mais satisfatórios. Essa etapa é contempla no módulo de Refinamento Interativo dos Resultados. Por exemplo, suponha que das regras apresentadas, o minerador de dados verificou que as seguintes poderiam ser refinadas:

Estado = São Paulo => Ficou_Sabendo = Televisão {suporte 35% confiança 84%}
Idade >20 e < 30 => Ficou_Sabendo = Televisão {suporte 40% confiança 86%}
Idade < 19 => Ficou_Sabendo = Televisão Jornal {suporte 40% confiança 86%}
Estado = São Paulo => Sexo = Masculino {Suporte 45% confiança 89%}
Estado = São Paulo => Escolaridade = Ensino Médio {Suporte 35% confiança 81%}

Uma das formas de refinar estas regras seria criar um micro-cenário a partir do cenário principal e então refinar os padrões gerados de acordo com o micro-cenário elaborado. Por exemplo, o minerador de dados estabeleceu que para o objetivo do projeto de mineração de dados os padrões que devem ser considerados são congressistas que são do estado de São Paulo, têm idade entre 20 e 30 anos e são do sexo masculino. Ao estabelecer estas restrições seriam retornados apenas os padrões que contemplam estas informações. Nos padrões considerados bons pelo o minerador de dados e com as restrições estabelecidas, seriam retornados os seguintes padrões:

Estado = São Paulo => Ficou_Sabendo = Televisão {suporte 35% confiança 84%}
Idade >20 e < 30 => Ficou_Sabendo = Televisão {suporte 40% confiança 86%}
Sexo = Masculino => Ficou_Sabendo = Televisão {Suporte 45% confiança 89%}

Na Figura 42 são destacados os pontos em que são executadas as tarefas de avaliação dos resultados gerados com a aplicação da mineração de dados na arquitetura.

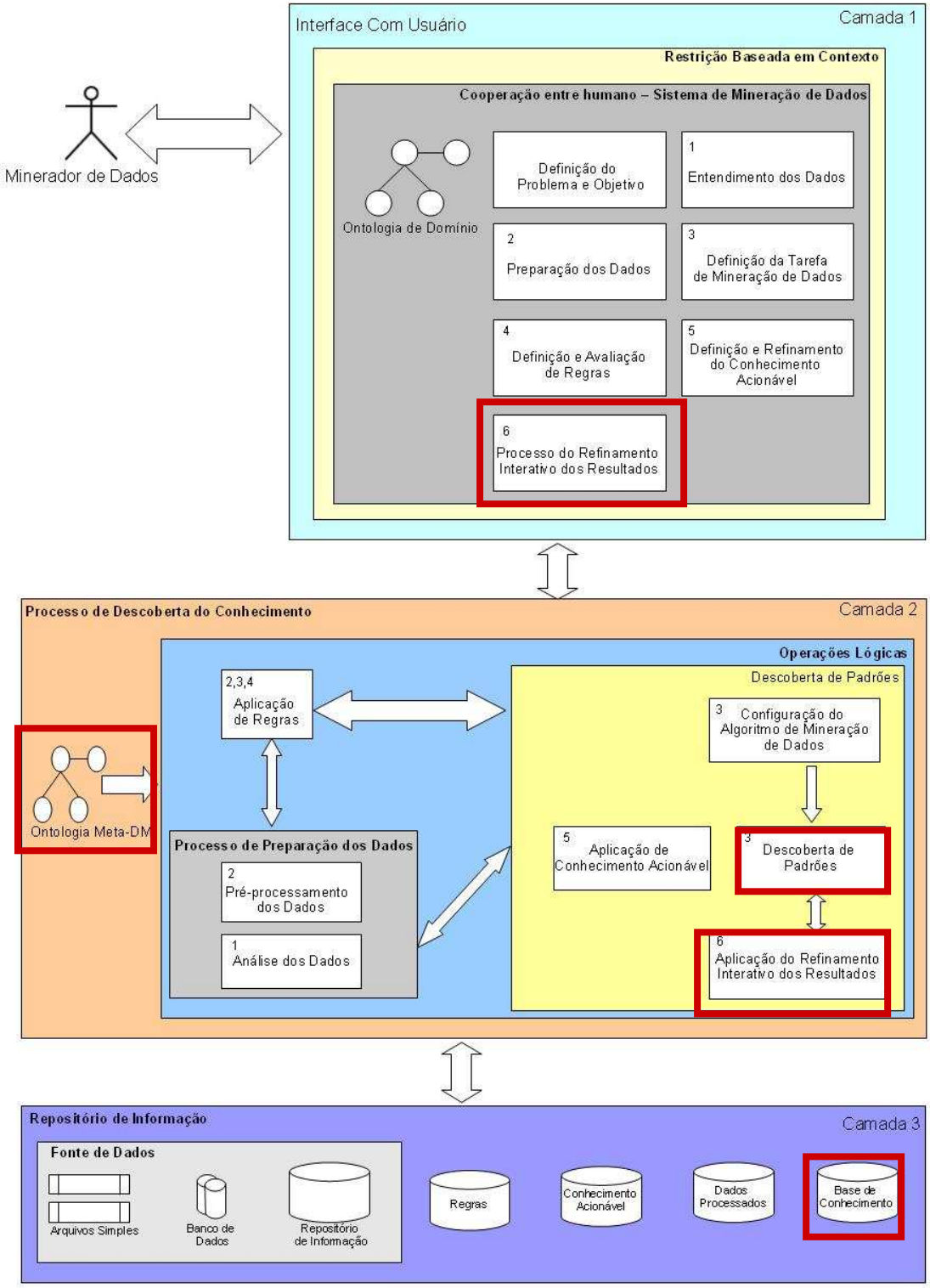


Figura 42: Execução do cenário - Avaliação dos Padrões Gerados

5.4.6 Execução do cenário na aplicação dos padrões gerados

Por fim o minerador de dados irá definir se o que foi gerado atende os objetivos do negócio e se os padrões gerados serão utilizados. Como exemplo, a partir do refinamento dos padrões gerados na seção 5.4.6 o minerador definiu que para o público que mora no estado de São Paulo, tem idade entre 20 e 30 anos, e é do sexo masculino a melhor forma de divulgar o congresso é através de televisão. De posse dessa informação o minerador de dados, juntamente com os envolvidos no projeto de mineração de dados, vai decidir se este resultado é interessante ou não ao objetivo do negócio.

5.5 Considerações Finais

A mineração de dados orientada ao domínio ainda está dando seus primeiros passos, mas tem mostrado que pode gerar resultados mais eficientes, pois leva em consideração uma mineração de dados mais interativa, onde o minerador de dados acompanha mais de perto todo o processo, restringindo-se a um determinado contexto, e insere informações de apoio ao processo.

Este capítulo definiu quais tarefas da metodologia D^3M estão presentes em cada fase da ontologia onde o conhecimento humano foi identificado. A partir dessa identificação uma arquitetura para ferramentas de mineração de dados foi proposta. A arquitetura leva em consideração as tarefas de uma metodologia orientada aos dados como a CRISP-DM, representadas na ontologia Meta-DM, e as tarefas da metodologia D^3M, tendo como intuito ser uma arquitetura para ferramentas de mineração de dados orientadas ao domínio.

Para ilustrar a utilização da arquitetura proposta foi apresentado um cenário, que mostra a influência que o minerador de dados tem durante o processo de mineração de dados, quando trabalha em conjunto com mineração orientada ao domínio.

6 CONCLUSÕES

6.1 Introdução

Neste trabalho, uma ontologia para o domínio de mineração de dados foi desenvolvida com a utilização de duas metodologias: Noy e McGuiness (2001) e METHONTOLOGY de Fernández-López (1997).

O levantamento dos requisitos para seu desenvolvimento foi feito a partir da metodologia CRISP-DM de Chapman et al. (2000), e também foram considerados livros, artigos e especialistas da área.

A ontologia Meta-DM oferece uma terminologia comum, que pode ser compartilhada e compreendida por ferramentas de mineração de dados. Diferente de outras ontologias para o domínio de mineração de dados encontradas na literatura, a Meta-DM identifica e formaliza em quais fases da mineração de dados o conhecimento humano deve ser inserido durante o processo de KDD. Esse diferencial é importante para que o conhecimento humano e de domínio possa ser inserido em ferramentas de mineração e, consequentemente, ajudar ou guiar o minerador de dados durante o processo de descoberta de conhecimento.

Uma das formas de se obter resultados mais satisfatórios durante o processo KDD é a partir dos pontos onde foram identificados conhecimento humano na ontologia inserir tarefas da metodologia D^3M, visando uma mineração de dados mais interativa entre a máquina e o minerador de dados. Com essa interação, é possível obter melhores resultados da mineração de dados. Por isso, com base no desenvolvimento da ontologia Meta-DM, foi proposta uma arquitetura para ferramentas de mineração de dados levando-se em consideração a metodologia D^3M.

6.2 Contribuições

Uma contribuição deste trabalho foi desenvolvimento de uma ontologia que oferece uma terminologia comum para ferramentas de mineração de dados. Diferente de outros trabalhos encontrados na literatura, a ontologia Meta-DM especifica pontos em que são essenciais o conhecimento humano, que é uma tendência na área de mineração de dados, como evidencia metodologias como D^3M;

Outra contribuição deste trabalho foi a proposta de uma arquitetura para ferramentas de mineração de dados levando em consideração a metodologia D^3M e a ontologia Meta-DM. A arquitetura baseada em ontologias vem contribuir com o estado da arte na área de semântica em mineração de dados, uma vez que insere conhecimento humano e de domínio durante o processo de mineração de dados realizado em ferramentas de mineração.

Com o desenvolvimento desse trabalho também foram produzidos alguns artigos científicos:

- “Desenvolvimento de um Metamodelo Baseado em Ontologias para o Domínio de Mineração de dados”, este artigo foi apresentado no 8° Congresso de Pós-Graduação da Universidade Metodista de Piracicaba, ano 2010;
- “Ontologias para o Domínio de Mineração de dados”, este artigo foi apresentado na 4ª JORNADA ACADÊMICA: EM DEBATE: CIÊNCIA, TECNOLOGIA E INOVAÇÃO na Universidade Estadual de Goiás – UnU Santa Helena de Goiás, ano 2010;
- “Meta-DM: Uma ontologia para o domínio de mineração de dados”, submetido para Revista de Sistemas de Informação da Faculdade Salesiana Maria Auxiliadora (FSMA), ano 2011.
- Um artigo abordando a arquitetura baseada em ontologias está sendo escrito.

6.3 Trabalhos Futuros

Com a ontologia Meta-DM formalizada e avaliada, e com a arquitetura proposta, trabalhos futuros podem focar na integração da arquitetura com uma ferramenta de mineração de dados, como, por exemplo, a Kira, com o intuito de melhorar a questão da participação humana e do conhecimento de domínio na ferramenta. Para tanto, uma sugestão é utilizar a Meta-DM e a arquitetura para identificar e implementar na ferramenta Kira interfaces mais interativas com o minerador de dados.

REFERÊNCIAS

Beckett D. and Berners-LeeT. “Turtle – terse rdf triple language,” *W3C Team Submission*. Disponível em: http://www.w3.org/TeamSubmission/turtle/, 2008. Acesso: 07 jan 2011.

BREITMAN, K. Web Semântica – A Internet do Futuro. LTC, 2005.

BREZANY, P. ; JANCIAK, I. ; TJOA, A. M. *Ontology-Based Construction of Grid Data Mining Workflows*. In: NIGRO, H. O. ; CÍSARO, S. E. G. ; & XODO, D. H. Data Mining with Ontologies: Implementations, Findings and Frameworks. London, IGI Global, 2008. p. 182-210.

BRICKLEY, D.; GUHA, R. V. RDF Vocabulary Description Language 1.0: RDF Schema. *W3C Recommendation*. 2003. Disponível em: <http://www.w3.org/TR/rdf-schema>. Acesso: 07 jan 2011.

CAO, L.; ZHANG, C. Domain-Driven Data Mining: A Practical Methodology. *International Journal of Data Warehousing & Mining*, v. 2, n. 4, p. 49-65, 2006.

CAO, L. Domain Driven Data Mining (D3M). In: *IEEE International Conference on Data Mining Workshops*, 74-76, 2008, Shanghai (China).

CHAPMAN, P.; CLINTON, J.; KERBER, R.; KHABAZA, T.; REINARTZ, T.; SHEARER, C. CRISP-DM 1.0 Step-by-step Data Mining Guide. 2000. Disponível em: <http://www.crisp-dm.org/download.htm>. Acesso em: 28 abr. 2010.

FALBO, R. A.; MENEZES, C. S.; ROCHA, A. R. C. A Systematic Approach for Building Ontologies, In: *Proceedings of the 6 th Ibero-American Conference on IA: Progress in Artificial Intelligence*, 349 – 360, 1998, London (UK).

FERNÁNDEZ, M; GÓMEZ-PÉREZ, A.; JURISTO, N. Methontology: From Ontological Art Towards Ontological Engineering, In: *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, 33-40, 1997, Stanford (USA).

GRUBER, T. R. A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, v. 5, n. 2, p. 199-220, 1993..

GRÜNINGER, M.; FOX, M. S. Methodology for the Design and Evaluation of Ontologies, In: *Proceedings of the Workshop on Workshop on Basic Ontological Issues in Knowledge Sharing*, IJCAI95, 1995, Montreal (Canadá). 10p.

GUARINO, N. Formal Ontology and Information Systems, In: *Proceedings*. Amsterdan: IOS on Formal Ontology and Information Systems (FOIS'98), 3 -15, 1998, Trento (Itália).

GUEDES, G. T. A. UML 2 – Uma Abordagem Prática. Novatec, 2009.

HAN, J.; KAMBER, M. *Data Mining: Concepts and Techniques*. Ed: Elsevier. Second Edition, 2006.

HORRIDGE, M. ; DRUMMOND, N. JUPP, S. ; MOULTON, G. ; STEVENS, R. *A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools.* Edition 1.2, The University of Manchester, 2009.

LINHALIS, F. *Mapeamento semântico entre UNL e Componentes de software para execução de requisições imperativa em linguagem natural*. 2007. 244 f, Tese (Programa de Pós Graduação em Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos.

LINHALIS, F. Web Semântica – Teoria e Prática. Centro de Tecnologia da Informação Renato Archer, 2010.

MANOLA, F.; MILLER, E. RDF Primer. *W3C Recommendation*. 2004. Disponível em: <http://www.w3.org/TR/rdf-primer/>. Acesso em: 07 jan. 2011.

MARTIMIANO, L. A. F. Sobre a estruturação de informação em sistemas de segurança computacional: o uso de ontologia. 2006. 185 f, Tese (Programa de Pós Graduação em Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos.

MENDES, E. F.; VIEIRA, M. T. P. Automatização da Técnica de Mineração de Dados Auxiliada Por Guias. In: *XX Simpósio Brasileiro de Informática na Educação*, 2009. Santa Catarina (Brasil), 10 p.

MENDES, Eduardo Fernando. *Kira: Uma Ferramenta Instrucional para Apoiar a Aplicação do Processo de Mineração de Dados*. 2009. 115 f, Dissertação (Programa de Pós Graduação em Ciência da Computação) - Faculdade de Ciências Exatas e da Natureza, da Universidade Metodista de Piracicaba – UNIMEP.

MCGUINNESS, D. L.; VAN HARMELEN, F. Web Ontology Language Overview. *W3C Recommendation*. 2004. Disponível em: <http://www.w3.org /TR/owl-features>. Acesso: 07 jan 2011.

NOY, N. F.; McGuinness, D. L. Ontology Development 101: A Guide to Creating Your First Ontology. 2001. Relatório Técnico – Stanford University, Stanford. Disponível em: <http://www.ksl.stanford.edu/people/dlm/papers/ontoloy101/ ontogy101-noy-mcguinners.html>. Acesso em 07 jun 2010.

PÁDUA, S. I. D. Método de Avaliação do Modelo de Processo de Negócio do EKD. 2004. 271 f, Tese (Programa de Pós Graduação em Engenharia Mecânica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.

PANOV, P. ; DŽEROSKI, S. ; STEFAN, J. SOLDATOVA, L. N. OntoDM: An Ontology of Data Mining. *In: IEEE International Conference on Data Mining Workshops*, 2008.

PINTO, F. M; SANTOS, M. F. Considering Application Domain Ontologies for Data Mining. *In WSEAS Transactions on Information Science and Appications*. 1478 – 1492. Stevents Point, Winconsin, USA.

RUSSEL, S.; NORVING, P. *Artificial Intelligence: A Modern Approach*. Upper Saddle River: Prentice Hall, 2003.

SANTOS, M. Y. ; RAMOS, R. Business Intelligence – Tecnologias da Informação na Gestão de Conhecimento, FCA, 2ª Edição, 2009.

SHARMA, S.; OSEI-BRYSON, K. Organization-Ontology Based Framework for Implementing the Business Understanding Phase of Data Mining Projects, In: *Proceedings of the 41 st Annual Hawaii International Conference on System Sciences* (HIC 33 2008), Waikoloa, Big Island, Hawaii, 2008.

SILVA, A. E. A. ; VIEIRA, M. T. P. ; PEIXOTO, C. S. A. ; MENDES, E. F. ; GOMIDE, R. S. KIRA – A Tool Based on Guides and Domain Knowledge to Instruct Data Miining*. In IADIS - International Conference Applied Computing*, 2009.

TAN, P.; STEINBACH, M.; KUMAR, V. *Introdução ao Data Mining (Mineração de Dados)*, Ciência Moderna, 2009.

USHOLD, M.; KING, M. *Towards a Methodology for Building Ontologies*, *In: IJCAI Workshop on Basic Ontological Issues in Kwowledge Sharing*, 1995, Montreal. *Proceedings*. Também disponível como AIAI-TR-183 form AIAI, University of Edinburg. 1995. 15p.

WITTEN, I. H.; FRANK, E. *Data Mining - Practical Machine Learning Tools and Techniques*. Elsevier. Second Edition, 2005.

ZHENG, L. ; LI X. An Ontology Reasoning Architecture for Data Mining Knowledge Management. *Whuhan University Journal of Natural Sciences*, v. 13, p. 396-400, 2008.

ANEXO 1

Este anexo tem como finalidade apresentar uma base de dados no qual é utilizada em várias seções desse trabalho para realização de testes.

A base de dados em questão é uma base de dados de um congresso de tecnologia que ocorreu na cidade de MOCOCA – SP no ano de 2007. A Figura 43 apresenta a modelagem dessa base de dados.

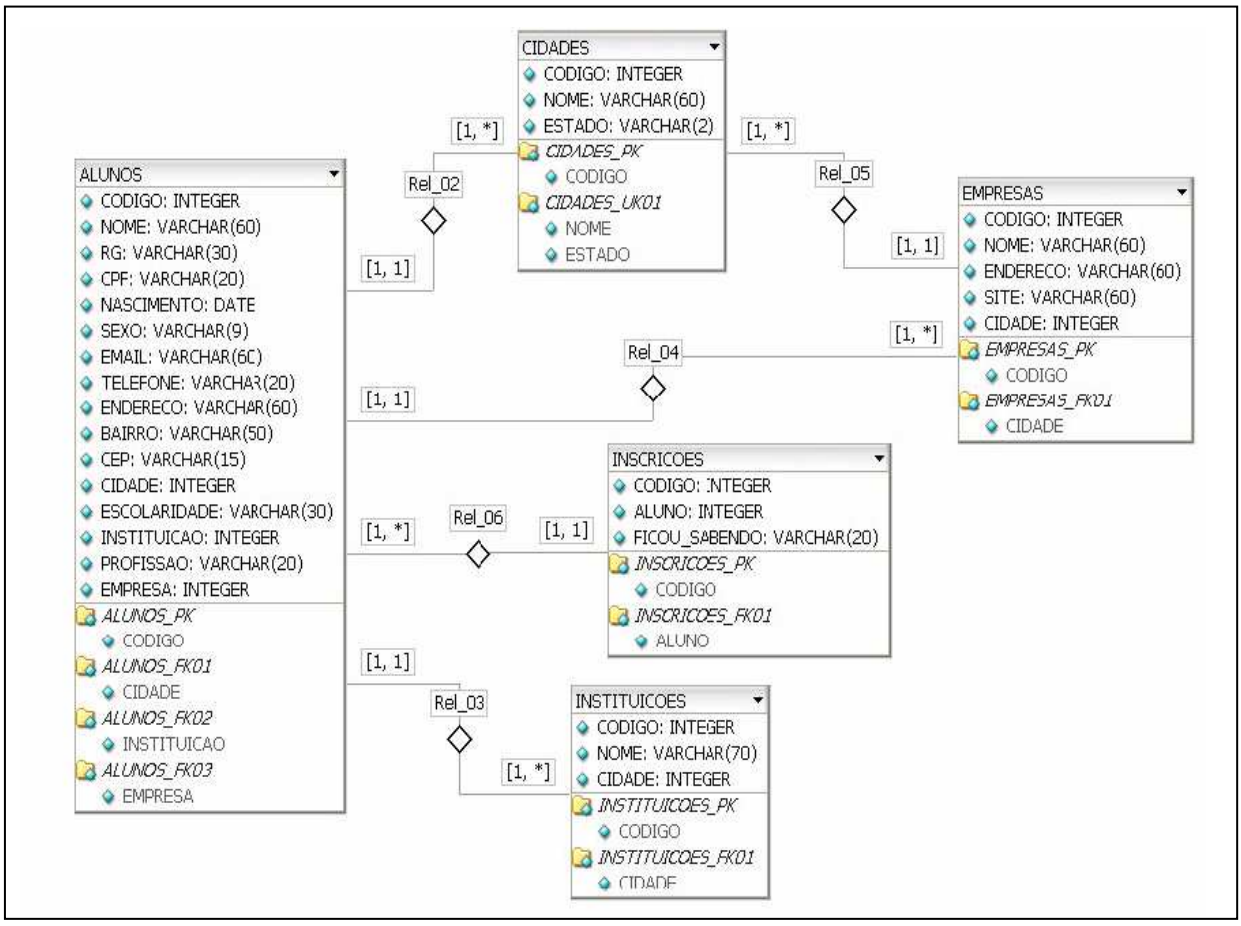


Figura 43: MODELAGEM DA BASE DE DADOS DO CONGRESSO
Fonte: Mendes (2009)

A partir dessa base de dados Mendes (2009) fez o tratamento dos dados de acordo com o processo KDD.Entre os tratamentos realizados estão a seleção dos dados (considerados relevantes para o projeto de mineração de dados em questão). Estes dados foram integrados em uma tabela, mostrada particialmente na Figura 44.

nascimento	sexo	Email	telefone	Bairro	Cidade	estado	escolaridade	instituicao	ficousabendo	profissao
20/10/1975	masculino	ratecmg@uol.com.br	(19)97075034	JARDIM SANTA CLARA	MOCOCA	SP	Primeiro Grau completo	OUTRAS	Televisao	Estudante
20/6/1988	feminino	alinec.ribeiro@hotmail.com	(19)3681-2168	VILA FORMOSA	SAO JOSE DO RIO PARDO	SP	Primeiro Grau completo	FATEC-MOCOCA	Televisao	Estudante
15/8/1989	feminino	malu_moreira_89@yahoo.com.br	(19)36565275	SANTA CECILIA	MOCOCA	SP	Primeiro Grau completo	ETec FRANCISCO GARCIA	Televisao	Estudante
25/6/1964	feminino	ete060dir@ig.com.br	(19)3656-6864	JARDIM ALVORADA	MOCOCA	SP	Primeiro Grau completo	ETec FRANCISCO GARCIA	Televisao	Estudante
9/3/1983	feminino	renity_rosa@yahoo.com.br	(19)3665-2609	NENE PEREIRA LIMA	MOCOCA	SP	Terceiro Grau completo	FATEC-MOCOCA	Amigos	Estudante
31/8/1981	masculino	luizcarlosvb@yahoo.com.br	035 3555 1178	CENTRO	GUARANÉSIA	MG	Pos Graduação	OUTRAS	Banners e faixas	Estudante
19/10/1969	masculino	cebolinha1969@bol.com.br	(16)99668919	COHAB	CASSIA DOS COQUEIROS	SP	Terceiro Grau incompleto	FATEC-MOCOCA	Outros	Estudante
10/2/1964	masculino	valdelinobrega@ig.com.br	(19)3681-3761	VILA BRASIL	SAO JOSE DO RIO PARDO	SP	Terceiro Grau completo	ETec SÃO JOSÉ DO RIO PARDO	Amigos	Estudante
3/12/1989	masculino	bruno_quack@hotmail.com	(19)9655-7740	COHAB 2	MOCOCA	SP	Primeiro Grau completo	ETec FRANCISCO GARCIA	Televisao	Estudante
12/1/1991	masculino	eriveltofermino@bol.com.br	(19)97658161	MOCOQUINHA	MCOCA	SP	Segundo Grau incompleto	ETec FRANCISCO GARCIA	Folhetos	Estudante
8/6/1977	masculino	cortez.junior@uol.com.br	(19)36560189	SÃO DOMINGOS	MOCOCA	SP	Pos Graduação	OUTRAS	Outros	Estudante
7/7/1990	masculino	arthudias@gmail.com	92781239	MOCOQUINHA	MOCOCA	SP	Primeiro Grau completo	ETec JOÃO BAPTISTA DE LIMA FIGUEIREDO	Televisao	Estudante
17/8/1981	masculino	ramonc@rantac.com.br	91908152-3295	JARDIM CHICO PISCINA	MOCOCA	SP	Terceiro Grau incompleto	FATEC-MOCOCA	Folhetos	Estudante
1/12/1986	feminino	anacarolina111@gmail.com	(19)3665-1750	CONDOMINIO MONTE BELO	MOCOCA	SP	Segundo Grau completo	ETec JOÃO BAPTISTA DE LIMA FIGUEIREDO	Televisao	Estudante
2/2/1985	masculino	éderson_damata@hotmail.com	(19)3656-2309	COHAB 2	MOCOCA	SP	Primeiro Grau completo	OUTRAS	Folhetos	Estudante
7/6/1984	masculino	nandofatec@gmail.com	(19)3608-6456	JD EUNICE	SAO JOSE DO RIO PARDO	SP	Terceiro Grau incompleto	FATEC-MOCOCA	Banners e faixas	Estudante
5/10/1984	masculino	thiago_tmbenedetti@hotmail.com	(19)9631-6565	JR. SÃO DOMINGHOS	MOCOCA	SP	Terceiro Grau incompleto	FATEC-MOCOCA	Amigos	Estudante
10/5/1987	feminino	M18vgs@yahoo.com.br	(19)36652373	SÃO DOMINGOS	MOCOCA	SP	Terceiro Grau incompleto	FATEC-MOCOCA	Outros	Estudante
16/4/1990	feminino	Maryane_caroline@hotmail.com	(19) 3665-4573	COHAB1	MOCOCA	SP	Segundo Grau incompleto	ETec FRANCISCO GARCIA	Folhetos	Estudante
10/5/1981	masculino	rotta@bol.com.br	19-9793 8597	JD. RIGOBELO	MOCOCA	SP	Terceiro Grau incompleto	FATEC-MOCOCA	Banners e faixas	Estudante

Figura 44: INTEGRAÇÃO DA BASE DE DADOS PARA A MD
Fonte: Mendes (2009)

A partir dos dados armazenados nessa tabela foram feitos testes ao longo do desenvolvimento do trabalho, com base nas seguintes questões levantadas por Mendes (2009), como: o problema que o projeto de mineração de dados tem que resolver é: “A divulgação do congresso gera custo alto e nem sempre é eficiente”. E o objetivo do projeto da mineração de dados é: “Definir a melhor forma de divulgação do congresso conforme o perfil do congressista”.

A partir dessa base de dados então foi feita a instanciação da ontologia, foi criado um cenário de simulação de um projeto de mineração de dados, foram exemplificados vários trechos do projeto. Dessa forma esta base de dados foi essencial para o desenvolvimento deste projeto.

APÊNDICE 1

O Apêndice 1 apresenta a implementação da ontologia, no qual foi utilizada a ferramenta Protégé onde foram utilizados as abas: classes, Object Properties e Data Properties e foram implementados os seguintes elementos: Namespaces, Classes, Object Properties, Data properties e General axioms. A seguir é apresentada a codificação OWL serializada em Turtle.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix : <http://www.semanticweb.org/metadm.owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@base <http://www.semanticweb.org/metadm.owl> .

<http://www.semanticweb.org/metadm.owl> rdf:type owl:Ontology .


#################################################################
#
#    Object Properties
#
#################################################################

###  http://www.semanticweb.org/metadm.owl#defines

:defines rdf:type owl:ObjectProperty ;

         rdfs:domain :Business_Understanding ;

         rdfs:range :Objective ,
                    :Problem ;

         rdfs:subPropertyOf owl:topObjectProperty .


###  http://www.semanticweb.org/metadm.owl#definesTask

:definesTask rdf:type owl:ObjectProperty ;

             rdfs:domain :Problem_Understanding ;

             rdfs:range :Task ;

             rdfs:subPropertyOf owl:topObjectProperty .


###  http://www.semanticweb.org/metadm.owl#definesTechnic

:definesTechnic rdf:type owl:ObjectProperty ;
```

```
                rdfs:domain :Task ;
                rdfs:range :Technic ;
                rdfs:subPropertyOf owl:topObjectProperty .

###  http://www.semanticweb.org/metadm.owl#discovers
:discovers rdf:type owl:ObjectProperty ;
           rdfs:range :Pattern ;
           rdfs:domain :Task ;
           rdfs:subPropertyOf owl:topObjectProperty .

###  http://www.semanticweb.org/metadm.owl#hasAlgorithm
:hasAlgorithm rdf:type owl:ObjectProperty ;
              rdfs:range :Algorithm ;
              rdfs:domain :Task ;
              rdfs:subPropertyOf owl:topObjectProperty .

###  http://www.semanticweb.org/metadm.owl#hasAnalysis
:hasAnalysis rdf:type owl:ObjectProperty ;
             rdfs:domain :Pattern ;
             rdfs:range :Result ;
             rdfs:subPropertyOf owl:topObjectProperty .

###  http://www.semanticweb.org/metadm.owl#hasCleaning
:hasCleaning rdf:type owl:ObjectProperty ;
             rdfs:range :Table_For_Analyis ;
             rdfs:domain :Table_For_Analyis ;
             rdfs:subPropertyOf owl:topObjectProperty .

###  http://www.semanticweb.org/metadm.owl#hasColumn
:hasColumn rdf:type owl:ObjectProperty ;
           rdfs:range :Column ;
           rdfs:domain :Table ;
           rdfs:subPropertyOf owl:topObjectProperty ;
```

```
          rdfs:domain [ rdf:type owl:Restriction ;
                        owl:onProperty :hasColumn ;
                        owl:onClass :Table ;
                        owl:minQualifiedCardinality
"1"^^xsd:nonNegativeInteger
                      ] .

###  http://www.semanticweb.org/metadm.owl#hasSelection

:hasSelection rdf:type owl:ObjectProperty ;

              rdfs:range :Table_For_Analyis ;

              rdfs:domain :Table_For_Analyis ;

              rdfs:subPropertyOf owl:topObjectProperty .


###  http://www.semanticweb.org/metadm.owl#hasTransformation

:hasTransformation rdf:type owl:ObjectProperty ;

                   rdfs:domain :Table_For_Analyis ;

                   rdfs:range :Table_For_Analyis ;

                   rdfs:subPropertyOf owl:topObjectProperty .


###  http://www.semanticweb.org/metadm.owl#hasValue

:hasValue rdf:type owl:ObjectProperty ;

          rdfs:domain :Column ,
                      :Table_For_Analyis ;

          rdfs:range :Value ;

          rdfs:subPropertyOf owl:topObjectProperty .


###  http://www.semanticweb.org/metadm.owl#is-PartOf

:is-PartOf rdf:type owl:ObjectProperty ;

           rdfs:domain :Business_Understanding ,
                       :Data_Understanding ;

           rdfs:range :Problem_Understanding ;

           rdfs:subPropertyOf owl:topObjectProperty .


###  http://www.semanticweb.org/metadm.owl#is-PartOfAlgorithm

:is-PartOfAlgorithm rdf:type owl:ObjectProperty ;

                    rdfs:domain :Algorithm ;

                    rdfs:range :Algorithm_Parameter ,
                               :Algorithm_Type ;
```

```
                    rdfs:subPropertyOf owl:topObjectProperty .

###  http://www.semanticweb.org/metadm.owl#uses

:uses rdf:type owl:ObjectProperty ;

      rdfs:domain :Algorithm ;

      rdfs:range :Technic ;

      rdfs:subPropertyOf owl:topObjectProperty .


###  http://www.w3.org/2002/07/owl#topObjectProperty

owl:topObjectProperty rdf:type owl:ObjectProperty .


#################################################################
#
#    Data properties
#
#################################################################

###  http://www.semanticweb.org/metadm.owl#congress

:congress rdf:type owl:DatatypeProperty ;

          rdfs:domain :Data ;

          rdfs:range xsd:string ;

          rdfs:subPropertyOf owl:topDataProperty .


###  http://www.semanticweb.org/metadm.owl#data

:data rdf:type owl:DatatypeProperty ;

      rdfs:domain :Column ,
                  :Value ;

      rdfs:range xsd:string ;

      rdfs:subPropertyOf owl:topDataProperty .


###  http://www.semanticweb.org/metadm.owl#dataDescription

:dataDescription rdf:type owl:DatatypeProperty ;

                 rdfs:domain :Data_Understanding ;

                 rdfs:range xsd:string ;

                 rdfs:subPropertyOf owl:topDataProperty .


###  http://www.semanticweb.org/metadm.owl#holder
```

```
:holder rdf:type owl:DatatypeProperty ;
        rdfs:domain :Pattern ;
        rdfs:range xsd:string ;
        rdfs:subPropertyOf owl:topDataProperty .

###  http://www.semanticweb.org/metadm.owl#objectiveDescription
:objectiveDescription rdf:type owl:DatatypeProperty ;
                      rdfs:domain :Objective ;
                      rdfs:range xsd:string ;
                      rdfs:subPropertyOf owl:topDataProperty .

###  http://www.semanticweb.org/metadm.owl#parameterAlgorithm
:parameterAlgorithm rdf:type owl:DatatypeProperty ;
                    rdfs:domain :Algorithm ;
                    rdfs:range xsd:string ;
                    rdfs:subPropertyOf owl:topDataProperty .

###  http://www.semanticweb.org/metadm.owl#path
:path rdf:type owl:DatatypeProperty ;
      rdfs:domain :Source ;
      rdfs:range xsd:string ;
      rdfs:subPropertyOf owl:topDataProperty .

###  http://www.semanticweb.org/metadm.owl#patternDescription
:patternDescription rdf:type owl:DatatypeProperty ;
                    rdfs:domain :Pattern ;
                    rdfs:range xsd:string ;
                    rdfs:subPropertyOf owl:topDataProperty .

###  http://www.semanticweb.org/metadm.owl#problemDescription
:problemDescription rdf:type owl:DatatypeProperty ;
                    rdfs:domain :Problem ;
                    rdfs:range xsd:string ;
```

```
                    rdfs:subPropertyOf owl:topDataProperty .

###  http://www.semanticweb.org/metadm.owl#resultDescription

:resultDescription rdf:type owl:DatatypeProperty ;
                   rdfs:domain :Result ;
                   rdfs:range xsd:string ;
                   rdfs:subPropertyOf owl:topDataProperty .

###  http://www.semanticweb.org/metadm.owl#trust

:trust rdf:type owl:DatatypeProperty ;
       rdfs:domain :Pattern ;
       rdfs:range xsd:string ;
       rdfs:subPropertyOf owl:topDataProperty .


###  http://www.semanticweb.org/metadm.owl#typeAlgorithm

:typeAlgorithm rdf:type owl:DatatypeProperty ;
               rdfs:domain :Algorithm ;
               rdfs:range xsd:string ;
               rdfs:subPropertyOf owl:topDataProperty .

###  http://www.semanticweb.org/metadm.owl#typeTask

:typeTask rdf:type owl:DatatypeProperty ;
          rdfs:domain :Task ;
          rdfs:range xsd:string ;
          rdfs:subPropertyOf owl:topDataProperty .

###  http://www.semanticweb.org/metadm.owl#typeTechnic

:typeTechnic rdf:type owl:DatatypeProperty ;
             rdfs:domain :Technic ;
             rdfs:range xsd:string ;
             rdfs:subPropertyOf owl:topDataProperty .

###  http://www.w3.org/2002/07/owl#topDataProperty
```

```
owl:topDataProperty rdf:type owl:DatatypeProperty .


#################################################################
#
#    Classes
#
#################################################################

###  http://www.semanticweb.org/metadm.owl#Algorithm

:Algorithm rdf:type owl:Class .


###  http://www.semanticweb.org/metadm.owl#Algorithm_Parameter

:Algorithm_Parameter rdf:type owl:Class ;

                     rdfs:subClassOf :Algorithm .


###  http://www.semanticweb.org/metadm.owl#Algorithm_Type

:Algorithm_Type rdf:type owl:Class ;

                rdfs:subClassOf :Algorithm .


###  http://www.semanticweb.org/metadm.owl#Association

:Association rdf:type owl:Class ;

             rdfs:subClassOf :Task .


###  http://www.semanticweb.org/metadm.owl#Association_Rule

:Association_Rule rdf:type owl:Class ;

                  rdfs:subClassOf :Technic .


###  http://www.semanticweb.org/metadm.owl#Business_Understanding

:Business_Understanding rdf:type owl:Class ;

                        rdfs:subClassOf :Problem_Understanding .


###  http://www.semanticweb.org/metadm.owl#Classification

:Classification rdf:type owl:Class ;

                rdfs:subClassOf :Task .


###  http://www.semanticweb.org/metadm.owl#Clustering

:Clustering rdf:type owl:Class ;

            rdfs:subClassOf :Task .
```

```
###  http://www.semanticweb.org/metadm.owl#Column

:Column rdf:type owl:Class ;

        rdfs:subClassOf :Structure .

###  http://www.semanticweb.org/metadm.owl#Data

:Data rdf:type owl:Class .

###  http://www.semanticweb.org/metadm.owl#Data_Understanding

:Data_Understanding rdf:type owl:Class ;

                    rdfs:subClassOf :Problem_Understanding .

###  http://www.semanticweb.org/metadm.owl#Decision_Tree

:Decision_Tree rdf:type owl:Class ;

               rdfs:subClassOf :Technic .

###  http://www.semanticweb.org/metadm.owl#Genetic_Algorithm

:Genetic_Algorithm rdf:type owl:Class ;

                   rdfs:subClassOf :Technic .

###  http://www.semanticweb.org/metadm.owl#Linear_Regression

:Linear_Regression rdf:type owl:Class ;

                   rdfs:subClassOf :Technic .

###  http://www.semanticweb.org/metadm.owl#Nearest_Neighbor

:Nearest_Neighbor rdf:type owl:Class ;

                  rdfs:subClassOf :Technic .

###  http://www.semanticweb.org/metadm.owl#Neural_Network

:Neural_Network rdf:type owl:Class ;

                rdfs:subClassOf :Technic .

###  http://www.semanticweb.org/metadm.owl#Objective

:Objective rdf:type owl:Class ;

           rdfs:subClassOf :Problem_Understanding .
```

```
###  http://www.semanticweb.org/metadm.owl#Pattern
:Pattern rdf:type owl:Class .

###  http://www.semanticweb.org/metadm.owl#Problem
:Problem rdf:type owl:Class ;
         rdfs:subClassOf :Problem_Understanding .

###  http://www.semanticweb.org/metadm.owl#Problem_Understanding
:Problem_Understanding rdf:type owl:Class .

###  http://www.semanticweb.org/metadm.owl#Result
:Result rdf:type owl:Class .

###  http://www.semanticweb.org/metadm.owl#Source
:Source rdf:type owl:Class ;
        rdfs:subClassOf :Data .

###  http://www.semanticweb.org/metadm.owl#Structure
:Structure rdf:type owl:Class ;
           rdfs:subClassOf :Data .

###  http://www.semanticweb.org/metadm.owl#Table
:Table rdf:type owl:Class ;
       rdfs:subClassOf :Structure .

###  http://www.semanticweb.org/metadm.owl#Table_For_Analyis
:Table_For_Analyis rdf:type owl:Class ;
                   rdfs:subClassOf :Table .

###  http://www.semanticweb.org/metadm.owl#Task
:Task rdf:type owl:Class .

###  http://www.semanticweb.org/metadm.owl#Technic
:Technic rdf:type owl:Class .
```

```
###  http://www.semanticweb.org/metadm.owl#Value

:Value rdf:type owl:Class ;

       rdfs:subClassOf :Data .

#################################################################
#
#    General axioms
#
#################################################################

[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :Source
                :Structure
                :Value
              )
] .
[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :Association_Rule
                :Decision_Tree
                :Genetic_Algorithn
                :Linear_Regression
                :Nearest_Neighbor
                :Neural_Network
              )
] .
[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :Association
                :Classification
                :Clustering
              )
] .
```

APÊNDICE 2

No apêndice 2 é apresentado a instanciação da ontologia META-DM, onde foram inseridos dados do congresso de Tecnologia abordado no Anexo 1.

Esta instanciação foi feita na aba Individuals da ferramenta Protégé e a codificação gerada é apresentada a seguir.

```
###  http://www.semanticweb.org/ontologies/ontodm.owl#BUinst

:BUinst rdf:type :Business_Understanding ,
                 owl:NamedIndividual ;

        :is-PartOf :PUinst ;

        :defines :desc001 ,
                 :desc002 .

###  http://www.semanticweb.org/ontologies/ontodm.owl#Dados

:Dados rdf:type :Table ,
                owl:NamedIndividual ;

       :hasColumn :col007 ,
                  :col008 ,
                  :col009 ,
                  :col010 ,
                  :col011 ,
                  :col012 ,
                  :col013 ,
                  :col014 ,
                  :col015 ,
                  :col016 ,
                  :col017 .

###  http://www.semanticweb.org/ontologies/ontodm.owl#Email

:Email rdf:type :Table ,
                owl:NamedIndividual ;

       :hasColumn :col018 ,
                  :col019 .

###  http://www.semanticweb.org/ontologies/ontodm.owl#Endereco

:Endereco rdf:type :Table ,
                   owl:NamedIndividual ;

          :hasColumn :col001 ,
                     :col002 ,
                     :col003 ,
                     :col004 ,
                     :col005 ,
                     :col006 .
```

```
###  http://www.semanticweb.org/ontologies/ontodm.owl#PUinst

:PUinst rdf:type :Problem_Understanding ,
                 owl:NamedIndividual .


###  http://www.semanticweb.org/ontologies/ontodm.owl#Resultado001

:Resultado001 rdf:type :Result ,
                       owl:NamedIndividual ;

              :resultDescription "Uma das regras geradas forneceu a
seguinte informacao: quem mora em no Estado de Sao Paulo e estuda, ficou
sabendo do congresso atraves da televisao. Estas informacoes tem um grau de
confianca de 70% com um suporte de 20%. Dessa forma para os proximos
congressos caso queira aumentar a presenca desse publico, e recomendavel
continuar a adotar e investir nesta forma de divulgacao."^^xsd:string ;

              :hasAnalysis :padrao001 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#Tarefa001

:Tarefa001 rdf:type :Task ,
                    owl:NamedIndividual ;

           :typeTask "Associacao"^^xsd:string ;

           :definesTechnic :Tecnica001 ;

           :hasAlgorithm :desAlgoritmo01 ;

           :discovers :padrao001 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#Tecnica001

:Tecnica001 rdf:type :Technic ,
                     owl:NamedIndividual ;

            :typeTechnic "Regras de Associacao"^^xsd:string ;

            :definesTechnic :Tecnica001 ;

            :uses :Tecnica001 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#cam001

:cam001 rdf:type :Source ,
                 owl:NamedIndividual ;

        :path "c:Arquivos de
programakiradatabasedados_fontes.fdb"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col001

:col001 rdf:type :Column ,
                 owl:NamedIndividual ;
```

```
        :data "Id_Endereco"^^xsd:string ;

        :hasValue :val001 ,
                  :val020 .

###  http://www.semanticweb.org/ontologies/ontodm.owl#col002

:col002 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Rua"^^xsd:string ;

        :hasValue :val002 ,
                  :val021 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col003

:col003 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Bairro"^^xsd:string ;

        :hasValue :val003 ,
                  :val022 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col004

:col004 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "CEP"^^xsd:string ;

        :hasValue :val004 ,
                  :val023 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col005

:col005 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Cidade"^^xsd:string ;

        :hasValue :val005 ,
                  :val024 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col006

:col006 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Estado"^^xsd:string ;

        :hasValue :val006 ,
                  :val025 .
```

```
###  http://www.semanticweb.org/ontologies/ontodm.owl#col007

:col007 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Id_Inscricao"^^xsd:string ;

        :hasValue :val007 ,
                  :val026 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col008

:col008 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Nome"^^xsd:string ;

        :hasValue :val008 ,
                  :val027 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col009

:col009 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Data_Nascimento"^^xsd:date ;

        :hasValue :val009 ,
                  :val028 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col010

:col010 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Sexo"^^xsd:string ;

        :hasValue :val010 ,
                  :val029 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col011

:col011 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Telefone"^^xsd:string ;

        :hasValue :val011 ,
                  :val030 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col012

:col012 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Escolaridade"^^xsd:string ;
```

```
        :hasValue :val012 ,
                  :val031 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col013

:col013 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Instituicao_Estuda"^^xsd:string ;

        :hasValue :val013 ,
                  :val032 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col014

:col014 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Profissao"^^xsd:string ;

        :hasValue :val014 ,
                  :val033 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col015

:col015 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Ficou_Sabendo"^^xsd:string ;

        :hasValue :val015 ,
                  :val034 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col016

:col016 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Id_Endereco"^^xsd:string ;

        :hasValue :val016 ,
                  :val035 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col017

:col017 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Id_Email"^^xsd:string ;

        :hasValue :val017 ,
                  :val036 .
```

```
###  http://www.semanticweb.org/ontologies/ontodm.owl#col018

:col018 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Id_Email"^^xsd:string ;

        :hasValue :val018 ,
                  :val037 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#col019

:col019 rdf:type :Column ,
                 owl:NamedIndividual ;

        :data "Email"^^xsd:string ;

        :hasValue :val019 ,
                  :val038 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#dados001

:dados001 rdf:type :Data ,
                   owl:NamedIndividual ;

          :congress "Congresso de Tecnologia"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#desAlgoritmo01

:desAlgoritmo01 rdf:type :Algorithm ,
                         owl:NamedIndividual ;

                :typeAlgorithm "Apriori"^^xsd:string ;

                :parameterAlgorithm "Suporte minimo = 30% e Confianca =
70%"^^xsd:string ;

                :uses :Tecnica001 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#desc001

:desc001 rdf:type :Problem ,
                  owl:NamedIndividual ;

         :problemDescription "A divulgacao do congresso gera custo alto e
nem sempre e eficiente."^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#desc002

:desc002 rdf:type :Objective ,
                  owl:NamedIndividual ;

         :objectDescription "Definir a melhor forma de divulgacao do
congresso conforme o perfil do congressista."^^xsd:string .
```

```
###  http://www.semanticweb.org/ontologies/ontodm.owl#desc003

:desc003 rdf:type :Data_Understanding ,
                  owl:NamedIndividual ;

         :dataDescription "Os dados apresentam em sua maioria dados
literais. Os campos que serao utilizados para mineracao de dados sao:
data_nascimento, telefone, CEP, cidade, estado, sexo, profissao e
ficou_sabendo."^^xsd:string ;

         :isDone :BUinst .


###  http://www.semanticweb.org/ontologies/ontodm.owl#padrao001

:padrao001 rdf:type :Pattern ,
                    owl:NamedIndividual ;

           :holder "30%"^^xsd:string ;

           :trust "70%"^^xsd:string ;

           :patternDescription "{Estado=SP, Profissao = Estudante} =>
{FicouSabendo = Televisao}"^^xsd:string ;

           :discovers :Tarefa001 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#registro001

:registro001 rdf:type :Table_for_Analysis ,
                      owl:NamedIndividual ;

             :hasSelection :registro001 ;

             :hasCleaning :registro001 ;

             :hasTransformation :registro001 ;

             :hasValue :val003 ,
                       :val005 ,
                       :val006 ,
                       :val009 ,
                       :val010 ,
                       :val011 ,
                       :val012 ,
                       :val013 ,
                       :val014 ,
                       :val015 ,
                       :val019 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#registro002

:registro002 rdf:type :Table_for_Analysis ,
                      owl:NamedIndividual ;

             :hasCleaning :registro002 ;

             :hasSelection :registro002 ;
```

```
            :hasTransformation :registro002 ;

            :hasValue :val022 ,
                      :val024 ,
                      :val025 ,
                      :val028 ,
                      :val029 ,
                      :val030 ,
                      :val031 ,
                      :val032 ,
                      :val033 ,
                      :val034 ,
                      :val038 .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val001

:val001 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "001"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val002

:val002 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Avenida das Nacoes Unidas n 102"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val003

:val003 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Jardim Santa Clara"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val004

:val004 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "13730-059"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val005

:val005 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Mococa"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val006

:val006 rdf:type :Value ,
                 owl:NamedIndividual ;
```

```
        :data "SP"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val007

:val007 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "001"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val008

:val008 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Rafael T. de Castro Marques"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val009

:val009 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "10/11/1975"^^xsd:date .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val010

:val010 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Masculino"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val011

:val011 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "(19) 9707 - 5034"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val012

:val012 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Primeiro Grau Completo"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val013

:val013 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Outras"^^xsd:string .
```

```
###  http://www.semanticweb.org/ontologies/ontodm.owl#val014

:val014 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Estudante"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val015

:val015 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Televisao"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val016

:val016 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "001"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val017

:val017 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "001"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val018

:val018 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "001"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val019

:val019 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "ratecmg@uol.com.br"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val020

:val020 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "002"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val021

:val021 rdf:type :Value ,
                 owl:NamedIndividual ;
```

```
        :data "Rua Teodomiro Martins de Paula"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val022

:val022 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Vila Formosa"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val023

:val023 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "13720 - 000"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val024

:val024 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Sao Jose do Rio Pardo"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val025

:val025 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "SP"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val026

:val026 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "002"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val027

:val027 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Aline Cristina Ribeiro"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val028

:val028 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "10/06/1988"^^xsd:date .
```

```
###  http://www.semanticweb.org/ontologies/ontodm.owl#val029

:val029 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Feminino"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val030

:val030 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "(19) 3681 - 2168"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val031

:val031 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Primeiro Grau Completo"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val032

:val032 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Fatec - Mococa"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val033

:val033 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Estudante"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val034

:val034 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "Televisao"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val035

:val035 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "002"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val036

:val036 rdf:type :Value ,
                 owl:NamedIndividual ;
```

```
        :data "002"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val037

:val037 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "002"^^xsd:string .


###  http://www.semanticweb.org/ontologies/ontodm.owl#val038

:val038 rdf:type :Value ,
                 owl:NamedIndividual ;

        :data "ribeiro@hotmail.com"^^xsd:string .
```